

Documentation Technique - **Diploma Gate Guardian**

1. Introduction et Objectifs

Contexte du projet

Ce projet vise à créer un site web permettant la vérification d'authenticité des certificats émis par une institution (école, université, organisme de formation, etc.). Face à la montée des fraudes documentaires, notamment des faux diplômes ou attestations, la vérification automatisée devient essentielle.

Problèmes que ce projet résout

- Lutte contre la falsification de certificats.
- Réduction du traitement manuel de demandes de vérification.
- Accès en ligne sécurisé et simple pour la vérification.
- Centralisation des certificats et de leur statut d'authenticité.

Objectifs principaux

- Authentification sécurisée.
- Scalabilité future.
- Importation simplifiée des étudiants via fichiers CSV/Excel.
- Réduction de la fraude documentaire.
- Validation officielle des documents par l'établissement.

Public cible

- Utilisateurs finaux : recruteurs, entreprises, établissements tiers.
- Administrateurs : personnel de l'établissement émetteur.
- Étudiants : peuvent visualiser et transmettre leur certificat.

2. Architecture du Site

2.1 Schéma de l'architecture

(Schéma visuel à intégrer ici plus tard lorsque les services cloud seront utilisés.)

2.2 Composants utilisés (actuellement en local)

- **Frontend** : Next.js avec TypeScript, Tailwind CSS.
- **Backend** : API Routes Next.js (en local), logique personnalisée.
- **Base de données** : JSON ou stockage temporaire.
- **Authentification** : JWT local (Cognito prévu plus tard).
- **Importation** : fichiers CSV ou Excel injectés localement.

Des services AWS tels que S3, Lambda, API Gateway, Cognito et RDS sont envisagés pour une future version hébergée.


