Documentation Stage Soup & Juice

Introduction

Durant mon stage j'ai réalisé le développement d'un espace de formation sur un site Wix existant, en utilisant JavaScript avec le mode Wix Velo. L'objectif principal est de gérer l'authentification des utilisateurs, la redirection et la gestion des sessions et l'ajout de vidéo de formation sur le site.

Fichier : Sign in page.js

Fonction : hashPassword(password)

- Description : Identique à celle du fichier login page.js. Elle hache le mot de passe en utilisant SHA-256.

Fonction : $w.onReady(function () {...})

- Description : Cette fonction est exécutée lorsque la page est prête. Elle initialise les événements de la page.
- Fonctionnement :
- Associe la fonction signupButtonclick à l'événement de clic sur le bouton d'inscription.
- Cache le message d'erreur initial.

Fonction : signupButtonclick(event)

- Description : Gère le processus d'inscription des nouveaux utilisateurs.
- Fonctionnement :
- Récupère les valeurs saisies pour l'email, le nom et le prénom.
- Vérifie si tous les champs sont remplis. Si non, affiche un message d'erreur.
- Vérifie si l'email est valide à l'aide d'une expression régulière.
- Effectue une requête à la base de données pour vérifier si l'email est déjà utilisé.
- Génère un mot de passe aléatoire, le hache et crée un nouvel utilisateur dans la base de données.
- Crée un contact dans Wix CRM et envoie un email de confirmation.
- Redirige l'utilisateur vers la page de connexion après une inscription réussie.

```
import wixData from 'wix-data';
import { triggeredEmails } from 'wix-crm';
import { contacts } from 'wix-crm';
import wixLocation from 'wix-location';

async function hashPassword(password) {
    const encoder = new TextEncoder();
    const data = encoder.encode(password);
    const hashBuffer = await crypto.subtle.digest('SHA-256', data);
    const hashArray = Array.from(new Uint8Array(hashBuffer));
    return hashArray.map(b => b.toString(16).padStart(2, '0')).join('');
}

$w.onReady(function () {
    $w("#signupButton").onClick(signupButtonclick);
    $w("#text1").hide();
});

export async function signupButtonclick(event) {
    const email = $w("#emailInput").value.trim();
    const nom = $w("#nomInput").value.trim();
    const prenom = $w("#prenomInput").value.trim();

    if (!nom || !prenom || !email) {
        $w("#text1").text = "Veuillez remplir tous les champs";
        $w("#text1").show();
        return;
    }

    const emailRegex = /^[a-zA-Z0-9._-]+@[a-zAZ0-9.-]+\.[a-zA-Z]{2,}$/;
    if (!emailRegex.test(email)) {
        $w("#text1").text = "Adresse email invalide";
        $w("#text1").show();
        return;
    }
```

```
        await wixData.insert("user", {
            email: email,
            nom: nom,
            prenom: prenom,
            password: hashedPassword,
            startdate: startdate.toISOString().split('T')[0],
            expirydate: expirydate.toISOString().split('T')[0],
            contactid: contactId
        });
        setTimeout(() => {
            wixLocation.to("/login");
        }, 1000);

        $w("#text1").text = "Inscription réussie ! Consultez votre email pour récupérer votre mot de passe.";
        $w("#text1").style.color = "#82917B";
        $w("#text1").show();
    } catch (error) {
        $w("#text1").text = `Une erreur est survenue lors de l'inscription : ${error.message}`;
        $w("#text1").show();
    }
}
```

```
    try {
        const existingUser = await wixData.query("user")
            .eq("email", email)
            .find();

        if (existingUser.items.length > 0) {
            $w("#text1").text = "Cet email est déjà utilisé";
            $w("#text1").show();
            return;
        }

        const password = Array.from({ length: 8 }, () => "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456
        const hashedPassword = await hashPassword(password);

        const startdate = new Date();
        startdate.setHours(0, 0, 0, 0);

        const expirydate = new Date(startdate);
        expirydate.setDate(startdate.getDate() + 30);
        expirydate.setHours(0, 0, 0, 0);

        const newContact = await contacts.appendOrCreateContact({
            name: {
                first: nom,
                last: prenom,
            },
            emails: [{
                email: email
            }]
        });

        const contactId = newContact.contactId;

        triggeredEmails.emailContact('Ua2Ow0a', contactId, {
            variables: {}
        });
```

