



DOSSIER PROFESSIONNEL (DP)

Nom de naissance ▶ ALOESODE
Nom d'usage ▶ ALOESODE
Prénom ▶ Alexandre
Adresse ▶ 2 Impasse Sainte-Félicie, 13004 Marseille

Titre professionnel visé

Concepteur Développeur d'Applications

MODALITÉ D'ACCÈS :

- Parcours de formation
- Validation des Acquis de l'Expérience (VAE)

DOSSIER PROFESSIONNEL (DP)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.
Ce titre est délivré par le Ministère chargé de l'emploi.

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel (DP)** dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- une déclaration sur l'honneur à compléter et à signer ;
- des documents illustrant la pratique professionnelle du candidat (facultatif)
- des annexes, si nécessaire.

DOSSIER PROFESSIONNEL (DP)

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.

 <http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

Développer une application sécurisée	p.	5	
- Développement d'un système de gestion de calendriers scolaires	p.	p.	5
- Développement d'un système de gestion des absences	p.	p.	9
Concevoir et développer une application sécurisée organisée en couches	p.	14	
- Développement d'un système de gestion de calendriers scolaires	p.	p.	14
- Développement d'un système de gestion des absences	p.	p.	18
- Migration de l'API de l'école La Plateforme	p	p.	21
Préparer le déploiement d'une application sécurisée	p.	31	
- Déploiement et hébergement des outils de La Plateforme	p.	p.	31
- Création d'une application mobile d'intranet étudiant	p.		39
Déclaration sur l'honneur	p.	51	

DOSSIER PROFESSIONNEL ^(DP)

EXEMPLES DE PRATIQUE PROFESSIONNELLE

DOSSIER PROFESSIONNEL (DP)

Activité-type 1 Développer une application sécurisée

Exemple n°1 - Développement d'un système de gestion de calendriers scolaires

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai effectué mes 2 années d'alternance au sein de l'Atelier, une structure rattachée à l'école La Plateforme, qui réalise divers projets informatiques d'envergures différentes pour des entreprises. L'un des principaux clients de l'Atelier n'est autre que l'école La Plateforme, et j'ai travaillé tout au long de mon alternance sur les outils internes de l'école, à savoir l'intranet côté étudiants et l'intranet côté administration.

Nous avons 3 étapes de développement, à savoir:

- La phase *développement* pendant laquelle nous travaillons sur nos ordinateurs dans un environnement local
- La phase *Preprod* qui est une copie en ligne des outils de l'école et permet de faire des tests
- La phase *Production*

Tous les outils de l'école sont matérialisés sous la forme de repositories dans GitHub. Ils sont réunis dans un repository global, nommé stack-plateforme-interne

Dossier personnel / atelier / stack-plateforme-interne			
Nom	Taille	Dernière modification	
API-LaPlateforme	16 éléments	30 juin 2025	☆
api-tests	5 éléments	1 avril 2025	☆
atlas-interne	3 éléments	1 avril 2025	☆
Authentication	7 éléments	26 juin 2025	☆
Intranet-ADM-2	8 éléments	30 juin 2025	☆
Intranet-Etudiant-2	7 éléments	30 juin 2025	☆
laplateforme-api-partners	5 éléments	10 avril 2025	☆
plateforme-connect	0 élément	14 avril 2025	☆
plateforme-soft-skill	0 élément	17 avril 2025	☆
docker-compose.yaml	10,0 Ko	9 juin 2025	☆
README.md	391 octets	17 avril 2025	☆

En environnement local, l'ensemble des outils est conteneurisé.

DOSSIER PROFESSIONNEL (DP)

L'une des missions qui m'ont été confiées fût de développer une interface de gestion des calendriers par promotion, avec les 4 principales fonctionnalités suivantes:

- Système d'import de calendriers via un fichier CSV

- Système de filtre et d'affichage des calendriers par promotions

- Système de navigation mois par mois

- Possibilité de modifier les informations stockées en base de données via l'interface front-end

Certaines promotions ayant un programme réalisé en alternance, l'outil devait donc être capable de différencier les jours en entreprise des jours en centre de formation. Ci-dessous une capture d'écran de l'outil mis en production.

The screenshot shows the 'La Plateforme' application interface. On the left is a sidebar with various navigation icons and labels: Home, Compétences, Catégories, Tâches, Projets, Etudiants, Units, Calendriers, Promotions, Sections, Alternances, Tutorat, and Utilisateurs. The main area has a blue header bar with 'La Plateforme' logo, 'Édition' switch, 'Dashboard', and user info 'Bonjour, Alexandre Aloesode.Pro'. Below the header are two main sections: 'Le calendrier' on the left and 'Liste des calendriers (19)' on the right. The 'Le calendrier' section shows a monthly calendar for July 2025 with days color-coded: teal for 'Ecole', orange for 'Entreprise', and red for 'Férié'. It includes buttons for 'Sauver' and 'Restaurer'. The 'Liste des calendriers' section displays a table with columns 'Nom' and 'Statut' for 19 entries, all marked as 'Actif'. A 'Filtrer' search bar is at the top of this section. At the bottom right is a 'Import de calendriers' button.

Une fois la base de données et le back-end fonctionnels, je me suis attaqué au front-end, en commençant par les requêtes en GET permettant de récupérer et afficher la liste des calendriers, puis le détail des jours une fois que l'on clique sur un calendrier.

DOSSIER PROFESSIONNEL (DP)

```
1 class CalendarList extends Component {
2   ready = () => {
3     this.connect("calendar_imported", () => {
4       });
5
6     `#${this.name}_import_button`).on("click", () => {
7       this.drawPopupUploadCalendar();
8       this.emit("loaded");
9     });
10   };
11
12   loadCalendars = () => {
13     Network.get({
14       route: "/calendar",
15       params: {
16         id: "",
17         name: this.datasources.filters.filter_name,
18         status: `#${this.name}_checkbox_dropdown_units_inactives`).prop("checked") ? "" : 1,
19         order: "name",
20       },
21       success: (data) => {
22         this.datasources.calendars = JSON.parse(data);
23         this.populateCalendars();
24       },
25       failure: (data) => {
26         if (data.responseJSON && data.responseJSON.message) {
27           this.emit("error", data.responseJSON.message);
28         } else {
29           console.error("Failure");
30         }
31       },
32     });
33   };
34 };
35 
```

Une fois l'affichage fonctionnel, j'ai intégré la possibilité, lorsqu'on clique sur un jour en particulier, de modifier le type de ce dernier, puis j'ai mis en place le système de navigation mois par mois.

Pour finir, j'ai implémenté un système d'import de calendrier permettant, via un fichier csv, de créer un calendrier, puis de stocker tous les jours renseignés en base de données.

DOSSIER PROFESSIONNEL (DP)

The screenshot shows the 'La Plateforme' software interface. On the left, there's a sidebar with various icons for Home, Compétences, Catégories, Titres, Projets, Étudiants, Units, Calendriers, Promotions, Sections, Alternances, Tutorat, and Utilisateurs. The main area has tabs for 'Le calendrier' and 'Marseille - M2 DPO - 2024'. A modal window titled 'Import de calendriers' is open, showing a file input field with 'Choisir un fichier' and 'Aucun fichier choisi', a 'Télécharger un template' button, and an 'Envoyer' button. To the right, there's a 'Filtrer' section with a search bar and 'Nom' dropdown, and a table titled 'Liste des calendriers (19)' with columns for 'Nom', 'Statut', and a 'Martigues - B2 Web - 2024/25' row. At the bottom right of the main area, there's a 'Import de calendriers' button.

2. Précisez les moyens utilisés :

- VSCode pour l'écriture du code
- GitHub Projects pour la rédaction des tickets et l'organisation du travail
- GitHub
- Docker

Le back-end était à l'époque en CodeIgniter, une légère surcouche du langage PHP.
Le front-end est un mélange de PHP, Javascript et JQuery.