3. Développement Frontend

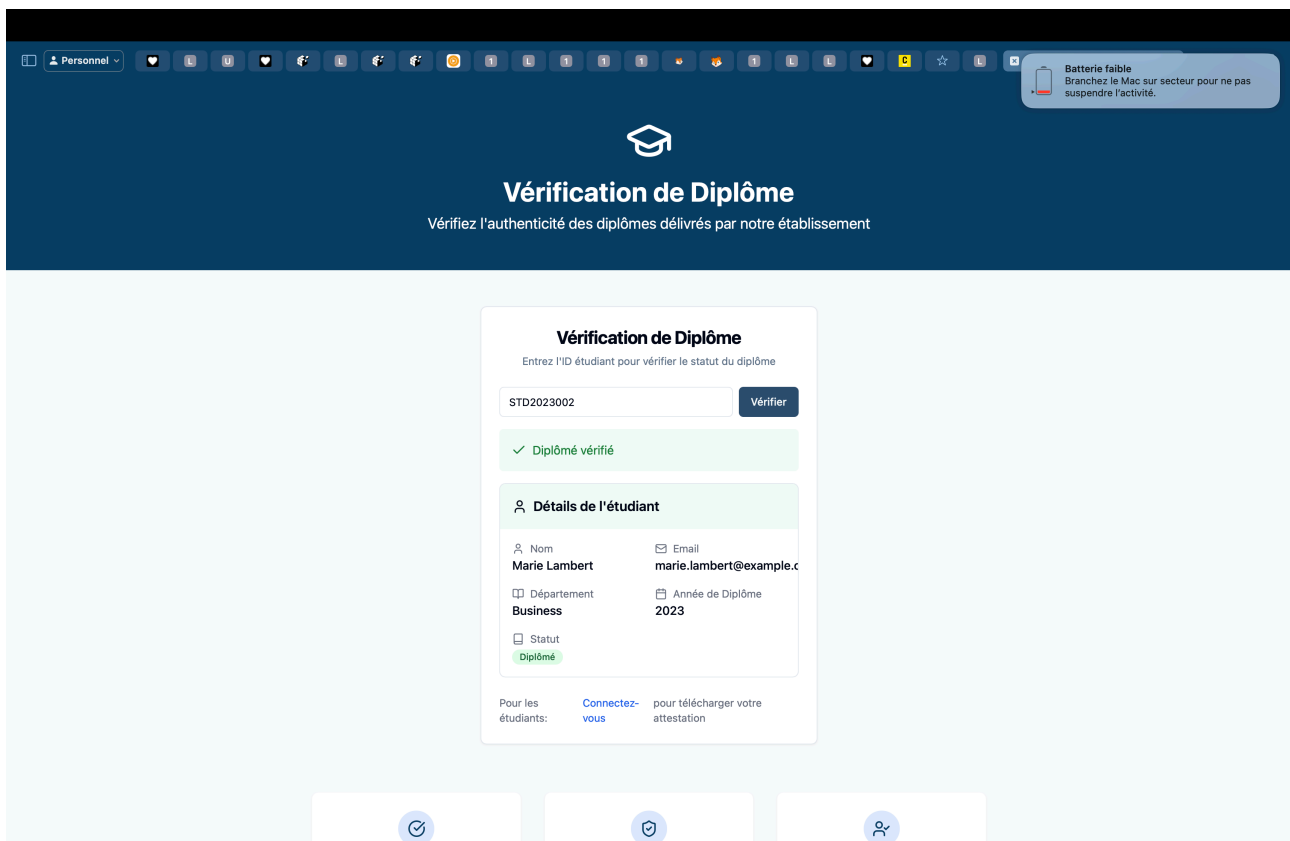
Technologies utilisées

- **Framework** : Next.js (React) avec TypeScript.
- **Styling** : Tailwind CSS.
- **Développement** : en local avec npm run dev.

Pages développées

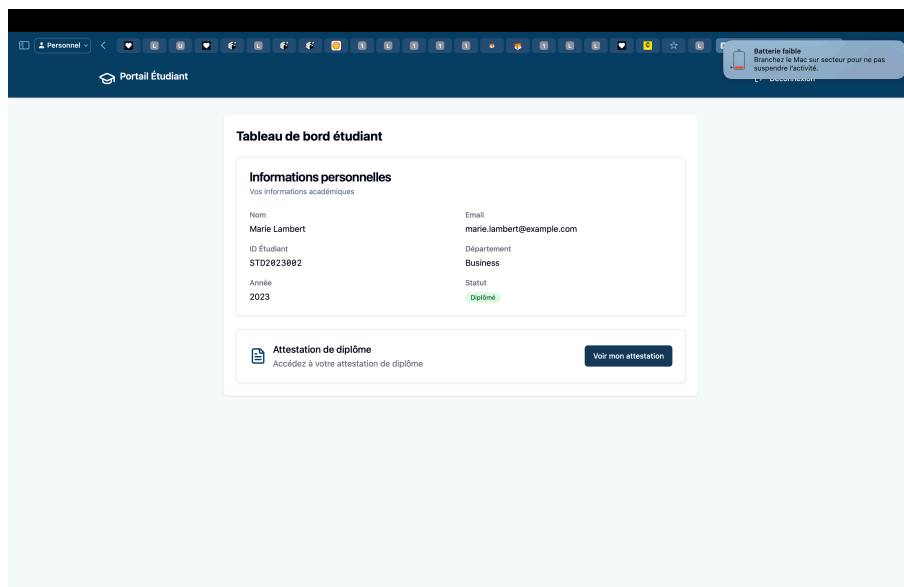
a. Page d'accueil externe

- Présente un champ de saisie de code unique.
- Permet à toute administration externe (entreprise, école) de vérifier l'authenticité d'un certificat à l'aide du code fourni par un étudiant.
-  **Capture d'écran à prévoir** : interface de saisie du code avec message de validation ou d'erreur.



