

Portfolio - Intégration

Nom : Eladji Fofana

Formation : Bachelor Informatique - École Hexagone Versailles

Aperçu

Ce dépôt contient l'intégration HTML/CSS/Tailwind d'une maquette Figma de portfolio.

- Page principale : [index.html](#)

Capture d'écran

Ajoutez ici vos captures d'écran (desktop, tablette, mobile) dans le dossier `assets/` puis insérez les images ci-dessous :

- Aperçu desktop : assets/desktop.png
- Aperçu tablet : assets/tablet.png
- Aperçu mobile : assets/mobile.png

(Option : vous pouvez générer des captures avec votre navigateur et les déposer dans `assets/`.)

Maquette Figma utilisée

Lien de la maquette choisie :

<https://www.figma.com/design/NMM4HyjAW9aDayZmnTdWre/Juan-Simmons---Portfolio-Website--Community--?node-id=78-874&m=dev>

Inclure dans le rendu : copie du lien Figma et captures d'écran de la maquette.

—
Eladji

Langages & frameworks

- HTML5
- CSS3
- Tailwind CSS (via CDN pour l'intégration statique)
- JavaScript (vanilla) pour validations et animations simples

Stack technique (du CV) :

- Backend : Python, Go, Rust, C++
 - Frontend : JavaScript, React, HTML/CSS
 - IoT : Arduino, ESP32, CAD (Solidworks, Fusion)
 - Autres : DevOps, Unix/Linux, Cisco, Gestion réseau
-

Structure du dépôt

```
.  
├── index.html          # Page principale  
├── api/  
│   └── contact.js      # Endpoint serverless pour formulaire  
└── contact  
    ├── public/           # Dossier pour les images et assets  
    │   ├── photo de profile.jpg  
    │   ├── arduino-logo-1.png  
    │   └── ... (autres images)  
    ├── docs/              # Documentation du projet  
    │   ├── cv alternence.pdf  
    │   ├── Projet final...pdf  
    │   └── ...  
    ├── vercel.json        # Configuration Vercel  
    ├── README.md          # Ce fichier  
    └── .vscode/            # Configuration VS Code
```

Exécuter le projet en local

Le projet est statique. Pour le tester localement, ouvrez `index.html` dans un navigateur ou servez-le avec un serveur HTTP simple.

Avec Python 3 (serveur simple) :

```
# depuis le dossier contenant index.html  
python3 -m http.server 8000  
# puis ouvrez http://localhost:8000
```

Avec Node.js (serveur statique rapide) :

Eladji

```
# installer http-server si nécessaire  
npm install -g http-server  
http-server -c-1  
# puis ouvrir l'URL fournie (ex. http://127.0.0.1:8080)
```

Déploiement (suggestions)

- **Vercel** (recommandé) : déploiement auto du site + backend API (serverless)
- GitHub Pages : site statique uniquement (formulaire contact ne fonctionne pas sans backend)
- Netlify : déploiement auto du site + Netlify Functions pour le backend

Déploiement rapide sur Vercel

Prérequis :

1. Compte Vercel : <https://vercel.com>
2. Ton webhook URL Discord (à garder secrète)

Étapes :

```
# 1. Si nécessaire, initialise Git
git init
git add .
git commit -m "Portfolio avec backend"

# 2. Installer Vercel CLI
npm install -g vercel

# 3. Déployer
vercel
# Réponds aux questions (connecte-toi à ton compte Vercel)
```

Après le déploiement :

1. Va dans les **Settings** du projet Vercel
2. Ajoute une variable d'environnement :
 - **Clé** : **DISCORD_WEBHOOK_URL**
 - **Valeur** : Colle ton webhook URL Discord
3. **Redéploie** le projet (ou il se redéploiera automatiquement)

Le formulaire contact enverra maintenant les messages vers Discord via ton backend sécurisé !

Accessibilité & bonnes pratiques

—
Eладжи

- Tous les champs du formulaire ont des **label** associés.
- Les images ont des attributs **alt** descriptifs.
- Contrastes de couleurs vérifiés par rapport à la maquette (foncés / clairs).
- Le formulaire effectue une validation côté client (JS) : champs requis et format d'email.
- Test lighthouse : <https://lighthouse-metrics.com/lighthouse/checks/17ff9c4e-02a6-4460-a43d-95cb15263ba3>

Checklist pour la remise (conforme aux consignes)

1. Respect de la maquette (graphisme, typographies, couleurs) — à vérifier visuellement.
2. Qualité du responsive — vérifier sur smartphone / tablette / desktop.
3. Structure du code — HTML sémantique et clair.
4. Accessibilité — labels, alt, focus states.
5. Fonctionnalités front — formulaire avec validations.
6. README.md présent et clair (vous êtes ici).

Informations de contact

- LinkedIn : (ajoutez votre profil si disponible)
-

Remarques finales

- Ce projet respecte les contraintes : pas d'utilisation de frameworks JS (React/Vue), Tailwind utilisé comme framework CSS via CDN, JavaScript vanilla pour les interactions.
- Si vous souhaitez que je génère des captures d'écran automatiques, ajoute des images dans `assets/` ou indique-moi les vues à capturer et je peux créer des mini-scripts pour aider.

Bonne livraison ! 