Documentation utilisée:

- <https://www.php.net/docs.php>
- <https://jquery.com/>

3. Avec qui avez-vous travaillé ?

J'ai travaillé en collaboration avec Terry CRISTINELLI, directeur informatique de l'école pour la définition des besoins métier.
J'ai ensuite réalisé cette mission de manière indépendante, jusqu'aux pull requests validées par le CEO de l'Atelier, Julien ATHOMAS.

DOSSIER PROFESSIONNEL (DP)

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **L'ATELIER DE LA PLATEFORME**

Chantier, atelier, service ▶ **Contrat de Développeur Web en alternance**

Période d'exercice ▶ Du : **01/09/2023** au : **31/08/2025**

5. Informations complémentaires (facultatif)

Activité-type 1 Développer une application sécurisée

Exemple n°2 ▶ Développement d'un système de gestion des absences

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Ce projet a également été réalisé dans le cadre de ma mission de développement des outils internes de l'école La Plateforme.

A l'époque, les demandes d'absence des étudiants étaient envoyées par mail, et l'école n'avait pas de système permettant de stocker ces informations en base de données. J'ai donc été chargé de développer un système permettant de:

- Faire une demande d'absence côté intranet étudiants
- Afficher, puis valider ou refuser une demande d'absence côté intranet administration
- Stocker en base de données les informations relatives à une demande d'absence
- Stocker sur un Google Drive les pièces jointes contenant les justificatifs.

Ci-dessous une capture de l'outil côté administration.

DOSSIER PROFESSIONNEL (DP)

The screenshot shows the La Plateforme application interface. On the left, there is a sidebar with various icons for Home, Compétences, Catégories, Titres, Projets, Étudiants, Units, Calendriers, Promotions, Sections, Alternances, Tutorat, and Utilisateurs. The main content area has two sections: "Absences (2238)" and "Alertes (187)".

Absences (2238): This section displays a table of student absences. The columns are Etudiant, Promotion, Début, Durée, Justif, Statut, and Action. The table lists 15 entries, such as Raphael Attias, Jean-Alexandre Galeazzi, and Sullivan Moreau, each with their respective details and validation status.

Alertes (187): This section displays a table of alerts. The columns are Etudiant, Promotion, Date, Action, Statut, and Action. The table lists 18 entries, such as Sarah Krouche, Salim Ababulu, and Lucas Ribard, each with their respective details and status.

Ci-dessous une capture de la partie étudiants

This screenshot shows the student section of the La Plateforme application. It includes a form for submitting absences and a table of historical absences.

Absences: This section allows students to report an absence. It requires a Date de début (start date) and a Date de retour (return date). A dropdown menu for Motif (Reason) shows "Accident de transport". There is also a file upload field labeled "Choisir un fichier" and a message "Aucun fichier choisi". A blue "Envoyer" button is present.

Période **Durée** **Statut** **Commentaire**

01/03/2025 - 02/04/2025	1 jour	Validée	
26/08/2024 - 29/08/2024	3 jours	Validée	
02/09/2024 - 06/09/2024	4 jours	Validée	
04/12/2024 - 12/12/2024	6 jours	Refusée	

DOSSIER PROFESSIONNEL (DP)

Après avoir défini les besoins, créé ma table puis mon CRUD, j'ai attaqué le développement du front-end côté étudiants.

Via un formulaire , l'étudiant doit renseigner une date de début, une date de retour, un motif, et importer un justificatif. La durée totale est calculée automatiquement.

L'import de fichiers est un élément particulièrement sensible, puisqu'il expose l'API et peut éventuellement permettre à des utilisateurs malveillants d'importer des fichiers ou scripts malveillants, de faire des attaques par content-type, etc...

Afin de sécuriser cette étape, j'ai d'abord inséré une vérification du type de fichier côté front-end.

```
postAbsence = () => {
  if (
    $('#${this.name}_start_date').val() == "" ||
    $('#${this.name}_end_date').val() == "" ||
    $('#${this.name}_absence_reason').val() == "" ||
    $('#${this.name}_image').val() == ""
  ) {
    Utility.showToast("Veuillez remplir tous les champs", IconManager.get("gray-cross", { height: "36", width: "36", color: "red" }), 1600, "var(--error)");
    return;
  }

  const acceptedTypes = ["image/png", "image/jpeg", "image/jpg", "application/pdf"];
  let file = document.getElementById(`#${this.name}_image`).files[0];

  if (!acceptedTypes.includes(file.type) && file.name.split(".").pop() != "HEIC") [
    Utility.showToast(
      "Veuillez choisir un fichier image ou pdf",
      IconManager.get("gray-cross", { height: "36", width: "36", color: "red" }),
      1600,
      "var(--error)"
    );
    return;
  ]
}
```

J'ai ensuite rajouté une validation supplémentaire côté back-end

```
$acceptedTypes = ["image/png", "image/jpeg", "image/jpg", "application/pdf"];
if (!in_array($params['fileType'], $acceptedTypes)) return ($this->Status->PreconditionFailed());
```

DOSSIER PROFESSIONNEL (DP)

La dernière étape était de développer le composant front-end permettant à l'administration de gérer les absences avec:

- Système de filtre par statut et affichage par défaut, au chargement de la page, des absences en attente de traitement
- Système de filtre par dates, et affichage par défaut de l'année scolaire en cours

La Plateforme Édition Dashboard

Absences (2238)

Validées	Statut	Durée	Justif	Statut	Action
Validées	En attente	4 jours	Lien	Validée par marion boyer	
Refusées	Kapnaer Alnas Cyber - 2024/25 04-08-2025	5 jours	Lien	Validée par marion boyer	
Jean-Alexandre Galeazzi	Marseille - B2 Cyber - 2024/25	04-08-2025	Lien	Validée par marion boyer	
Sullivan Moreau	Marseille - B2 IA - 2024/25	04-08-2025	Lien	Validée par marion boyer	
William Rauwens-Olivier	Marseille - B2 Web - 2024/25	04-08-2025	Lien	Validée par marion boyer	
Kamelia Mohamdi	Marseille - B2 JVSI - 2024/25	15-07-2025	Lien	Validée par marion boyer	
Abakar-Issa Ali	Marseille - B2 JVSI - 2024/25	15-07-2025	Lien	Validée par marion boyer	
Salvatore Anania	Marseille - B3 Log - 2023/24	15-07-2025	Lien	Validée par marion boyer	
Mounir Merzoud	Marseille - B2 Log - 2024/25	15-07-2025	Lien	Validée par marion boyer	
Yoel Settbon	Marseille - B1 Prepa - 2024	11-07-2025	Lien	Validée par marion boyer	
Lucas Ricart	Marseille - B1 Prepa - 2024	11-07-2025	Lien	Validée par marion boyer	
Viktoria Naidiuk	Marseille - B1 Prepa - 2024	11-07-2025	Lien	Validée par marion boyer	
Amandine Satin	Cannes - B2 Cyber - 2024/25	11-07-2025	Lien	Validée par marion boyer	
Hajer Ben-Salah	Cannes - CDPI - 2025	11-07-2025	Lien	Validée par marion boyer	

DOSSIER PROFESSIONNEL (DP)

La partie front-end permet également, au clique sur une absence, d'afficher une pop-up permettant d'avoir les détails de l'absence, ainsi qu'une e-frame affichant le fichier justificatif stocké dans Google Drive. Cette pop-up permet enfin de modifier l'absence si besoin, puis de la valider ou refuser. Pour des raisons de protection des données, je ne suis pas en mesure de fournir une capture d'écran de cette pop-up.

2. Précisez les moyens utilisés :

- VSCode pour l'écriture du code
- GitHub Projects pour la rédaction des tickets et l'organisation du travail
- GitHub
- Docker

Le back-end était à l'époque en CodeIgniter, une légère surcouche du langage PHP.
Le front-end est un mélange de PHP, Javascript et JQuery.

