

DESCRIPTION D'UNE SITUATION PROFESSIONNELLE

PARCOURS	SISR <input type="checkbox"/>	SLAM <input checked="" type="checkbox"/>
----------	-------------------------------	--

Lieu de réalisation	Campus Montsouris	 CFA Cerfal Campus Montsouris
Période de réalisation	Du : 10/02/25	Au : 11/02/25
Modalité de réalisation	SEUL <input checked="" type="checkbox"/>	EN EQUIPE <input type="checkbox"/>

Intitulé de la mission	Développement et intégration d'une vue mensuelle au sein de l'outil de gestion de salle de l'Académie.
Description du contexte de la mission	<p>L'Académie disposait déjà d'un outil de gestion de salle développé via Google Apps Script, HTML, CSS et JavaScript, permettant la gestion des réservations et leur visualisation sous forme de calendrier interactif avec des vues journalière et hebdomadaire. Cet outil avait pour but de remplacer l'utilisation de Google Sheets pour une gestion plus centralisée et intuitive des salles.</p> <p>Afin d'améliorer la planification à long terme et d'offrir une vision d'ensemble plus large des occupations, il a été décidé d'ajouter une fonctionnalité de vue mensuelle à ce calendrier interactif.</p>

Contraintes & Résultat	Ressources fournies / contraintes techniques / Résultats attendu
	<p>Ressources : Google App Script</p> <p>Contraintes : Ne rien utiliser d'autre que la suite Google Workspace</p> <p>Résultat : Outil permettant l'ajout, la modification, la suppression des salles ainsi que leur visualisation dans un calendrier interactif.</p>
Productions associées	Liste des documents produits et description
	Google App Script

Modalités d'accès aux productions	Identifiants, mots de passe, URL d'un espace de stockage et présentation de l'organisation du stockage
	Compte Google, Accès au Tableau Google Sheets, Accès à Internet

Description détaillée de la situation professionnelle retenue et des productions réalisées	
en mettant en évidence la démarche suivie, les méthodes et les techniques utilisées	
Objectifs de la mission 	<ul style="list-style-type: none"> • Analyser les besoins spécifiques pour une vue mensuelle et concevoir son intégration dans l'interface existante.

- Développer la fonctionnalité de vue mensuelle en utilisant les technologies HTML, CSS et JavaScript.
- Assurer la récupération et l'affichage corrects des événements provenant de Google Sheets pour le mois sélectionné.
- Maintenir l'interactivité des événements (modification, consultation des détails) au sein de la nouvelle vue.
- Permettre une navigation intuitive entre les différentes vues (jour, semaine, mois) et entre les mois.
- Garantir la performance et la réactivité de l'application malgré l'augmentation potentielle du volume de données affichées.
- Tester et valider le bon fonctionnement de la vue mensuelle et son intégration globale.

