



ZOOM EPREUVE E5

Yulian Guinand

Sommaire

Zoom 1 (Application C#)	3
Description d'une réalisation professionnelle	3
Zoom sur une fonctionnalité : Système de Commentaires & TODOS	5
A. Évolution de la modélisation (MCD)	5
B. Architecture logicielle et Couche Accès aux Données (DAL)	6
C. Interface Utilisateur (IHM) et Expérience Utilisateur.....	7
Zoom 2 (Application Laravel)	8
Description d'une réalisation professionnelle	8
Zoom sur une fonctionnalité : Système de Upload et création à la volé	10
A. Architecture et flux de données.....	10
B. Résolution des conflits "à la volée" (UX & Intégrité)	11
C. Implémentation technique Back-end (Sécurité et Performance)	12
D. Extension : Création dynamique hors import	12

Zoom 1 (Application C#)

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS		SESSION 2026
ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle (recto)		
Épreuve E5 - Conception et développement d'applications (option SLAM)		
DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation : 1
Nom, prénom : Guinand Yulian		N° candidat :
Épreuve ponctuelle <input type="checkbox"/>	Contrôle en cours de formation <input checked="" type="checkbox"/>	Date : / /
Organisation support de la réalisation professionnelle Lycée Pasteur Mont Roland (Groupes scolaires Mont Roland La Salle - Dole). Le commanditaire est M. Pernelle, organisateur de la journée annuelle "Santé Citoyenneté". L'établissement nécessite une application pour remplacer la gestion manuelle actuelle des interventions associatives, qui repose sur des échanges de courriels et des documents partagés. L'objectif pour l'organisation est de centraliser, automatiser et sécuriser le processus d'invitation et de suivi. Cela vise à réduire la charge administrative du professeur principal coordonnateur, à éviter les erreurs ou pertes d'informations, et à améliorer la coordination avec les associations partenaires grâce à une base de données unique.		
Intitulé de la réalisation professionnelle AP3 : Application de gestion des associations pour les journées Santé Citoyenneté.		
Période de réalisation : Du 18/09/2025 au 13/01/2026 Lieu : Dole. Modalité : <input type="checkbox"/> Seul(e) <input checked="" type="checkbox"/> En équipe		
Compétences travaillées <input checked="" type="checkbox"/> Concevoir et développer une solution applicative <input checked="" type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative <input checked="" type="checkbox"/> Gérer les données		
Conditions de réalisation¹ (ressources fournies, résultats attendus) M. Pernelle souhaitait centraliser, automatiser et sécuriser le processus d'invitation et de suivi des associations participant aux journées Santé Citoyenneté. Le résultat attendu est une architecture divisée en deux : une application lourde en C# pour l'administration et une interface web publique en React. Les échanges transitent par une API Laravel connectée à une base de données SQL Server.		
Description des ressources documentaires, matérielles et logicielles utilisées² <ul style="list-style-type: none">- Description des ressources documentaires, matérielles et logicielles utilisées- Gestion de projet et communication : Microsoft Teams.- Gestion du code source : GitHub (méthodologie GitFlow).- Environnement de développement : Visual Studio / Visual Studio Code.- Hébergement et BDD : Ubuntu Server, MySQL.- Frameworks et langages : C#, Laravel, React, Tailwind CSS, Vite.		

¹ En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

² Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

Modalités d'accès aux productions³ et à leur documentation⁴

	Accès au code	Accès a la production
C#	https://github.com/YulianGuinand/AssociationsManagerCSharp 	cloud.yulian-server.duckdns.org Utilisateur : public Mdp : eryjvYx0pE0k2GE 
Laravel	https://github.com/Mathis-mollier/AssociationManagerLaravel 	journee-santee.duckdns.org 

³ Conformément au référentiel du BTS SIO « Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve. ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

⁴ Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle
(verso, éventuellement pages suivantes)

Épreuve E5 - Conception et développement d'applications (option SLAM)

Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs

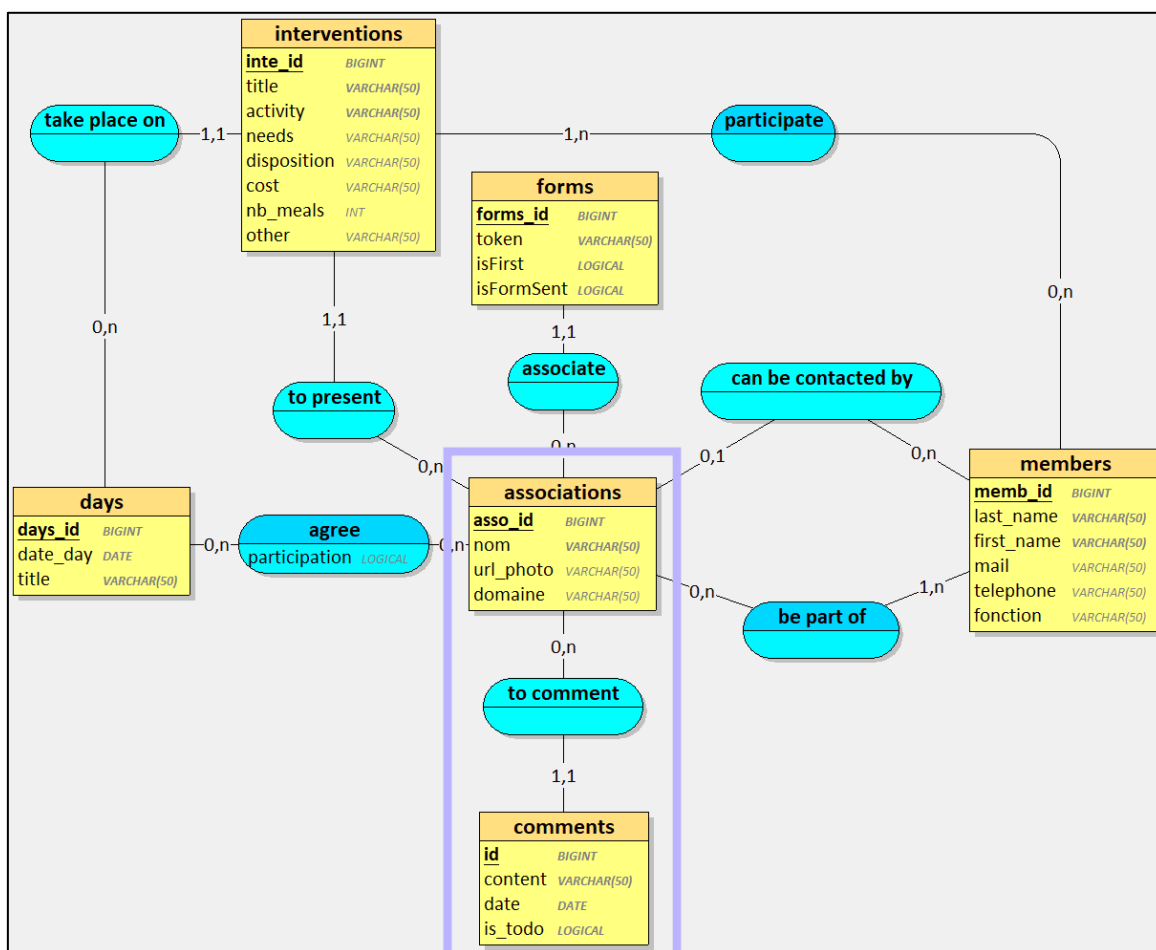
Zoom sur une fonctionnalité : Système de Commentaires & TODOS

Cette fonctionnalité a été développée suite à un besoin spécifique exprimé par Sébastien Pernelle. Lors des premiers retours sur l'application, il a souligné la nécessité de ne pas seulement stocker des notes informatives, mais de pouvoir transformer certaines d'entre elles en actions concrètes. J'ai donc conçu un système hybride permettant de gérer à la fois l'historique des échanges et le suivi des tâches opérationnelles.

A. Évolution de la modélisation (MCD)

Le schéma initial de la base de données ne prévoyait aucun stockage pour des notes ou des rappels. J'ai donc dû concevoir et intégrer une brique entièrement nouvelle.

L'évolution du **Modèle Conceptuel des Données** (MCD) a consisté à créer une entité **Comments** totalement absente de la première version. J'ai établi une relation (0,n) et (1,1) entre Associations et Comments car une associations peut avoir plusieurs commentaires alors qu'un commentaire n'existe que pour une seul association. L'innovation majeure réside dans l'introduction de l'attribut is_todo. Ce champ n'était pas prévu : il permet de transformer une simple donnée textuelle en une donnée métier actionnable, séparant les comptes-rendus passifs des tâches urgentes à traiter.