Documentation utilisée:

- <https://www.php.net/docs.php>
- <https://jquery.com/>

3. Avec qui avez-vous travaillé ?

J'ai travaillé en collaboration avec Terry CRISTINELLI, directeur informatique de l'école pour la définition des besoins métier.

J'ai ensuite réalisé cette mission de manière indépendante, jusqu'aux pull requests validées par le CEO de l' Atelier, Julien ATHOMAS.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **L'ATELIER DE LA PLATEFORME**

Chantier, atelier, service ▶ **Contrat de Développeur Web en alternance**

Période d'exercice ▶ Du : **01/09/2023** au : **31/08/2025**

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Activité-type 2 Concevoir et développer une application sécurisée organisée en couches

Exemple n° 1 - Développement d'un système de gestion de calendriers scolaires

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Au sein de l'Atelier, l'organisation du travail s'inspire de la méthode Agile. En effet, nous planifions généralement au moins 2 fois par mois des réunions avec le Directeur Informatique de l'école, afin de discuter de l'état actuel des outils et des nouveaux projets.

Chaque besoin est ensuite traduit sous forme de tickets à l'aide de GitHub Projects. Ci-dessous une capture d'écran à l'instant T du travail accompli et restant.

The screenshot shows a GitHub Projects board for the 'Intra' project. The board is organized into five columns: Backlog, Ready, In progress, In review - équipe, and Done - a review client. Each column contains several items represented as cards, each with a title, a progress bar, and a small icon. The 'Backlog' column has 6 items, 'Ready' has 3, 'In progress' has 3, 'In review - équipe' has 35, and 'Done' has 11. The 'In review - équipe' column is currently the most active, containing many items related to API development and documentation. The 'Done' column contains items related to client reviews and merges. The 'Backlog' column includes items like 'Intranet-ADM-2 #321 Intra-ADM > Etudiant > Widget Anomalies'. The 'Ready' column includes 'Intranet-ADM-2 #318 Script avertissement : prise en charge activités'. The 'In progress' column includes 'API-LaPlateforme #229 Import étudiants - Bug'. The 'In review - équipe' column includes 'API-LaPlateforme #229 Import étudiants - Bug'. The 'Done' column includes 'API-LaPlateforme #222 API > error messages'.

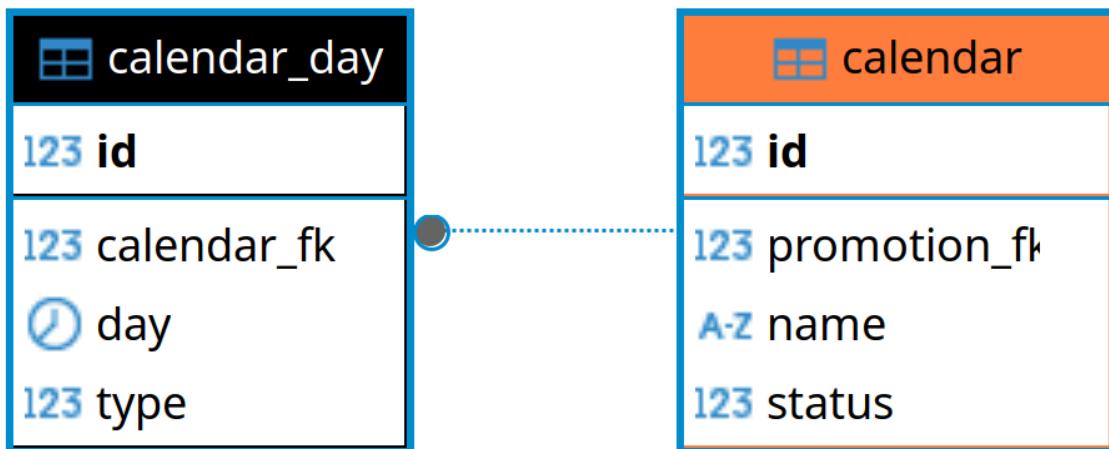
La première étape, "Backlog", sert à créer des *issues* pour des tâches imaginées mais pas encore discutées. Lors des réunions précédemment citées, nous transformons ces idées en véritables tickets, qui passent ensuite en étape "Ready". Les tickets sur lesquels nous travaillons sont ensuite passés dans l'étape "In progress". Lorsque le travail est terminé et la pull request est faite vers la branche master, nous passons le ticket à l'étape "In review - équipe". C'est alors au CEO de l'Atelier de consulter la pull request; s'il la valide et passe le ticket à l'étape "Done - à review client", cela veut donc dire que le travail est désormais disponible sur la Preprod, afin d'être testé dans des conditions réelles. Enfin, lors des réunions appelées "sprint reviews" avec le Directeur Informatique de l'école, nous passons en revue tous les tickets sur la Preprod; si les tests sont concluants, le ticket passe à son étape finale, à savoir "Go"; le travail fait est donc désormais en production.

DOSSIER PROFESSIONNEL (DP)

Dans le cadre de ma mission consistant à réaliser un système de gestion de calendriers, la première étape fût de conceptualiser cet outil côté base de données, avec les contraintes suivantes:

- Chaque calendrier doit être lié à une promotion via une clé étrangère
- Chaque jour de présence en entreprise ou à l'école doit être stocké et différencié
- Le calendrier doit pouvoir passer en statut "inactif" une fois l'année scolaire terminée

Ci-dessous, un aperçu du diagramme de données.



J'ai donc créé deux tables:

- La table **calendar**, permettant de faire le lien avec la promotion via une clé étrangère, de stocker le nom du calendrier, et enfin le statut "actif" ou "inactif"
- La table **calendar_days**, permettant d'identifier le calendrier concerné via une clé étrangère, et enfin de stocker chaque jour de type 1 (école) ou 2 (entreprise)

Pour finir, j'avais besoin de vider également la table **calendar_days** à la suppression d'un calendrier; j'ai donc ajouté une suppression en cascade.

L'interface MySQL Workbench montre la configuration de la table **calendar_day** :

- Nom de la table: **calendar_day**
- Moteur: **InnoDB**
- Auto-Incrémation: **8983**
- Encodage: **utf8mb3**
- Collation: **utf8mb3_general_ci**
- Description: (vide)

À la base de la page, une section détaille les contraintes :

Colonnes	Nom	Colonne	Propriétaire	Référence de la table	Type	Référence de l'objet	A la suppression	A la mise à jour
Contraintes	calendar_day_FK	calendar_day	calendar	calendar	FOREIGN KEY	PRIMARY	Cascade	Restrict
Clefs étrangères	123_calendar_fk	123_calendar_fk	—	—	—	id	—	—

DOSSIER PROFESSIONNEL (DP)

Suite à la phase de modélisation, j'ai commencé à coder la partie back-end en réalisant mon CRUD (Create, Read, Update, Delete, associés aux requêtes POST, GET, PUT, DELETE)

```
class Calendar_Model extends LPTF_Model
{
    protected $api_helper;
    protected $protectFields = false;
    protected $stable = 'calendar';

    public function __construct()
    {
        parent::__construct();

        $this->api_helper = new \App\Libraries\APIHelper($this->db, $this->table);
    }

    public function getCalendar($params)
    {
        $constraints = [
            ['id', 'optional', 'number', true],
            ['promotion_id', 'optional', 'number', true],
            ['name', 'optional', 'string', true],
            ['status', 'optional', 'number', true]
        ];
        if ($this->api_helper->checkParameters($params, $constraints) == false) {
            return ($this->Status->PreconditionFailed());
        }

        $fields = $this->getCalendarFields();
        $this->api_helper->buildGet($params, $fields);
        $this->api_helper->addLimitAndOffset($params);

        return ($this->api_helper->getBuilder()->get()->getResultArray());
    }
}

You, 20 months ago * CRUD calendar
public function postCalendar($params)
{
    $constraints = [
        ['promotion_id', 'mandatory', 'number'],
        ['name', 'mandatory', 'string'],
        ['status', 'mandatory', 'number']
    ];
    if ($this->api_helper->checkParameters($params, $constraints) == false) {
        return ($this->Status->PreconditionFailed());
    }

    $data = [
        'promotion_fk' => $params['promotion_id'],
        'name' => $params['name'],
        'status' => $params['status'],
    ];

    try {
        $this->insert($data);
        if($this->db->error()['code'] != 0) {
            return $this->Status->DatabaseError($this->db->error());
        }
        return $this->getInsertID();
    } catch (\Exception $e) {
        return $this->Status->DatabaseError($this->db->error());
    }
}
```