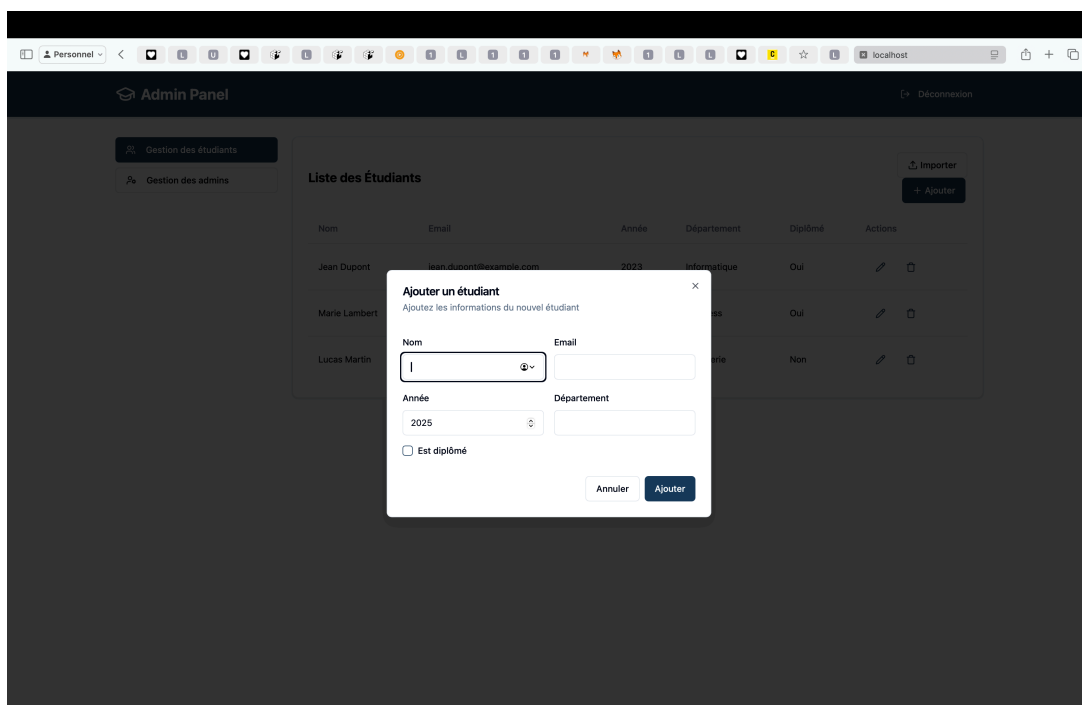
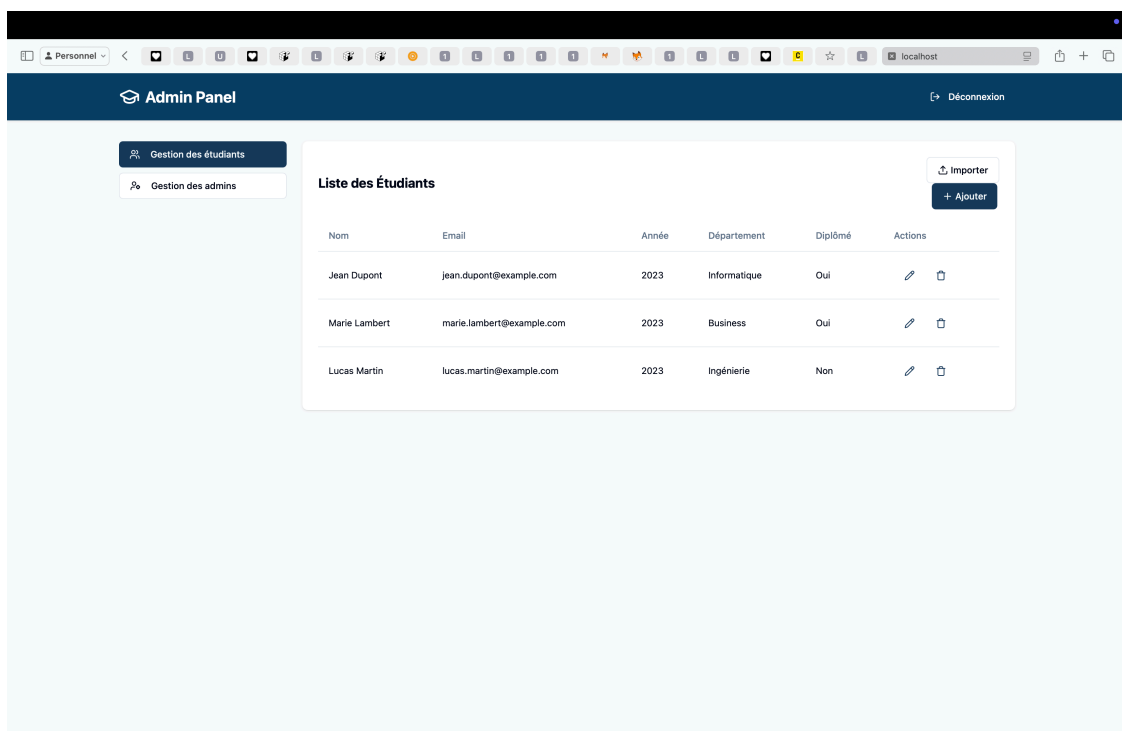
b. Interface Étudiant (après authentification)

