

Nom : Fofana
Prénoms : Mamadou Lamine
Classe : GLSI A

Rapport Technique : Formulaire de Vérification d'Identité

1. Introduction

Ce document décrit la structure et le fonctionnement d'une page web dédiée à la vérification d'identité dans le cadre d'une plateforme de parrainage. La page est conçue pour permettre aux utilisateurs de saisir leurs informations d'identité (numéro d'électeur, numéro de CIN, et code d'authentification) et de vérifier leur éligibilité pour parrainer un candidat.

2. Structure du Code

2.1. En-tête HTML

- DOCTYPE: Déclare le type de document comme HTML5.
- Langue: La langue est définie sur `fr` (français) pour une meilleure accessibilité.
- Méta-informations:
 - `charset="UTF-8"` : Assure la prise en charge des caractères spéciaux.
 - `viewport` : Rend la page responsive pour les appareils mobiles.
- Titre: "Vérification d'Identité - Parrainage" s'affiche dans l'onglet du navigateur.

2.2. Intégration des Bibliothèques

- Bootstrap: Utilisé pour le style et la mise en page responsive.
- Font Awesome: Fournit des icônes pour améliorer l'interface utilisateur.

2.3. Styles Personnalisés

- Body:
 - Flexbox est utilisé pour centrer le contenu verticalement et horizontalement.
 - Le fond est de couleur claire (`#f0f0f0`) pour un contraste agréable.
- Navbar:
 - Barre de navigation noire (`#333`) avec un texte blanc.
 - Le texte "Plateforme Parrains" est cliquable et renvoie vers la page d'accueil (`index.html`).
- Formulaire:
 - Conteneur blanc avec des bordures arrondies et une ombre pour un effet moderne.
 - Les champs de formulaire sont espacés et accompagnés d'icônes pour une meilleure lisibilité.
- Footer:
 - Barre noire pleine largeur en bas de la page, contenant le texte "© 2025 Plateforme des Parrains - Tous droits réservés".

3. Fonctionnalités

3.1. Formulaire de Vérification

- Champs de Formulaire:
 - Numéro d'électeur: Champ texte pour saisir le numéro d'électeur.

- Numéro de CIN: Champ texte pour saisir le numéro de carte d'identité nationale.
- Code d'authentification: Champ texte pour saisir un code unique.
- Validation:
 - Tous les champs sont obligatoires (`required`).
 - Les données sont envoyées via une requête HTTP POST.

3.2. Affichage des Résultats

- Message de Retour:
 - Un message dynamique est affiché en fonction de la réponse du serveur (succès ou échec).
 - Utilisation de `alert` Bootstrap pour un affichage visuel clair.
- Informations de l'Utilisateur:
 - Si la vérification est réussie, les informations de l'utilisateur (nom, prénom, date de naissance, bureau de vote) sont affichées.
 - Un bouton "Voir la liste des candidats" redirige vers une autre page (`list_candidates.html`).

3.3. Stockage Local

- Les données de l'utilisateur (numéro d'électeur, numéro de CIN, code d'authentification) sont stockées dans le `localStorage` du navigateur pour une utilisation ultérieure.

4. Fonctionnement Technique

4.1. JavaScript

- Écouteur d'Événement:
 - Le formulaire écoute l'événement `submit` pour intercepter la soumission.
 - `e.preventDefault()` empêche le rechargement de la page.
- Envoi des Données:
 - Les données du formulaire sont récupérées via `FormData` et envoyées au serveur via une requête `fetch`.
 - La méthode HTTP utilisée est `POST`, avec un en-tête `Content-Type: application/json`.
- Traitement de la Réponse:
 - Si la réponse est positive, les informations de l'utilisateur sont affichées et stockées.
 - En cas d'erreur, un message d'alerte est affiché.

4.2. Intégration avec le Backend

- Le formulaire envoie les données à l'URL suivante :

```

```javascript
fetch('http://localhost:8000/parrainages/register_sponsorship.php', {
 method: 'POST',
 headers: { 'Content-Type': 'application/json' },
 body: JSON.stringify(data)
});
```

```

- Le backend doit renvoyer une réponse JSON avec :
 - `success` : Booléen indiquant si la vérification a réussi.
 - `message` : Message à afficher à l'utilisateur.