DOSSIER PROFESSIONNEL (DP)

La sécurisation de l'accès aux données est gérée via un système de role_helpers. En effet, chaque requête envoyée à l'API contient un token qui, une fois déchiffré, renseigne l'id_user, l'email et le rôle de l'utilisateur faisant la requête. Ainsi on peut, comme le montre l'exemple ci-dessous, définir l'accès au CRUD des différentes tables pour chaque rôle, en l'occurrence ici le rôle CRM (administration).

```
<?php

namespace App\Libraries\AccessControl;

You, 1 second ago | 3 authors (You and others)
class CrmAccess
{
    private $controller;

    public function __construct(&$controller)
    {
        $this->controller = &$controller;
    }

    public function Access($call, $payload, &$params)
    {
        $granted = [
            'getAbsence',
            'postAbsence',
            'putAbsence',

            'getActivity',
            'getActivityAttendance',

            'getAlternance',
            'putAlternance',
            'postAlternance',

            'getAlert',
            'putAlert',

            'getApplicant',
            'postApplicant',
            'putApplicant',
            'deleteApplicant',

            'getCalendar',
            'postCalendar', | You, 1 second ago • Uncommitted changes

            'getCalendarDay',
            'postCalendarDay',
            'putCalendarDay',
            'deleteCalendarDay',

            'getCalendarHistory',

            'getFollowup',
            'getStudentFollowup',

            'postInvoice',
            'getInvoiceCount',

            'getJob',

            'getLogtime',
            'getLastLog',

            'getRegistration',
        ];
    }
}
```

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

- IDE: VSCode
- DBeaver pour l'affichage et la gestion des bases de données
- Langage SQL pour les bases de données

3. Avec qui avez-vous travaillé ?

J'ai réalisé cette mission de manière indépendante, jusqu'aux migrations de la base de données, faites par le CEO de l' Atelier, Julien ATHOMAS.

4. Contexte

Nom de l'entreprise, organisme ou association ➔ **L'ATELIER DE LA PLATEFORME**

Chantier, atelier, service ➔ **Contrat de Développeur Web en alternance**

Période d'exercice ➔ Du : **01/09/2023** au : **31/08/2025**

5. Informations complémentaires (*facultatif*)

Activité-type 2 Concevoir et développer une application sécurisée organisée en couches

Exemple n° 2 ➔ Développement d'un système de gestion des absences

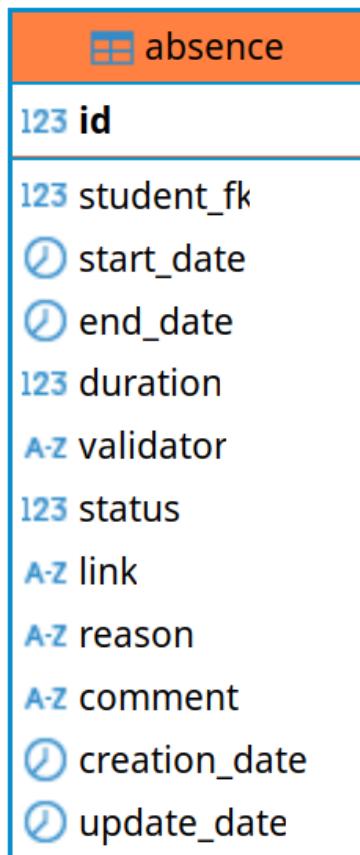
1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Après avoir défini les besoins, j'ai commencé par faire ma table en base de données, en prenant en compte les principales contraintes suivantes:

- mise en relation de l'absence avec un étudiant, via une clé étrangère
- date de début et de fin de l'absence
- durée de l'absence
- Si absence traitée, enregistrement de l'email de la personne ayant validé/refusé l'absence
- Statut de l'absence (0 = en cours, 1 = validée, 2 = refusée)

DOSSIER PROFESSIONNEL (DP)

-Lien Google Drive du justificatif



Le plus gros challenge de cette mission fût de mettre en place dans l'API un système permettant de stocker le justificatif sur Google Drive, puis de récupérer l'url générée pour la stocker en base de données.

```
class Absence_Model extends LPTF_Model
public function postAbsence($params)
{
    $GDrive = new \App\Libraries\GoogleDriveHelper();

    $upload = $GDrive->uploadBasic($data, $params['imageName'], $params['fileType'], GOOGLE_DRIVE_ABSENCE_FOLDER);
    if (!$upload) return ($this->Status->Error());
}
```

J'ai donc créé une classe permettant de récupérer les identifiants Google d'un fichier de constantes, puis de faire l'upload

DOSSIER PROFESSIONNEL (DP)

```
You, 2 months ago | 1 author (You)
<?php

namespace App\Libraries;

use Google\Client;
use Google\Service\Drive;

You, 2 months ago | 1 author (You)
class GoogleDriveHelper
{
    public function uploadBasic($fileToUpload, $fileName, $fileType, $folder)
    {
        try {
            $client = new Client();
            $client->setAuthConfig(array(
                "type" => "service_account",
                "client_id" => GOOGLE_CLIENT_ID,
                "client_email" => GOOGLE_CLIENT_EMAIL,
                "private_key" => GOOGLE_PRIVATE_KEY,
                "signing_algorithm" => "HS256"
            ));

            $client->useApplicationDefaultCredentials();
            $client->addScope(Drive::DRIVE);
            $driveService = new Drive($client);
            $fileMetadata = new Drive\DriveFile(array(
                'name' => $fileName,
                'parents' => array($folder),
            ));

            $file1 = $driveService->files->create($fileMetadata, array(
                'data' => $fileToUpload,
                'mimeType' => $fileType,
                'uploadType' => 'media',
                'fields' => 'id,webViewLink,webContentLink,thumbnailLink',
                'supportsAllDrives' => true
            ));

            return $file1->id;
        } catch (\Exception $e) {
            echo "Error Message: " . $e;
            return false;
        }
    }
}

You, 2 months ago • maj
```

2. Précisez les moyens utilisés :

- IDE: VSCode
- DBeaver pour l'affichage et la gestion des bases de données
- Langage SQL pour les bases de données

3. Avec qui avez-vous travaillé ?

J'ai réalisé cette mission de manière indépendante, jusqu'aux migrations de la base de données, faites par le CEO de l' Atelier, Julien ATHOMAS.

4. Contexte

Nom de l'entreprise, organisme ou association ▶

LA PLATEFORME

DOSSIER PROFESSIONNEL (DP)

Chantier, atelier, service -

Projet de fin d'année pour le diplôme CDA

Période d'exercice

- Du : 01/09/2024

au : 31/07/2025

5. Informations complémentaires (facultatif)

Activité-type 2

Concevoir et développer une application sécurisée
organisée en couches

Exemple n° 3 - Migration de l'API de l'école La Plateforme

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Comme précisé plus haut, l'API de l'école La Plateforme a été développée en CodeIgniter 1, qui jusqu'à sa version 3 n'était qu'une surcouche du langage PHP, très légère et permissive. Ces versions s'apparentaient plus à un ensemble d'outils utilitaires et de conventions autour de PHP qu'à un vrai framework structurant.