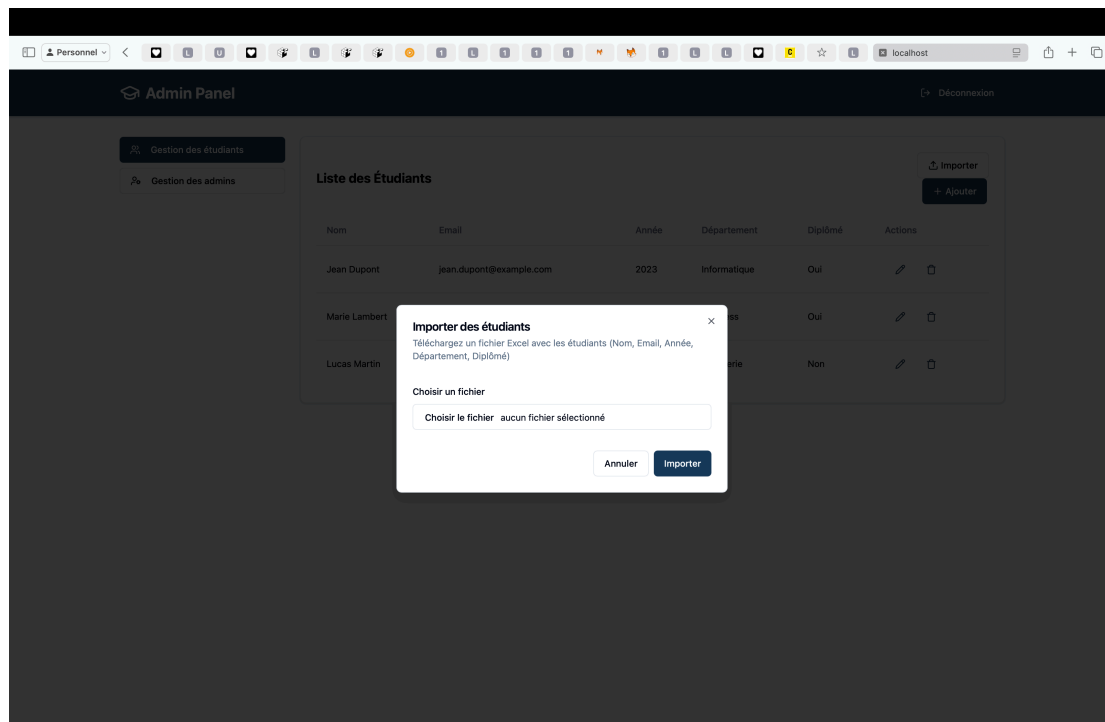
- Vue personnelle des documents officiels.
- Possibilité de copier ou transmettre le code unique à une tierce personne.
- **⚠ Capture d'écran à prévoir** : tableau de certificats avec boutons d'action.