- `data` : Informations de l'utilisateur (nom, prénom, etc.).

5. Choix de Conception

5.1. Interface Utilisateur

- Simplicité : Le formulaire est minimaliste et facile à utiliser.
- Icônes : Les icônes de Font Awesome améliorent la compréhension des champs.
- Responsive Design : La page s'adapte à tous les écrans grâce à Bootstrap.

5.2. Expérience Utilisateur

- Feedback Immédiat : Les messages d'erreur ou de succès sont affichés immédiatement après la soumission.
- Redirection : Après une vérification réussie, l'utilisateur peut accéder à la liste des candidats en un clic.

5.3. Sécurité

- Validation Côté Client : Les champs obligatoires sont validés avant l'envoi.
- Stockage Local : Les données sensibles sont stockées localement pour éviter de les renvoyer plusieurs fois.

6. Conclusion

Ce formulaire de vérification d'identité est conçu pour être simple, efficace et visuellement attrayant. Il intègre des technologies modernes (Bootstrap, Font Awesome, JavaScript) pour offrir une expérience utilisateur fluide. La communication avec le backend est robuste, et les données sont traitées de manière sécurisée. Ce code peut être facilement étendu ou modifié pour s'adapter à des besoins spécifiques.

```

<!-- Contenu principal -->
<main>
  <div class="form-container">
    <h2 class="text-center text-primary mb-4">Vérification d'Identité</h2>
    <div id="message"></div>
    <form id="verifyForm">
      <fieldset>
        <legend>Informations de vérification</legend>
        <div class="input-group">
          <i class="fas fa-id-card"></i>
          <input type="text" name="numero_electeur" class="form-control" placeholder="Numéro d'électeur" required>
        </div>
        <div class="input-group">
          <i class="fas fa-id-badge"></i>
          <input type="text" name="numero_cin" class="form-control" placeholder="Numéro de CIN" required>
        </div>
        <div class="input-group">
          <i class="fas fa-key"></i>
          <input type="text" name="code_auth" class="form-control" placeholder="Code d'authentification" required>
        </div>
        <button type="submit" class="btn btn-primary">Vérifier</button>
      </fieldset>
    </form>
    <div id="parrainInfo" class="mt-3">
      <h4>Vos informations :</h4>
      <p id="infoDisplay"></p>
      <a href="list_candidates.html" class="btn btn-success">Voir la liste des candidats</a>
    </div>
  </div>
</main>

```

```

<!-- Footer -->
<footer>
  <p class="mb-0">&copy; 2025 Plateforme des Parrains – Tous droits réservés</p>
</footer>

<!-- Bootstrap JS -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
<script>
  const form = document.getElementById('verifyForm');
  const messageDiv = document.getElementById('message');
  const parrainInfoDiv = document.getElementById('parrainInfo');
  const infoDisplay = document.getElementById('infoDisplay');

  form.addEventListener('submit', async (e) => {
    e.preventDefault();
    messageDiv.innerHTML = '';
    const data = {};
    new FormData(form).forEach((value, key) => data[key] = value.trim());

    try {
      const response = await fetch('http://localhost:8000/parrainages/register_sponsorship.php', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(data)
      });
      const res = await response.json();
      messageDiv.innerHTML = `<div class="alert alert-${res.success ? 'success' : 'danger'}">${res.message}</div>`;
      if (res.success && res.data) {
        localStorage.setItem('numero_electeur', data.numero_electeur);
        localStorage.setItem('numero_cin', data.numero_cin);
        localStorage.setItem('code_auth', data.code_auth);
        infoDisplay.innerHTML = `
          <strong>Nom :</strong> ${res.data.nom}<br>
          <strong>Prénom :</strong> ${res.data.prenom}<br>
          <strong>Date de naissance :</strong> ${res.data.date_naissance}<br>
          <strong>Bureau de vote :</strong> ${res.data.bureau_vote}
        `;
        parrainInfoDiv.style.display = 'block';
        form.style.display = 'none';
      }
    } catch (err) {
      messageDiv.innerHTML = `<div class="alert alert-danger">Erreur lors de la vérification.</div>`;
    }
  });
</script>
</body>
</html>

```

Vérification d'Identité

Informations de vérification



Numéro d'électeur



Numéro de CIN



Code d'authentification

Vérifier