Les limites de la version initiale utilisée ont commencé à poser de véritables soucis lorsque nous avons eu de plus en plus besoin d'installer des "librairies" supplémentaires, afin de gérer des fonctionnalités telles que l'envoi de mail, la gestion de Google Drive, ou l'implémentation d'un bucket S3. En effet, j'étais à chaque fois obligé de chercher des versions antérieures de ces librairies, compatibles avec CodeIgniter 1, et de les installer manuellement.

Le projet de migration vers CodeIgniter 4 est donc arrivé sur la table, avec les deux objectifs principaux suivants:

- Installation des librairies facilitée grâce à l'utilisation de Composer
- Code mis à jour et maintenu plus facilement
- Système intégré de migration de base de données

Voici les points clés qui font de CodeIgniter 4 un framework PHP complet, structuré, professionnel et conforme aux standards modernes.

Structure et normes modernes

- Utilisation complète des namespaces et de l'autoloading PSR-4.
- Architecture 100 % orientée objet.
- Compatible avec PHP 7.2+ (puis 8+), tirant parti des nouvelles fonctionnalités du langage.

DOSSIER PROFESSIONNEL (DP)

Composants robustes et découplés

- Gestion propre des routes, des services, et de l'injection de dépendances.
- Système d'événements, middleware, et filtres.
- CLI intégré pour générer du code, gérer les migrations, etc.

Tests et développement pro

- Intégration native de PHPUnit pour les tests.
- Séparation nette entre environnement de développement et production.

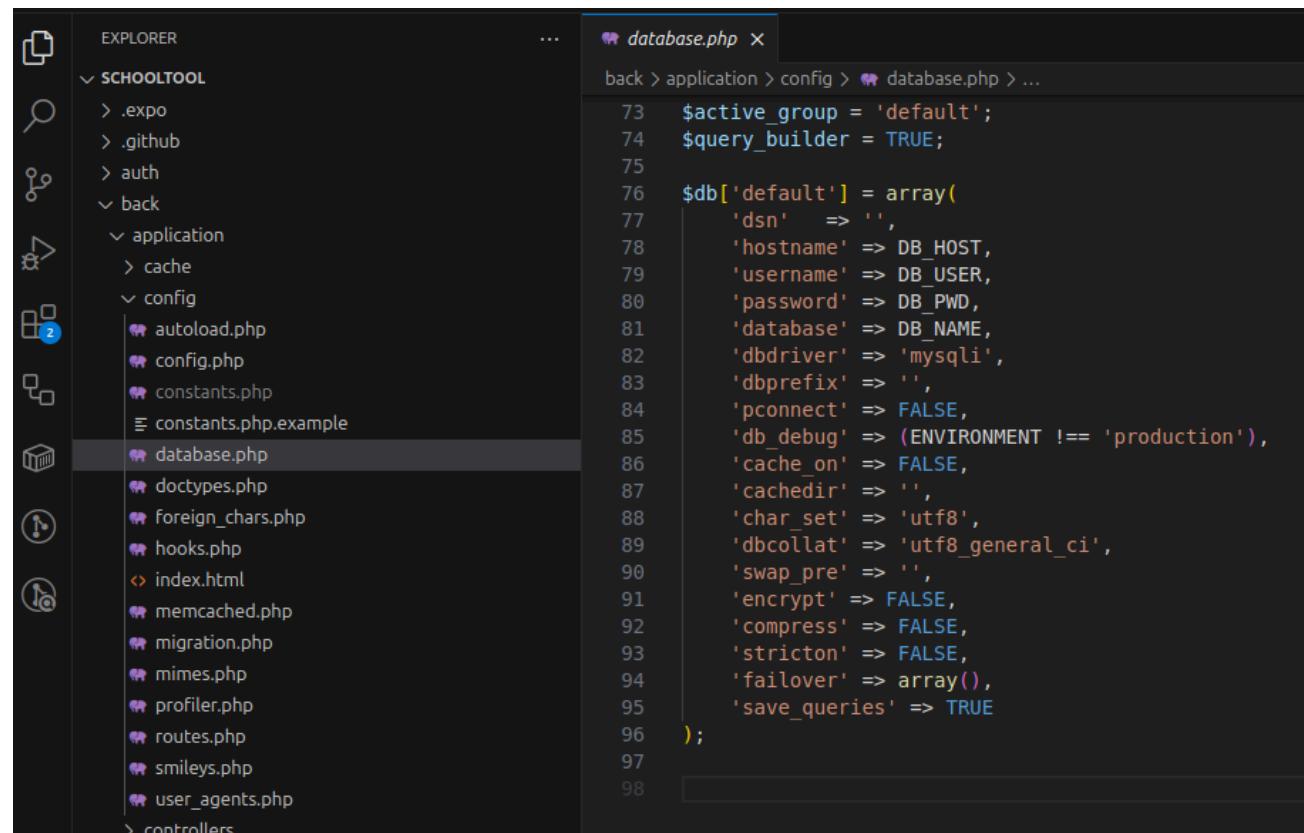
Meilleure sécurité et performance

- Protection CSRF, XSS, validation côté serveur améliorée.
- Moteur de template léger (View Cells, etc.).
- Système de caching, journaux, tâches planifiées, etc.

J'ai donc passé 3 semaines complètes à réaliser ce projet de migration. J'ai commencé par la base de données en utilisant le système de migration CodeIgniter4. Cependant, l'outil de base ne permet pas de générer des fichiers de migration à partir d'une base de données déjà existante. J'ai donc dû utiliser un package nommé DB-Craft, qui ajoute des commandes additionnelles à l'outil de migration CI4.

J'ai ensuite configuré mon fichier de configuration dans app/config/database.php de la manière suivante

DOSSIER PROFESSIONNEL (DP)



```
database.php X
back > application > config > database.php > ...
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => DB_HOST,
79     'username' => DB_USER,
80     'password' => DB_PWD,
81     'database' => DB_NAME,
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );
97
98
```

Enfin, j'ai pu générer les fichiers de migration initiaux via la commande **php spark get:migration**. Ci-dessous un exemple de fichier généré