B. Architecture logicielle et Couche Accès aux Données (DAL)

Le développement a été réalisé en respectant une séparation stricte des couches pour garantir la maintenabilité de cette nouvelle brique.

1. Le Modèle (**Comments.cs**) : J'ai créé cette classe pour encapsuler les nouvelles données (id, contenu, date et le flag is_todo).

```
public class Comments
{
    private Int64 id;
    private string comments_content = null!;
    private DateTime comments_date;
    private Int64 comments_asso_id;
    private bool is_todo;
    2 références
    public Comments(string comments_content, DateTime comments_date, Int64 comments_asso_id, bool is_todo = false)
    {
        this.Comments_content = comments_content;
        this.Comments_date = comments_date;
        this.Comments_asso_id = comments_asso_id;
        this.Is_todo = is_todo;
    }
    9 références
    public Int64 Id { get => id; set => id = value; }
    7 références
    public string Comments_content { get => comments_content; set => comments_content = value; }
    10 références
    public DateTime Comments_date { get => comments_date; set => comments_date = value; }
    4 références
    public Int64 Comments_asso_id { get => comments_asso_id; set => comments_asso_id = value; }
    5 références
    public bool Is_todo { get => is_todo; set => is_todo = value; }
}
```

2. La DAL (**CommentsADO.cs**) : L'implémentation de l'accès aux données a été le point critique pour la sécurité. J'ai rédigé des méthodes utilisant des requêtes paramétrées (**SqlParameter**) pour neutraliser les injections SQL, une protection indispensable puisque l'utilisateur peut désormais saisir du texte libre.

- a. **Optimisation technique** : Pour rendre l'ajout de commentaire fluide, j'utilise **SCOPE_IDENTITY()** lors de l'**INSERT**. Cela permet de récupérer l'identifiant généré par **SQL Server** et de synchroniser l'objet local en temps réel sans recharger toute la liste.

```
public static bool InsertComment(Comments comment)
{
    // Validation : objet non null
    if (comment == null)
    {
        MessageBox.Show("Le commentaire fourni est nul.");
        return false;
    }

    // Requête INSERT avec récupération de l'ID auto-généré
    string query = @"
        INSERT INTO COMMENTS (content, date, asso_id, is_todo)
        VALUES (@Content, @Date, @AssoId, @IsTodo);
        SELECT CAST(SCOPE_IDENTITY() AS BIGINT);";

    // Paramètres sécurisés (empêche les injections SQL)
    SqlParameter[] parameters = new SqlParameter[]
    {
        new SqlParameter("@Content", comment.Comments_content),
        new SqlParameter("@Date", comment.Comments_date),
        new SqlParameter("@AssoId", comment.Comments_asso_id),
        new SqlParameter("@IsTodo", comment.Is_todo)
    };

    // ExecuteScalar retourne l'ID généré par SCOPE_IDENTITY()
    object? result = ExecuteScalarSafe(query, parameters, "Erreur lors de l'ajout du commentaire.");

    if (result is Int64 id && id > 0)
    {
        // Assignment de l'ID au commentaire pour permettre les opérations futures
        comment.Id = id;
        return true;
    }
    else
    {
        MessageBox.Show("Impossible de récupérer l'identifiant du commentaire nouvellement créé.");
        return false;
    }
}
```

C. Interface Utilisateur (IHM) et Expérience Utilisateur

L'interface a dû être repensée pour intégrer ces nouveaux flux de travail. J'ai conçu deux modales, **AddCommentDialog** et **EditCommentDialog**, qui n'existaient pas dans le prototype initial. Elles intègrent un Toggle Bouton permettant de définir si le commentaire est une simple note ou une tâche "À faire".

The image shows two modal dialogs side-by-side. The left one is titled 'AJOUTER UN COMMENTAIRE' and contains a form with a date field (14/03/2026), a text area for the comment, and a button 'Marquer comme tâche à faire'. The right one is titled 'MODIFIER UN COMMENTAIRE' and contains a text area with the comment 'Mail envoyer pour organisation Intervention à valider', a date field (12/12/2025), and a green button '✓ Marqué comme tâche à faire'. Both modals have 'ANNULER' and 'AJOUTER'/'ENREGISTRER' buttons at the bottom.

