

Créez une page web dynamique avec JavaScript

Rôle :

Développeur front-end pour l'agence ArchiWebos.

Mission :

Travailler sur la conception du site portfolio d'une architecte d'intérieur.

Matériel fourni :

- Maquettes Figma
- Code back-end
- Code front-end version statique
- Kanban de la mission

Objectifs :

- Développer la page de présentation des travaux de l'architecte (à partir du code html fourni),
- Développer la page de connexion administrateur du site ("from scratch"),
- Développer la modale permettant l'envoi de nouveaux projets à l'API ("from scratch" également).

1. Récupération des projets de la galerie et ajout des filtres.

```
// Récupère les travaux
const getWorks = async () => {
  const fetchResult = await fetch("http://localhost:5678/api/works");
  const jsonResult = await fetchResult.json();
  console.log(jsonResult);
  return jsonResult;
}

// Récupère les catégories
const getCat = async () => {
  const fetchResult = await fetch("http://localhost:5678/api/categories");
  const jsonResultCat = await fetchResult.json();
  console.log(jsonResultCat);
  return jsonResultCat;
}
```


2. Réalisation de la connexion.

```
//Redirection vers page d'accueil après connexion
✓ const redirect = function () {
  window.location.href = "index.html";
}

//Afficher message d'erreur
✓ const error = function () {
✓   if (!document.getElementById('error')) {
    const parentNode = document.querySelector("form");
    const beforeNode = document.getElementById("password");
    const msgErreur = document.createElement("p");
    msgErreur.classList.add("error");
    msgErreur.id = "error";
    msgErreur.innerHTML = 'Erreur dans l'identifiant ou le mot de passe';
    parentNode.insertBefore(msgErreur, beforeNode);
  }
}

✓ async function LogIn(event) {
  event.preventDefault();
  const email = document.getElementById("email").value;
  const password = document.getElementById("password").value;
  ✓ const user = {
    "email": email,
    "password": password,
  }; // créé objet user
  console.log(JSON.stringify(user));

  ✓ const fetchResult = await fetch("http://localhost:5678/api/users/login", {
    method: "POST",
    ✓ headers: {
      "Content-Type": "application/json",
    },
    body: JSON.stringify(user),
  });
  const jsonResult = await fetchResult.json();

  ✓ if (jsonResult.userId) {
    localStorage.setItem("token", jsonResult.token);
    redirect();
  } else {
    error();
  }
}

//Bouton fonctionnel
const authButton = document.getElementById("auth");
console.log(authButton.value);
authButton.addEventListener('click', LogIn);
```

3. Ajout de travaux à la galerie.

```
//Envoyer nvx travaux
const workForm = document.getElementById("new-work_form");
const sendWork = document.getElementById("valid");

const newProject = async (e) => {
  e.preventDefault();
  const token = localStorage.getItem("token");
  const formulaire = new FormData();
  formulaire.append("title", workForm.title.value);
  formulaire.append("category", workForm.category.value);
  formulaire.append("image", imageUpload.files[0]);
  const response = await fetch("http://localhost:5678/api/works", {
    method: 'POST',
    headers: {
      'Authorization': `Bearer ${token}`
    },
    body: formulaire,
  });
  console.log(formulaire);
  const result = await response.json();
  console.log(result);
  localStorage.setItem("log", JSON.stringify(result));
  console.log(formulaire.get("image"));

  //Affichage dynamique des nouveaux projets
  let figure = document.createElement("figure");
  figure.id = "display" + result.id;
  let figcaption = document.createElement("figcaption");
  figcaption.innerHTML = result.title;
  let image = document.createElement("img");
  image.src = result.imageUrl;
  image.alt = result.title;
  figure.appendChild(image);
  figure.appendChild(figcaption);
  document.getElementById("divForWorks").appendChild(figure);
  const figureModale = figure.cloneNode(true);
  const trash = document.createElement("i");
  trash.classList.add("fa-regular", "fa-trash-can", "fa-2xs");
  figureModale.querySelector("img").insertAdjacentElement("beforebegin", trash);
  document.getElementById("listing").appendChild(figureModale);
}
```

4. Points d'amélioration

- Découpage du code par fonctionnalités
- Gestion d'erreur pour les appels Fetch

5. Difficultés

- Appels Fetch
- Utilisation des paramètres / arguments