DOSSIER PROFESSIONNEL (DP)

```
> .github
  < app
    > Config
    > Controllers
    > Core
  < Database
    < Migrations
      < .gitkeep
        2025-06-05-203940_create_absence_table.php
        2025-06-05-203940_create_acquired_skill_table.php
        2025-06-05-203940_create_activity_attendance_table.php
        2025-06-05-203940_create_activity_table.php
        2025-06-05-203940_create_alert_table.php
        2025-06-05-203940_create_alternance_table.php
        2025-06-05-203940_create_applicant_table.php
        2025-06-05-203940_create_calendar_day_table.php
        2025-06-05-203940_create_calendar_history_table.php
        2025-06-05-203940_create_calendar_table.php
        2025-06-05-203940_create_class_table.php 4
        2025-06-05-203940_create_DATABASECHANGELOG.php
        2025-06-05-203940_create_DATABASECHANGELOGLOCK.php
        2025-06-05-203940_create_followup_table.php
        2025-06-05-203940_create_invoice_table.php
        2025-06-05-203940_create_job_skill_table.php
        2025-06-05-203940_create_job_table.php
        2025-06-05-203940_create_log_table.php
        2025-06-05-203940_create_logtime_event_table.php
        2025-06-05-203940_create_logtime_table.php
        2025-06-05-203940_create_promotion_history_table.php
        2025-06-05-203940_create_promotion_table.php
        2025-06-05-203940_create_promotion_unit_table.php
        2025-06-05-203940_create_registration_table.php
        2025-06-05-203940_create_section_table.php
        2025-06-05-203940_create_skill_table.php
        2025-06-05-203940_create_student_table.php
        2025-06-05-203940_create_title_class_table.php
        2025-06-05-203940_create_title_table.php
        2025-06-05-203940_create_unit_completed_table.php
        2025-06-05-203940_create_unit_goal_table.php
        2025-06-05-203940_create_unit_history_table.php
        2025-06-05-203940_create_unit_table.php
        2025-06-05-203940_create_unit_viewer_table.php
        2025-06-05-203940_create_waiting_list_table.php
        2025-06-06-135247_create_staff_table.php
        2025-06-06-150727_create_appointment_table.php
        2025-06-06-195256_create_staff_availability_table.php
        2025-06-09-075607_AlterStaffTable.php
        2025-06-09-203533_create_staff_availability_exception_table.php
        2025-06-16-144453_AlterClassTable.php
        2025-06-19-123045_AlterAppointmentTable.php
      > Seeds
      > Filters
```

```
12  */
13
14 class CreateClassTable extends Migration
15 {
16   public function up()
17   {
18     // disable foreign key check ...
19     $this->db->disableForeignKeyChecks();
20
21     $this->forge->addField([
22
23       'id' => [
24         'type' => 'INT',
25         'constraint' => 11,
26         'null' => false,
27         'auto_increment' => true,
28       ],
29       'name' => [
30         'type' => 'VARCHAR',
31         'constraint' => 255,
32         'null' => false,
33       ],
34       'code' => [
35         'type' => 'VARCHAR',
36         'constraint' => 100,
37         'null' => true,
38       ],
39     ]);
40
41     // table keys ...
42
43     $this->forge->addPrimaryKey('id');
44
45
46     $this->forge->addUniqueKey('name');
47
48
49     // Create Table ...
50     $this->forge->createTable('class');
51
52     //enable foreign key check ...
53     $this->db->enableForeignKeyChecks();
54   }
55
56   //-----
57
58   public function down()
59   {
59     // disable foreign key check ...
60     $this->db->disableForeignKeyChecks();
61
62     // Drop Table ...
63     $this->forge->dropTable('class');
64
65     //enable foreign key check ...
66     $this->db->enableForeignKeyChecks();
67   }
68 }
```

DOSSIER PROFESSIONNEL (DP)

Grâce au package DB-Spark, j'ai par la suite pu générer des fichiers de seed afin d'alimenter les tables, grâce à la commande php spark get:seed.

Ci-dessous un exemple de fichier généré.

```

> gitlab
  15   class ActivitySeeder extends Seeder
  20     public function run(): void
    2575     ['id' => '2788', 'type' => 'Kick-off', 'is_mandatory' => '1', 'author' => 'akram.toumani@laplateforme.io', 'date' => '2025-03-31', 'unit_fk' => '305', 'comment' => NULL],
    2578     ['id' => '2789', 'type' => 'Kick-off', 'is_mandatory' => '1', 'author' => 'aurelie.praud@laplateforme.io', 'date' => '2025-03-31', 'unit_fk' => '313', 'comment' => NULL],
    2579     ['id' => '2790', 'type' => 'Kick-off', 'is_mandatory' => '1', 'author' => 'luc.van.der-meijden.pro@laplateforme.io', 'date' => '2025-03-31', 'unit_fk' => '271', 'comment' => NULL],
    2580     ['id' => '2791', 'type' => 'Kick-off', 'is_mandatory' => '1', 'author' => 'morgan.picard@laplateforme.io', 'date' => '2025-03-31', 'unit_fk' => '270', 'comment' => NULL],
    2581     ['id' => '2792', 'type' => 'Kick-off', 'is_mandatory' => '1', 'author' => 'luc.van.der-meijden.pro@laplateforme.io', 'date' => '2025-03-31', 'unit_fk' => '270', 'comment' => NULL],
    2582     ['id' => '2793', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'alicia.cordial.pro@laplateforme.io', 'date' => '2025-03-31', 'unit_fk' => '306', 'comment' => NULL],
    2583     ['id' => '2794', 'type' => 'Consultation technique', 'is_mandatory' => '1', 'author' => 'thierry.ramii@laplateforme.io', 'date' => '2025-03-31', 'unit_fk' => '304', 'comment' => NULL],
    2584     ['id' => '2795', 'type' => 'Consultation technique', 'is_mandatory' => '1', 'author' => 'morgan.picard@laplateforme.io', 'date' => '2025-03-31', 'unit_fk' => '297', 'comment' => NULL],
    2585     ['id' => '2796', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'thierry.ramii@laplateforme.io', 'date' => '2025-03-31', 'unit_fk' => '297', 'comment' => NULL],
    2586     ['id' => '2797', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'thierry.ramii@laplateforme.io', 'date' => '2025-03-31', 'unit_fk' => '244', 'comment' => NULL],
    2587     ['id' => '2798', 'type' => 'Kick-off', 'is_mandatory' => '1', 'author' => 'thierry.ramii@laplateforme.io', 'date' => '2025-03-31', 'unit_fk' => '244', 'comment' => NULL],
    2588     ['id' => '2799', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'morgan.picard@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '300', 'comment' => NULL],
    2589     ['id' => '2800', 'type' => 'Consultation technique', 'is_mandatory' => '1', 'author' => 'morgan.picard@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '300', 'comment' => NULL],
    2590     ['id' => '2801', 'type' => 'Consultation technique', 'is_mandatory' => '1', 'author' => 'nadir.ziane.pro@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '302', 'comment' => NULL],
    2591     ['id' => '2802', 'type' => 'Kick-off', 'is_mandatory' => '1', 'author' => 'morgan.picard@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '306', 'comment' => NULL],
    2592     ['id' => '2803', 'type' => 'Consultation technique', 'is_mandatory' => '1', 'author' => 'nadir.ziane.pro@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '303', 'comment' => NULL],
    2593     ['id' => '2804', 'type' => 'Consultation technique', 'is_mandatory' => '1', 'author' => 'morgan.picard@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '300', 'comment' => NULL],
    2594     ['id' => '2805', 'type' => 'Consultation technique', 'is_mandatory' => '1', 'author' => 'morgan.picard@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '306', 'comment' => NULL],
    2595     ['id' => '2806', 'type' => 'Consultation technique', 'is_mandatory' => '1', 'author' => 'william.kies.pro@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '306', 'comment' => NULL],
    2596     ['id' => '2807', 'type' => 'Consultation technique', 'is_mandatory' => '1', 'author' => 'morgan.picard@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '307', 'comment' => NULL],
    2597     ['id' => '2808', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'nadir.ziane.pro@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '303', 'comment' => NULL],
    2598     ['id' => '2809', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'akram.toumani@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '308', 'comment' => NULL],
    2599     ['id' => '2810', 'type' => 'Consultation technique', 'is_mandatory' => '1', 'author' => 'akram.toumani@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '302', 'comment' => NULL],
    2600     ['id' => '2811', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'aurelie.praud@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '313', 'comment' => NULL],
    2601     ['id' => '2812', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'aurelie.praud@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '313', 'comment' => NULL],
    2602     ['id' => '2813', 'type' => 'Consultation technique', 'is_mandatory' => '1', 'author' => 'thierry.ramii@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '304', 'comment' => NULL],
    2603     ['id' => '2814', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'morgan.picard@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '242', 'comment' => NULL],
    2604     ['id' => '2815', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'morgan.picard@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '309', 'comment' => NULL],
    2605     ['id' => '2816', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'thierry.ramii@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '244', 'comment' => NULL],
    2606     ['id' => '2817', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'nadir.ziane.pro@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '301', 'comment' => NULL],
    2607     ['id' => '2818', 'type' => 'Consultation technique', 'is_mandatory' => '1', 'author' => 'joris.vergueldeze.compro@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '310', 'comment' => NULL],
    2608     ['id' => '2819', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'luc.van.der-meijden.pro@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '301', 'comment' => NULL],
    2609     ['id' => '2820', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'alicia.cordial.compro@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '242', 'comment' => NULL],
    2610     ['id' => '2821', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'akram.toumani@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '208', 'comment' => NULL],
    2611     ['id' => '2822', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'thierry.ramii@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '304', 'comment' => NULL],
    2612     ['id' => '2823', 'type' => 'Suivi de projet', 'is_mandatory' => '1', 'author' => 'aurelie.praud@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '308', 'comment' => NULL],
    2613     ['id' => '2824', 'type' => 'Consultation technique', 'is_mandatory' => '1', 'author' => 'aurelie.praud@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '308', 'comment' => NULL],
    2614     ['id' => '2825', 'type' => 'Consultation technique', 'is_mandatory' => '1', 'author' => 'akram.toumani@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '247', 'comment' => NULL],
    2615     ['id' => '2826', 'type' => 'Soutenance', 'is_mandatory' => '1', 'author' => 'nadir.ziane.pro@laplateforme.io', 'date' => '2025-04-01', 'unit_fk' => '270', 'comment' => NULL];
    2616
    2617 // Cleaning up the table before seeding ...
    2618 $this->db->table('activity')->truncate();
    2619
    2620 //Using Query Builder Class ...
    2621 try {
    2622     $this->db->table('activity')->insertBatch($activity);
    2623 } catch (ReflectionException $e) {
    2624     throw new ReflectionException($e->getMessage());
    2625 }
    2626
    2627 //enable foreign key check
    2628 $this->db->enableForeignKeyChecks();
    2629
    2630 //UnitCompletedSeeder.php
    2631 //UnitSeeder.php
    2632 //UnitTypeSeeder.php
    2633 //UnitWorkSeeder.php
    2634 //UnitWorkTypeSeeder.php
    2635 //UnitWorkUserSeeder.php
    2636 //UnitUserSeeder.php
    2637 //UnitUserTypeSeeder.php
    2638 //UnitWorkListSeeder.php
    2639
    2640 // Filters
    2641
    2642 // Helpers
    2643 // gitkeep
    2644 // assets_helper.php
    2645 // Language
    2646 // Libraries
    2647 // AccessControl
    2648 // Plugins
    2649 // ApikitHelper.php
    2650 // BucketHelper.php
    2651 // GoogleDriveHelper.php
    2652 // GradeHelper.php
    2653 // JWT.php
  