Pour assurer une réactivité maximale, j'ai utilisé une ObservableCollection. Ce choix technique permet un affichage "à chaud" : dès qu'un utilisateur valide une note ou marque une tâche comme terminée, l'interface graphique se met à jour instantanément sans "clignotement" de l'écran. Enfin, pour faciliter le pilotage, j'ai ajouté une méthode CountTodosByAsso qui alimente des badges de notification visuels sur le tableau de bord principal, permettant de repérer immédiatement les associations nécessitant une action prioritaire.

A screenshot of a 'Gérer les commentaires' section. It features a table with columns for 'COMMENTAIRE' and 'DATE'. A '+ AJOUTER UN COMMENTAIRE' button is in the top right. The table contains one row with the comment 'Mail envoyer pour organisation Intervention à valider' and the date '12/12/2025'. Action buttons 'Modifier' and 'Supprimer' are visible at the end of the row.

COMMENTAIRE	DATE
Mail envoyer pour organisation Intervention à valider	12/12/2025

A screenshot of a 'Tâches à faire' section. It has a search bar 'Rechercher dans les tâches...' and a '2' badge. Below are two task cards. The first card has a date '20/01/2026' and the text 'A appeler pour confirmer le thème et l'affinier', with a 'VOIR' button. The second card has a date '14/01/2026' and the text 'Voir mail pour réponse', also with a 'VOIR' button.

Zoom 2 (Application Laravel)

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS		SESSION 2026
ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle (recto)		
Épreuve E5 - Conception et développement d'applications (option SLAM)		
DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation : 1
Nom, prénom : Guinand Yulian		N° candidat :
Épreuve ponctuelle <input type="checkbox"/>	Contrôle en cours de formation <input checked="" type="checkbox"/>	Date : / /
Organisation support de la réalisation professionnelle Le projet Élan a été initié pour répondre aux exigences réglementaires du CFA Excellence Pro Franche-Comté. L'établissement doit justifier de 14 missions de service public, notamment le suivi de l'employabilité et de la satisfaction des apprentis, afin de maintenir ses financements et sa certification Qualiopi. Jusqu'alors, ces données étaient collectées via des outils disparates (Excel, Google Forms), entraînant une dispersion de l'information et un taux de réponse plafonnant à 30%. L'objectif était donc d'industrialiser ce processus par une solution web unique capable de gérer une collecte bimodale (e-mail et téléphone) tout en garantissant une conformité stricte au RGPD et aux normes d'accessibilité RGAA.		
Intitulé de la réalisation professionnelle Élan : Plateforme web de collecte de données et suivi d'employabilité (CFA Excellence Pro)		
Période de réalisation : Du 12/01/2026 à fin Avril 2026 Lieu : Dole.		
Modalité : <input type="checkbox"/> Seul(e) <input checked="" type="checkbox"/> En équipe		
Compétences travaillées <input checked="" type="checkbox"/> Concevoir et développer une solution applicative <input type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative <input checked="" type="checkbox"/> Gérer les données		
Conditions de réalisation⁵ (ressources fournies, résultats attendus) Ressources fournies : Cahier des charges fonctionnel et technique, liste des 14 missions réglementaires, fichiers d'exemples (Excel) des promotions d'apprentis, accès à un serveur de test. Résultats attendus : Une application web sécurisée permettant la gestion des enquêtes, une collecte bimodale (web/téléphone) performante, et la génération de rapports statistiques conformes aux exigences Qualiopi.		
Description des ressources documentaires, matérielles et logicielles utilisées⁶ <ul style="list-style-type: none">• Matérielles : Serveur Linux (Debian) hébergé chez OVHcloud, postes de développement locaux.• Logicielles : PHP 8.4, Laravel 12, MySQL, React, Tailwind CSS 4, Git/GitHub, VS Code.• Documentaires : Référentiel Qualiopi, documentation officielle Laravel, normes RGAA et RGPD.		

