
Ombrières solaires, retrouve ton énergie

SUNPOWER



SOMMAIRE

- Le Projet & Les Besoins
 - Maquettage & Choix
- Répartition du travail & jalons
 - Difficultés rencontrées
 - Un peu de JavaScript
 - Projet finalisé

LE PROJET & LES BESOINS

Le Projet : Proposer un 1er site navigable avec l'ensemble des fonctionnalités demandée par le client SunPower à savoir :

- Présentation de la société et de ses fondateurs
 - Présentation du produit : les ombrières solaires
 - Une carte interactive pour présenter les ombrières déjà installée à Montpellier
 - Présentation des partenaires
 - Un formulaire de contact différenciant entreprises, particuliers, collectivités, syndicats de copropriété
-

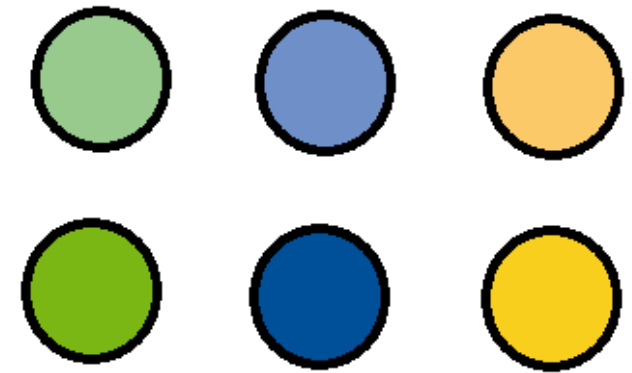
LE PROJET & LES BESOINS

Les Besoins :

- S'adresser à plusieurs publics différents
 - Créer une charte graphique (logo & slogan)
 - Une carte interactive pour présenter les ombrières déjà installée à Montpellier
 - Mettre en avant le produit afin d'attirer à la fois clients et investisseurs
-

MAQUETTAGE & CHOIX

- Avant la phase de maquettage, nous avons étudié le dossier afin de bien prendre en compte les besoins du client.
- Nous avons établi une première maquette et un nuancier à présenter au client lors de notre 1er entretien avec lui.
- Ce premier entretien avait pour but de valider ces choix
- Nous avons choisi de travailler avec la librairie Bootstrap pour sa facilité de mise en œuvre et ne pas multiplier les documents CSS.



RÉPARTITION DU TRAVAIL & JALON

- Charte graphique et zones récurrentes des pages (header / footer) : Tyson
 - Pages informations : Charlie & Tyson
 - Recherche de visuels : Charlie
 - Pages JavaScript : Jérémy
-
- Chaque début / milieu / fin de journée : point équipe pour échanger sur la progression de chaque partie
 - Tous les soirs mise en commun des fichiers en un seul dossier
-

DIFFICULTÉS RENCONTRÉES

Communication (quiproquo)

Mise en commun des fichiers de chacun
(pas de connaissance github)

Ne pas interpréter les manquements
d'informations du client

Contraste de niveau technique

UN PEU DE JAVASCRIPT...

```
const radios = document.getElementsByName('interlocuteur');//je selectionne mes 3 input:radio
```

```
//Pour tous mes boutons radios
radios.forEach(btn => {
  /**L'evenement se déclenche au clic sur l'un d'entres eux
   * et va vérifier lequel est coché avec .checked
   * et va injecter le formulaire html qui suit avec innerHtml
   */
  btn.addEventListener('click', () => {
    if (btn.checked) {
      switch (btn.getAttribute("id")) {
        case "particulier": //cas 1 pour les particuliers
          form.style.border = "1px solid black"; // on fait apparaitre une bordure à notre formulaire
          form.innerHTML = formContact.map((formElement) => //grave au .map on peut sélectionner les éléments de notre tab
            `
              <div class="formContainer">
                <div class ="informations">
                  <label for="${formElement[0]}">${formElement[0]}(*)</label>
                  <input type="text" name="${formElement[0]}" id="${formElement[0]}" required="required">
                  <span id="spName"></span>

                  <label for="${formElement[1]}">${formElement[1]}(*)</label>
                  <input type="text" name="${formElement[1]}" id="${formElement[1]}" required="required">
                  <span id="spFirstname"></span>
            `
          )
        }
      }
    }
  })
})
```

UN PEU DE JAVASCRIPT...

```
/**
 * fonction qui permet de vérifier si un champs de formulaire est correct et d'affichier ou non une erreur
 * @param {string} tag correspond à la string équivalente à l'id de l'input
 * @param {string} span correspond à la string équivalente à l'id du span qui suit l'input
 * @param {string} message message d'erreur à afficher
 * @param {boolean} valid booleen qui renvoi si le champs est rempli de manière valide ou non
 */
const errorDisplay = (tag, span, message, valid) => {
  const container = document.getElementById(tag);
  const errorContainer = document.getElementById(span);
  if (!valid) {
    container.classList.add("text-danger");
    errorContainer.classList.add('text-danger')
    errorContainer.textContent = message;
  } else {
    container.classList.remove("text-danger");
    errorContainer.classList.remove('text-danger')
    errorContainer.textContent = message;
  }
};
```

UN PEU DE JAVASCRIPT...

```
/**
 * fonction qui vérifie le contenu d'un champ mail
 * @param {string} value la valeur rentrée par l'utilisateur
 * @param {string} input valeur à passer en 1er paramètre à errorDisplay() qui correspond à l'id de l'input;
 * @param {string} span valeur à passer en 2eme paramètre à errorDisplay() qui correspond à l'id du span qui suit l'input
 */
const mailChecker = (value, input, span) => {
    if (!value.match(/^[\\w_-]+@[\\w_-]+\\. [a-z]{2,4}$/i)) {
        errorDisplay(input, span, "Entrez une adresse mail valide", false);
        mail = false;
    } else {
        errorDisplay(input, span, "", true);
        mail = true;
    }
};
```

UN PEU DE JAVASCRIPT...

```
inputs.forEach((input) => {  
  //Pour chaque input je crée un événement  
  input.addEventListener("input", (e) => {  
    //j'établis un switch pour tous mes cas d'input différents et pour chacun d'entre eux, j'appelle la fonction de vérification  
    switch (e.target.id) {  
      case "Raison Sociale":  
        textChecker(e.target.value, "Raison Sociale", "spRaisonSoc");  
        break;  
      case "Nom":  
        textChecker(e.target.value, "Nom", "spName", name);  
        break;  
      case "Prénom":  
        textChecker(e.target.value, "Prénom", "spFirstname", surname);  
        break;  
      case "Mail":  
        mailChecker(e.target.value, "Mail", "spMail");  
        break;  
      case "Rôle au sein de votre organisation":  
        textChecker(e.target.value, "Rôle au sein de votre organisation", "spQual");  
        break;  
      case "Adresse":  
        addressChecker(e.target.value, "Adresse", "spAdresse");  
        break;  
    }  
  })  
})
```

PROJET FINALISÉ

Et maintenant si
on allait voir ce
que ça donne ?