```

Pour finir, j'ai créé un fichier nommé DatabaseSeeder, permettant de tout lancer tous les seeds avec la commande php spark get:seed DatabaseSeeder.

Ces deux étapes furent par la suite très utiles pour initier le projet sur des nouveaux ordinateurs, ainsi que pour le nouveau système d'hébergement des outils de l'école, que je détaillerai dans une autre section de ce dossier.

Une fois la partie base de données achevée, je me suis lancé dans la plus grosse partie de ce projet de migration, à savoir adapter la syntaxe des fichiers en fonction des nouvelles normes CodeIgniter.

Ci-dessous un exemple, avec l'adaptation du système de routing

DOSSIER PROFESSIONNEL (DP)

application/config/routes.php CodeIgniter 1

```
routes.php x
back > application > config > routes.php > ...
51  */
52
53 $route['default_controller'] = 'main';
54 $route['404_override'] = '';
55 $route['translate_uri_dashes'] = FALSE;
56
57 /* CLASS */
58 $route['class'] = 'Classe/Class';
59
60 /* UNIT */
61 $route['unit'] = 'Unit/Unit';
62 $route['unit/job'] = 'Unit/UnitJob';
63 $route['unit/duplicate'] = 'Unit/UnitDuplicate';
64 $route['unit/end'] = 'Unit/UnitEnd';
65
66 /* PROMOTION */
67 $route['promotion'] = 'Promotion/Promotion';
68 $route['promotion/end'] = 'Promotion/PromotionEnd';
69 $route['promotion/faithfulness'] = 'Promotion/PromotionFaithfulness';
70
71 /* SECTION */
72 $route['section'] = 'Section/Section';
73
74 /* SKILL */
75 $route['skill'] = 'Skill/Skill';
76
77 /* JOB */
78 $route['job'] = 'Job/Job';
79 $route['student/job/available'] = 'Job/StudentJobAvailable';
80 $route['job/student/available'] = 'Job/JobStudentAvailable';
81
82 /* JOB_SKILL */
83 $route['job/skill'] = 'Job_Skill/JobSkill';
84
```

app/Config/routes.php CodeIgniter 4

```
/* ABSENCE */
$routes->match(['GET', 'POST', 'PUT', 'DELETE'], 'absence', 'Absence::Absence');

/* ACQUIRED_SKILL */
$routes->match(['GET', 'PUT'], 'student/skill', 'Acquired_Skill::StudentSkill');
$routes->get('student/skill/total', 'Acquired_Skill::StudentSkillTotal');
$routes->get('student/class/total', 'Acquired_Skill::StudentClassTotal');
$routes->get('student/promotion/class/total', 'Acquired_Skill::StudentPromotionClassTotal');
$routes->get('promotion/class/total', 'Acquired_Skill::PromotionClassTotal');

/* Activity */
$routes->match(['GET', 'POST', 'PUT', 'DELETE'], 'activity', 'Activity::Activity');

/* Activity_Attendance */
$routes->match(['GET', 'POST', 'PUT', 'DELETE'], 'activity_attendance', 'Activity_Attendance::ActivityAttendance');

/* Alert */
$routes->match(['GET', 'POST', 'PUT', 'DELETE'], 'alert', 'Alert::Alert');

/* Alternance */
$routes->match(['GET', 'POST', 'PUT', 'DELETE'], 'alternance', 'Alternance::Alternance');

/* Upload new alternants */
$routes->post('new_alternants', 'Alternance::NewAlternants');

/* APPLICANT */
$routes->match(['GET', 'POST', 'PUT', 'DELETE'], 'applicant', 'Applicant::Applicant');
$routes->get('applicant/status', 'Applicant::ApplicantStatus');
$routes->get('applicant/situations', 'Applicant::Situations');
$routes->get('applicant/studies', 'Applicant::Studies');
/* Add new student */
$routes->post('new_applicant', 'Applicant::NewApplicant');
/* Upload new students */
$routes->post('new_applicants', 'Applicant::NewApplicants');

/* APPOINTMENT */
$routes->match(['GET', 'POST', 'PUT', 'DELETE'], 'appointment', 'Appointment::Appointment');

/* BUCKET S3 */
$routes->match(['GET', 'POST', 'DELETE'], 'file', 'BucketS3::File');

/* Calendar */
$routes->match(['GET', 'POST', 'PUT', 'DELETE'], 'calendar', 'Calendar::Calendar');
```

DOSSIER PROFESSIONNEL (DP)

J'ai respecté pour ce projet une architecture de type MVC. Grâce au système de classes et d'héritage, j'ai créé un Controller d'entrée permettant de déchiffrer le token présent dans une requête, puis d'enregistrer chaque requête dans une table de logs.

```
You, 2 months ago | 1 author (You)
1 <?php
2
3 namespace App\Core;
4
5 use App\Controllers\BaseController;
6 use App\Libraries\JWT;
7 use App\Libraries\TokenHelper;
8
9 You, 2 months ago | 1 author (You)
10 class LPTFController extends BaseController
11 {
12     protected JWT $jwt;
13     protected TokenHelper $tokenHelper;
14
15     public function initController(\CodeIgniter\HTTP\RequestInterface $request,
16                                     \CodeIgniter\HTTP\ResponseInterface $response,
17                                     \Psr\Log\LoggerInterface $logger)
18     {
19         parent::initController($request, $response, $logger);
20
21         $this->jwt = new JWT();
22         $this->tokenHelper = new TokenHelper($this->request);
23
24         if (!$this->tokenHelper->verifyToken()) {
25             $this->logAction();
26             $this->Status->ExpectationFailed()->send();
27             exit();
28         }
29
30         if (!$this->tokenHelper->tokenValid()) {      You, 2 months ago * LPTFController added
31             $this->logAction();
32             $this->Status->TokenExpired()->send();
33             exit();
34         }
35
36         $this->logAction($this->tokenHelper->getPayload());
37     }
38
39     protected function logAction(?array $payload = null): void
40     {
41         $logModel = new \App\Models\Log_Model();
42
43         $logData = [
44             'ip' => $this->request->getIPAddress(),
45             'email' => $payload['user_email'] ?? null,
46             'role' => $payload['role'] ?? null,
47             'url' => $_SERVER['HTTP_HOST'] . '/' . $this->request->getPath(),
48             'scope' => implode(',', $payload['scope']) ?? [],
49             'params' => $this->request->getGet() ? parse_url($_SERVER['REQUEST_URI'], PHP_URL_QUERY) : file_get_contents('php://input'),
50             'method' => $this->request->getMethod(),
51             'date' => date('Y-m-d H:i:s'),
52         ];
53
54         $logModel->postLog($logData);
55     }
56
57     protected function fail(array $response): \CodeIgniter\HTTP\ResponseInterface
58     {
59         return $this->response
60             ->setStatusCode(http_response_code())
61             ->setJSON($response);
62     }
63 }
```