⁵ En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

⁶ Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

Modalités d'accès aux productions⁷ et à leur documentation⁸

	Accès au code	Accès a la production
Liens	https://github.com/YulianGuinand/elan	https://yulian-server.duckdns.org/
QR-Code		

- **Documentation** : Dossier technique et manuel utilisateur disponibles sur le dépôt GitHub.

⁷ Conformément au référentiel du BTS SIO « Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve. ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

⁸ Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle
(verso, éventuellement pages suivantes)

Épreuve E5 - Conception et développement d'applications (option SLAM)

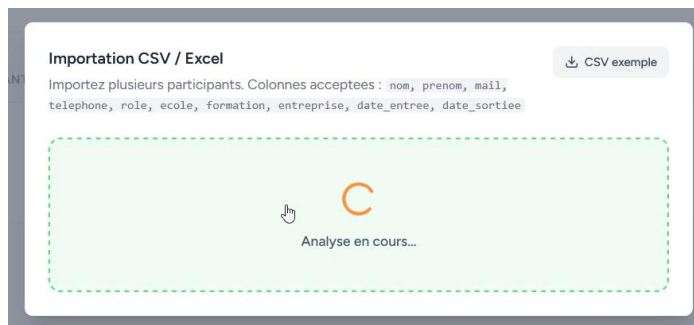
Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs
Zoom sur une fonctionnalité : Système de Upload et création à la volé

Le module d'importation des participants est un module critique du projet Élan. L'enjeu technique était de transformer des fichiers plats (Excel/CSV) en données relationnelles complexes tout en garantissant une base de données "propre". J'ai conçu un workflow en **deux étapes** : une **prévisualisation** avec détection de conflits et un **import final** sécurisé.

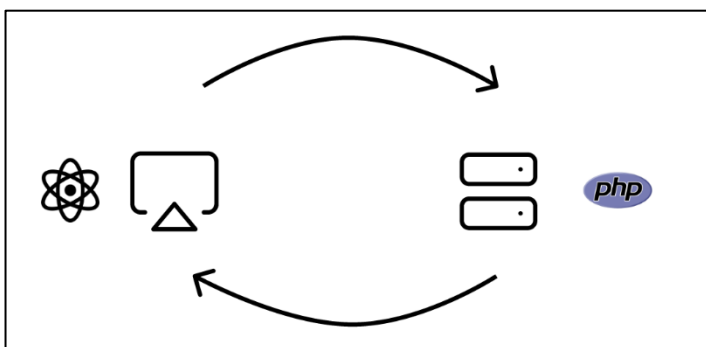
A. Architecture et flux de données

Le système repose sur une architecture découplée entre un **front-end réactif** (React/Inertia) et un **back-end** (Laravel 12).

1. **Phase d'analyse (Preview)** : Le composant StudentCsvImport.tsx transmet le fichier au ParticipantController. Le ExcelService normalise les données et compare les entités du fichier (Écoles, Formations, Entreprises) avec celles présentes en base de données.



2. **Gestion des conflits** : Le serveur retourne au front-end non seulement les lignes de données, mais aussi des listes d'entités inconnues (unknownEcoles, etc.).



Aperçu — 5 participant(s) Creez d'abord les entites manquantes ci-dessus

	RÔLE	ECOLE	FORMATION	ENTREPRISE
...@email.com	Apprenti	⊙ ENSIIE	BTS SIO SLAM	⊙ Tech Solutions SAS
...@mail.com	Apprenti	⊙ Lycée Turgot	BTS NDRC	Boutique Paris
...@email.com	Formateur	—	—	—
...@email.com	Apprenti	⊙ ENSIIE	Licence Pro	⊙ Agence Web Creativ
...@mail.com	Apprenti	⊙ École du Web	Master Dev	⊙ Google France

```

1 {
2   "rows": [
3     {
4       "nom": "Durand",
5       "prenom": "Sophie",
6       "mail": "sophie.durand@email.com",
7       "telephone": "0601020304",
8       "role": "Apprenti",
9       "ecole": "ENSIIE",
10      "formation": "BTS SIO SLAM",
11      "entreprise": "Tech Solutions SAS",
12      "date_entree": "2024-09-01",
13      "date_sortiee": "2026-06-30"
14    }
15    // ...
16  ],
17  "unknownEcoles": [
18    "ENSIIE",
19    "Lycée Turgot",
20    // ...
21  ],
22  "unknownFormations": [
23    "BTS SIO SLAM",
24    // ...
25  ],
26  "unknownEntreprises": [
27    "Tech Solutions SAS",
28    // ...
29  ]
30 }
31