Fichier : login page.js

Fonction : hashPassword(password)

- Description : Cette fonction prend un mot de passe en clair et le hache en utilisant l'algorithme SHA-256.
- Fonctionnement :
- Utilise TextEncoder pour encoder le mot de passe en un tableau de bytes.
- Applique crypto.subtle.digest pour générer le hachage.
- Convertit le hachage en un tableau hexadécimal et le retourne sous forme de chaîne.

Fonction : loginButtonClick(event)

- Description : Cette fonction est appelée lorsque l'utilisateur clique sur le bouton de connexion. Elle gère le processus de connexion.
- Fonctionnement :
- Récupère l'email et le mot de passe saisis par l'utilisateur.
- Vérifie si les champs sont remplis. Si non, affiche un message d'erreur.
- Appelle checkPasswordExpiration(email) pour vérifier si le mot de passe de l'utilisateur a expiré.
- Hache le mot de passe saisi et effectue une requête à la base de données pour trouver l'utilisateur correspondant à l'email.
- Si l'utilisateur n'existe pas ou si le mot de passe ne correspond pas, affiche un message d'erreur.
- Si la connexion est réussie, stocke les informations de l'utilisateur dans la session et redirige vers la page de redirection.

Fonction : checkPasswordExpiration(email)

- Description : Vérifie si le mot de passe de l'utilisateur a expiré.
- Fonctionnement :
- Effectue une requête à la base de données pour trouver l'utilisateur par email.
- Compare la date actuelle avec la date d'expiration du mot de passe.
- Retourne un objet indiquant si le mot de passe est valide, expiré ou si l'utilisateur n'est pas trouvé.

```
import wixData from 'wix-data';
import { session } from 'wix-storage';
import wixLocation from 'wix-location';

$w.onReady(function () {
    $w("#loginButton").onClick(loginButtonClick);
    $w("#text1").hide();
});

async function hashPassword(password) {
    const encoder = new TextEncoder();
    const data = encoder.encode(password);
    const hashBuffer = await crypto.subtle.digest('SHA-256', data);
    const hashArray = Array.from(new Uint8Array(hashBuffer));
    return hashArray.map(b => b.toString(16).padStart(2, '0')).join('');
}

export async function loginButtonClick(event) {
    const email = $w("#emailInput").value;
    const password = $w("#passwordInput").value.trim();

    if (!email || !password) {
        $w("#text1").text = "Veuillez remplir tous les champs";
        $w("#text1").show();
        return;
    }

    try {
        const expirationStatus = await checkPasswordExpiration(email);

        if (!expirationStatus.valid && expirationStatus.expired) {
            $w("#text1").text = expirationStatus.message;
            $w("#text1").show();
            return;
        }

        const hashedPassword = await hashPassword(password);
        const userQuery = await wixData.query("user")
            .eq("email", email)
            .find();

        if (userQuery.items.length === 0) {
            $w("#text1").text = "Email ou mot de passe incorrect";
            $w("#text1").show();
            return;
        }

        const user = userQuery.items[0];

        if (user.password !== hashedPassword) {
            $w("#text1").text = "Email ou mot de passe incorrect";
            $w("#text1").show();
            return;
        }

        session.setItem("userEmail", email);
        session.setItem("userId", user._id);
        session.setItem("userNom", user.nom);
        session.setItem("userPrenom", user.prenom);

        $w("#text1").text = "Connexion réussie ! Redirection en cours...";
        $w("#text1").style.color = "#82917B";
        $w("#text1").show();

        await new Promise(resolve => setTimeout(resolve, 1000));

        wixLocation.to("/redirection");

    } catch (error) {
        $w("#text1").text = "Une erreur est survenue lors de la connexion";
        $w("#text1").show();
    }
}
```

```
async function checkPasswordExpiration(email) {
    try {
        const user = await wixData.query("user")
            .eq("email", email)
            .find();

        if (user.items.length === 0) {
            return { valid: false, message: "Utilisateur non trouvé" };
        }

        const userData = user.items[0];
        const today = new Date();
        today.setHours(0, 0, 0, 0);
        const expiryDate = new Date(userData.expirydate);

        if (today > expiryDate) {
            return {
                valid: false,
                expired: true,
                message: "Votre compte a expiré. Veuillez le renouveler."
            };
        }

        const daysUntilExpiry = Math.ceil((expiryDate.getTime() - today.getTime()) / (1000 * 60 * 60 * 24));
        if (daysUntilExpiry <= 7) {
            return {
                valid: true,
                warning: true,
                message: `Votre compte expirera dans ${daysUntilExpiry} jours`
            };
        }

        return { valid: true };
    } catch (error) {
        return { valid: false, message: "Erreur lors de la vérification" };
    }
```