Travaux réalisés

1. Phase de conception et d'analyse :
 - Étude de l'existant (vues journalière et hebdomadaire) pour assurer une cohérence ergonomique et fonctionnelle.
 - Définition de la structure d'affichage des événements sur un mois complet.
 - Planification de l'intégration du nouveau bouton/contrôle de sélection pour la "Vue par mois".
 - Anticipation des aspects de navigation (mois précédent/suivant).
2. Développement de l'interface utilisateur (Front-End) :
 - Modification de la structure HTML pour accueillir la grille mensuelle.


```
<div class="month-view">
  <div class="button-month">
    <div class="button-prev-month"></div>
    <span>/</span>
    <div class="button-next-month"></div>
  </div>
  <div class="month-view-container">
    <div id="chooseMonthView">Fevrier</div>
    <div class="month-view"></div>
    <div class="selector-month-view-part" id="month-view-selector">
      <span>Janvier</span>
      <select>
        <option value="1">Janvier</option>
        <option value="2">Fevrier</option>
        <option value="3">Mars</option>
        <option value="4">Avril</option>
        <option value="5">Mai</option>
        <option value="6">Juin</option>
        <option value="7">Juillet</option>
        <option value="8">Aout</option>
        <option value="9">Septembre</option>
        <option value="10">Octobre</option>
        <option value="11">Novembre</option>
        <option value="12">Decembre</option>
      </select>
    </div>
    <div class="selector-year-view-part" id="month-view-selector-year">
      <span>/</span>
      <select></select>
    </div>
  </div>
  <div class="month-calendar">
    <div class="month-calendar-header">
      <div class="month-calendar-header-day">Lundi</div>
      <div class="month-calendar-header-day">Mardi</div>
      <div class="month-calendar-header-day">Mercredi</div>
      <div class="month-calendar-header-day">Jeudi</div>
      <div class="month-calendar-header-day">Vendredi</div>
      <div class="month-calendar-header-day">Samedi</div>
      <div class="month-calendar-header-day">Dimanche</div>
    </div>
    <div class="month-calendar-content" id="month-calendar-content"></div>
  </div>
</div>
```
 - Stylisation en CSS pour un affichage clair et esthétique de la vue mensuelle, en accord avec le design existant.
 - Développement en JavaScript de la logique d'affichage du calendrier mensuel :
 - Génération dynamique des jours du mois.
 - Mise en place des fonctions pour récupérer les événements pertinents pour le mois affiché depuis la source de données (Google Sheets via Google Apps Script).
 - Positionnement et affichage des événements sur la grille mensuelle.

```
function loadMonth(dateExplicit = '') {
  appearLoadingSpinner();
  const yearSelector = document.querySelector('.selector-year-view-part');
  const yearOfSelector = yearSelector.querySelector('span');
  const yearSelect = yearSelector.querySelector('select');

  const calendarContent = document.getElementById('month-calendar-content');
  calendarContent.innerHTML = '';
  let year = 0;
  let month = 0;
  roomsArray.forEach((room) => {
    if (getLocalStorage(room) == 'false') {
      document.querySelector('.selector-month-view label input[name="' + room + '"]').checked = false;
    } else {
      document.querySelector('.selector-month-view label input[name="' + room + '"]').checked = true;
    }
  });
  const moisListe = ['Janvier', 'Fevrier', 'Mars', 'Avril', 'Mai', 'Juin', 'Juillet', 'Aout', 'Septembre', 'Octobre', 'Novembre', 'Decembre'];
  if (dateExplicit == '') {
    year = new Date(chosenDate).getFullYear();
    month = new Date(chosenDate).getMonth();
  } else {
    year = new Date(dateExplicit).getFullYear();
    month = new Date(dateExplicit).getMonth();
  }

  yearSelector.querySelector('span').innerHTML = year;
  yearSelect.innerHTML = '';
  for (i = 3; i > 0; i--) {
    yearSelect.innerHTML += '<option value="' + year - i + '">' + (year - i) + '</option>';
  }
  for (i = 0; i < 5; i++) {
    yearSelect.innerHTML += '<option value="' + (year + i) + '">' + (year + i) + '</option>';
  }
  yearSelect.querySelectorAll('option').forEach((option) => {
    option.value == year ? (option.selected = true) : (option.selected = false);
  });
  document.querySelectorAll('.selector-month-view-part select option').forEach((option) => {
    option.value == month ? (option.selected = true) : (option.selected = false);
  });
  document.querySelector('.selector-month-view-part span').innerHTML = moisListe[month];
```

3. Intégration des fonctionnalités et de l'interactivité :

- Adaptation de la logique JavaScript existante pour la récupération des données via l'API Google Apps Script afin de charger les événements du mois entier.
- Implémentation de la navigation entre les mois (fonctions "mois précédent" et "mois suivant").
- Conservation des fonctionnalités de clic sur un événement pour afficher les détails ou permettre la modification, adaptées à la vue mensuelle.
- Liaison du bouton "Vue par mois" pour activer l'affichage correspondant et gérer l'état des autres boutons de sélection de vue.

Compétences mobilisées 💡

- Développement Front-End : Maîtrise du HTML, CSS et JavaScript pour la création d'interfaces dynamiques et interactives.
- Manipulation du DOM : Génération et modification dynamique des éléments du calendrier.
- Algorithmique : Logique pour le calcul des jours du mois, la distribution des événements.
- Interaction avec API : Adaptation des appels à Google Apps Script pour la récupération des données mensuelles.
- Conception UI/UX : Réflexion sur l'ergonomie de la nouvelle vue et son intégration harmonieuse.
- Adaptation et évolution de code : Capacité à intervenir sur une base de code existante pour y ajouter de nouvelles fonctionnalités.
- Résolution de problèmes (Debugging) : Identification et correction des erreurs de développement.
- Tests et validation : Mise en place de scénarios de test pour assurer la qualité de la fonctionnalité.

Résultat :

The screenshot shows a user interface for managing events in May 2025. The left sidebar contains a navigation menu with links like 'Ajouter événement', 'Ajouter événement', 'Filtrer', 'Nouveau', 'Rechercher', 'Vue', 'Vue', and 'Paramètres'. The main area is a monthly calendar grid. Each day cell contains multiple event blocks. Most events are labeled with either '(1) Salle des mariages' or '(2) Salle des fêtes'. Specific events are highlighted with larger boxes, such as 'Fondation Edgar Morin' on May 19th and 'Eco-Working' on May 20th. The bottom of the calendar shows time slots from 18:00 to 20:30.