```

B. Résolution des conflits "à la volée" (UX & Intégrité)

```
1 /**
2  * Helper to resolve new vs existing entity.
3  */
4  private function resolveInlineCreation($modelClass, $field, $value)
5  {
6      if (str_starts_with($value, 'NEW_')) {
7          $newName = substr($value, 4);
8          $entity = $modelClass::create([$field => $newName]);
9          return $entity->id;
10     }
11     return $value;
12 }
```

Pour offrir une expérience fluide à l'utilisateur, j'ai implémenté une logique de résolution interactive. Si le système détecte une entreprise inconnue dans le fichier Excel, le bouton d'import final est désactivé (`disabled={hasUnresolved}`).

L'utilisateur doit alors utiliser la modale `InlineCreateModal.tsx`. Cette dernière communique avec le `InlineCreateController` pour créer réellement l'entité manquante en base de données. Une fois l'ID récupéré, le front-end marque le conflit comme résolu (`resolvedEntreprises`). Ce mécanisme garantit qu'aucun participant ne sera importé sans être rattaché à une structure valide, respectant ainsi les contraintes d'intégrité référentielle du modèle de données.

Entités inconnues détectées — créez-les avant d'importer

ecoles

- ENSIE
- Lycée Turgot
- École du Web

Formations

- BTS SIO SLAM
- BTS NDRC
- Licence Pro
- Master Dev

Entreprises

- Tech Solutions SAS
- Agence Web Creativ
- Google France

Aperçu — 5 participant(s) Creez d'abord les entites manquantes en-dessus

NOM	PRENOM	MAIL	RÔLE	ECOLE	FC
Durand	Sophie	sophie.durand@email.com	Apprenti	ENSIE	
Martin	Lucas	lucas.m@email.com	Apprenti	Lycée Turgot	
Lefebvre	Marie	m.lefebvre@email.com	Formateur		
Dubois	Thomas	thomas.dubois@email.com	Apprenti	ENSIE	
Roux	Amélie	amelie.r@email.com	Apprenti	École du Web	

ⓘ Creez d'abord les entites manquantes

Nouvelle entreprise ✕

Raison sociale *

Email *

Téléphone Ville

Interlocuteur

C. Implémentation technique Back-end (Sécurité et Performance)

Le traitement des données côté serveur a été optimisé pour gérer des volumes importants :

Normalisation (ExcelService.php) : J'ai développé une méthode de mapping flexible capable de reconnaître différentes têtes de colonnes (ex: "mail", "Email", "Email Address") afin de rendre l'outil plus tolérant aux variations des fichiers sources.

```
1 'mail' => $line['mail'] ?? $line['Mail'] ?? $line['Email'] ?? $line['email'] ?? null,
```

Déduplication et Idempotence : Lors de l'import final (importCsv), j'utilise la méthode firstOrCreate sur le modèle Participant en prenant l'adresse email comme clé unique. Cela permet de relancer un import plusieurs fois sans créer de doublons.

Transactions et Liens : Pour chaque ligne, le système crée ou récupère le participant, puis génère un Contrat faisant le lien entre le participant, son école, sa formation et son entreprise. L'utilisation de firstOrCreate pour les entités liées assure qu'aucune donnée n'est orpheline.

```
1 foreach ($donnees as $row) {
2     // Créer ou trouver le participant
3     $participant = Participant::firstOrCreate(
4         ['mail' => $row['mail']],
5         [
6             'nom'         => $row['nom'],
7             'prenom'      => $row['prenom'],
8             'telephone'   => $row['telephone'] ?? null,
9             'role'        => $row['role'] ?? 'Apprenti',
10        ]
11    );
```

D. Extension : Création dynamique hors import

En complément de l'import massif, j'ai intégré une convention de nommage (NEW_) dans le formulaire de création manuelle. La méthode resolveInlineCreation détecte ce préfixe pour créer automatiquement une nouvelle école ou formation si l'utilisateur saisit une valeur inexistante dans la liste déroulante. Cette cohérence fonctionnelle assure que l'utilisateur n'est jamais bloqué dans son workflow de saisie.