Fonctionnalités Principales :

- Gestion de la Connexion : Vérifie les informations d'identification de l'utilisateur et gère les sessions.
- Vérification de l'Expiration du Mot de Passe : Vérifie si le mot de passe de l'utilisateur a expiré.

Fichier : redirection page.js

Fonction : $w.onReady(async () => {...})

- Description : Cette fonction est exécutée lorsque la page est prête. Elle gère la vérification de l'authentification de l'utilisateur.
- Fonctionnement :
- Récupère l'ID utilisateur stocké dans la session.
- Si l'ID n'existe pas, redirige l'utilisateur vers la page d'accès refusé.
- Effectue une requête à la base de données pour vérifier si l'utilisateur existe.
- Si l'utilisateur est trouvé, redirige vers la page de formation.

```
import wixLocation from 'wix-location';
import { session } from 'wix-storage';
import wixData from 'wix-data';

$w.onReady(async () => {
    const userId = session.getItem("userId"); // Récupérer l'ID utilisateur
    if (!userId) {
        wixLocation.to('/acces-refuse'); // Rediriger vers la page d'accès refusé
        return;
    }

    try {
        // Vérifier si l'utilisateur existe dans la base de données
        const userQuery = await wixData.query("user")
            .eq("_id", userId) // Recherche par ID utilisateur
            .find();

        if (userQuery.items.length === 0) {
            wixLocation.to('/acces-refuse'); // Rediriger vers la page d'accès refusé
            return;
        }

        wixLocation.to('/formation'); // Rediriger vers la page de formation

        // Si l'utilisateur est trouvé, récupérer ses informations
        const user = userQuery.items[0];

    } catch (error) {
        console.error("Erreur lors de la vérification de l'utilisateur :", error);
    }
});
```

Vérification de l'Utilisateur

Le code suivant est utilisé pour vérifier si un utilisateur est connecté avant d'accéder à l'espace de formation :

```
import wixLocation from 'wix-location';
import { session } from 'wix-storage';
import wixData from 'wix-data';

$w.onReady(async () => {
    const userId = session.getItem("userId");
    if (!userId) {
        wixLocation.to('/acces-refuse');
        return;
    }

    try {
        const userQuery = await wixData.query("user")
            .eq("_id", userId)
            .find();

        if (userQuery.items.length === 0) {
            wixLocation.to('/acces-refuse');
            return;
        }
    } catch (error) {
        console.error("Erreur lors de la vérification de l'utilisateur :", error);
    }
});
```

Importation des Modules :

- Le code commence par importer les modules nécessaires pour la navigation (wix-location), la gestion des sessions (wix-storage), et l'accès aux données (wix-data).
- **Vérification de la Session** : Lors de l'initialisation de la page, le code vérifie si un userId est stocké dans la session. Si ce n'est pas le cas, l'utilisateur est redirigé vers une page d'accès refusé.
- **Requête à la Base de Données** : Si un userId est présent, le code effectue une requête pour vérifier si l'utilisateur existe dans la base de données. Si l'utilisateur n'est pas trouvé, il est également redirigé vers la page d'accès refusé.
- **Gestion des Erreurs** : En cas d'erreur lors de la requête, un message d'erreur est affiché dans la console.