c. Interface Admin (après authentification)

- Dashboard de gestion des promotions et étudiants.
- Importation via fichiers CSV ou Excel.
- Liste de tous les documents générés et possibilité de les valider.
- **⚠ Capture d'écran à prévoir** : tableau de bord admin, formulaire d'importation, formulaire d'ajout de fichiers (Excel ou CSV), interface de validation.





4. Services AWS spécifiques (À intégrer plus tard)

- **AWS Lambda** : pour la logique métier.
- **API Gateway** : endpoints REST.
- **AWS IAM** : gestion des permissions.

5. Authentification (via Microsoft Entra ID à venir)

Choix de la solution d'authentification

Afin de simplifier la gestion des accès tout en renforçant la sécurité, **le système d'authentification reposera sur Microsoft Entra ID** (anciennement Azure Active Directory), **puisque l'école dispose déjà d'une base d'identifiants professionnels et scolaires (emails et mots de passe)** pour tous les étudiants et personnels. Cela permet :

- **D'éviter une redondance** dans la création de comptes utilisateurs.
- De **s'appuyer sur une solution éprouvée, centralisée et sécurisée**.
- De **réduire la charge de développement liée à la gestion des comptes, mots de passe et MFA**.
- **D'intégrer directement les utilisateurs existants de l'école dans le système**.

Fonctionnalités prévues via Microsoft Entra ID

- Authentification unique (SSO) avec l'email scolaire.
- Accès différencié selon les rôles (étudiant, admin).
- Connexion sécurisée par le biais de l'identité Microsoft.
- Support possible pour l'authentification multifacteur.



Flux d'authentification prévu

1. L'utilisateur est redirigé vers Microsoft Entra.
2. Il se connecte avec son adresse email professionnelle ou scolaire.
3. Le système reçoit un token sécurisé validant son identité.
4. L'accès est ensuite accordé à l'espace étudiant ou admin.



Implémentation (à venir)

- Intégration avec NextAuth.js ou MSAL.js pour connecter Microsoft Entra avec Next.js.
- Paramétrage des scopes et des groupes dans Azure Portal pour définir les accès.
- Vérification du token côté serveur dans les API routes de Next.js.

6. Hébergement et Déploiement

Hébergement actuel

- En local, pas encore sur AWS ou autre hébergeur.

Déploiement CI/CD

- Non mis en place à ce stade.

Environnements

- **Développement** : local avec fausses données et fichiers CSV.
- **Production** : en attente d'approbation client.

7. Sécurité (Locale)

- Gestion des rôles via logique backend local.
- Protection basique avec JWT.
- SSL/TLS prévu pour la mise en production.

8. Monitoring et Optimisation

Monitoring actuel

- Logs en local (console ou fichiers).

Optimisation

- Prévue dans la version cloud (CloudWatch, Cost Explorer, etc.).

9. Tests

- **Tests unitaires** : sur logique backend (importation, validation, etc.).
- **Tests fonctionnels** : sur l'authentification et affichage des certificats.
- **Tests de charge** : pas encore réalisés.

10. Documentation et Livraison

- Guide utilisateur (administration locale).
- Documentation technique : structure du projet, endpoints locaux, format CSV.
- Livraison complète prévue après retour client.

11. Phase de Prototypage et Données Factices

- **Importation via Excel/CSV** : injecte une base étudiante fictive.

- **Génération de certificats simulés** : sans valeur légale mais représentatifs.
- **Fonctionnalités développées** : authentification, tableau de bord admin, vérification d'un code unique.
- **Décision stratégique** : arrêt temporaire du développement en attendant l'approbation officielle du client pour valider les besoins réels avant de passer à l'implémentation cloud (AWS).