DOSSIER PROFESSIONNEL (DP)

J'ai ensuite créé un fichier de controller pour chaque table, héritant de la classe montrée plus haut

```
You, last month | 2 authors (You and one other)
<?php

namespace App\Controllers;

use App\Core\LPTFController;

You, last month | 2 authors (You and one other)
class Absence extends LPTFController
{
    protected $Absence_Model;

    public function initController(\CodeIgniter\HTTP\RequestInterface $request, \CodeIgniter\HTTP\ResponseInterface $response, \Psr\Log\LoggerInterface $logger)
    {
        parent::initController($request, $response, $logger);

        $this->Absence_Model = new \App\Models\Absence_Model();
    }

    public function index()
    {
        echo json_encode(["whoami" => "api"]);
    }

    public function Absence()
    {
        $method = $_SERVER['REQUEST_METHOD'];
        $actions = []
        'GET' => 'getAbsence',
        'POST' => 'postAbsence',      You, 21 months ago * absence class and student helper
        'PUT' => 'putAbsence',
        'DELETE' => 'deleteAbsence';
    ];

        if (array_key_exists($method, $actions)) {
            $call = $actions[$method];
            $response = $this->$call();
        } else {
            $response = $this->Status->BadMethod();
        }
        if ($response === false) {
            return $this->Status->Error();
        }

        return json_encode($response);
    }

    private function getAbsence()
    {
        $params = $this->request->getGet();

        if ($this->RoleHelper->Access($params)) {
            return $this->Absence_Model->getAbsence($params);
        }
        return $this->Status->Denied();
    }

    private function postAbsence()
    {
        $params = $this->request->getRawInput();
        if ($this->RoleHelper->Access($params)) {
            return ($this->Absence_Model->postAbsence($params));
        }
        return ($this->Status->Denied());
    }
}
```

DOSSIER PROFESSIONNEL (DP)

Enfin, ci-dessous un exemple d'adaptation des fichiers de modèle

application/models/Class_Model.php CodeIgniter 1

```
<?php
You 5 months ago | author(You)
defined('BASEPATH') OR exit('No direct script access allowed');

class Class_Model extends LPTF_Model
{
    private $table = 'class';

    public function __construct()
    {
        parent::__construct();

        $this->load->helper('API_Helper');
        $this->api_helper = new API_Helper($this->db, $this->table);
    }

    public function getClass($params)
    {
        $constraints = [
            ['class_id', 'optional', 'number', true], ['class_name', 'optional', 'string', true]
        ];
        if ($this->api_helper->checkParameters($params, $constraints) == false)
        { return ($this->status()->PreconditionFailed()); }

        $fields = $this->getClassFields();
        $asked_fields = $this->api_helper->buildGet($params, $fields);
        $this->api_helper->addLimitAndOffset($params);

        $query = $this->db->get($this->table);

        return ($query->result_array());
    }

    public function postClass($params)
    {
        $constraints = [
            ['class_name', 'mandatory', 'string']
        ];
        if ($this->api_helper->checkParameters($params, $constraints) == false)
        { return ($this->status()->PreconditionFailed()); }

        $data = [
            'name' => $params['class_name']
        ];

        $response = $this->db->insert($this->table, $data);
        return ($response === true ? $this->db->insert_id() : false);
    }
}
```

app/Models/Class_Model.php CodeIgniter 4

```
class Class_Model extends LPTF_Model
{
    protected $api_helper;
    protected $protectedFields = false;
    protected $table = 'class';

    public function __construct()
    {
        parent::__construct();
        $this->api_helper = new \App\Libraries\ApiHelper($this->db, $this->table);
    }

    public function getClass($params)
    {
        $constraints = [
            ['class_id', 'optional', 'number', true],
            ['class_name', 'optional', 'string', true],
            ['class_code', 'optional', 'string', true]
        ];
        if ($this->api_helper->checkParameters($params, $constraints) == false) {
            return ($this->status()->PreconditionFailed());
        }

        $fields = $this->getClassFields(); // You 2 months ago + me
        $this->api_helper->buildGet($params, $fields);
        $this->api_helper->addLimitAndOffset($params);

        return ($this->api_helper->getBuilder()->get()->getResultArray());
    }

    public function postClass($params)
    {
        $constraints = [
            ['class_name', 'mandatory', 'string'],
            ['class_code', 'optional', 'string']
        ];
        if ($this->api_helper->checkParameters($params, $constraints) == false) {
            return ($this->status()->PreconditionFailed());
        }

        $data = [
            'name' => $params['class_name'],
        ];

        $optional_fields = [
            'code', 'class_code',
        ];

        foreach ($optional_fields as $field) {
            if (array_key_exists($field[1], $params)) {
                $data[$field[0]] = $params[$field[1]];
            }
        }

        try {
            $this->insert($data);
            if ($this->db->error()['code'] != 0) {
                return $this->status()->DatabaseError($this->db->error());
            }
            return $this->getInsertID();
        } catch (\Exception $e) {
            return $this->status()->DatabaseError($this->db->error());
        }
    }
}
```

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

-IDE: VSCode
-DBeaver pour l'affichage et la gestion des bases de données
-Composer
Documentation utilisée:
<https://www.codeigniter.com/>
https://github.com/robinNcode/db_craft

3. Avec qui avez-vous travaillé ?

J'ai réalisé cette mission de manière indépendante jusqu'à la mise en production, faite par le CEO de l' Atelier, Julien ATHOMAS.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **LA PLATEFORME**

Chantier, atelier, service ▶ **Projet de fin d'année pour le diplôme CDA**

Période d'exercice ▶ Du : **01/09/2023** au : **30/08/2025**

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Activité-type 3 Préparer le déploiement d'une application sécurisée

Exemple n° 1 - Déploiement et hébergement des outils de La Plateforme

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Initialement, l'hébergement des outils de l'école était géré par la direction informatique, qui avait choisi Plesk. C'est un panneau de gestion d'hébergement web tout-en-un, très utilisé pour le déploiement d'applications PHP/Apache, la gestion de bases de données, de domaines, de certificats SSL, etc.

Avantages :

- Interface graphique intuitive
- Déploiement rapide de sites web classiques
- Support natif des technologies web comme PHP, MySQL, Apache/Nginx

Limites :

- Peu adapté à une architecture moderne orientée microservices
- Difficulté à gérer la montée en charge et les redondances
- Gestion des versions et déploiements peu automatisée

Dans le but d'améliorer les processus de prise de décisions, la gestion des outils et de l'hébergement a été intégralement confiée à l'Atelier dans le courant de l'année 2025.

Afin de conserver une homogénéité dans la gestion des projets, l'Atelier a donc pris la décision de migrer l'hébergement des outils de l'école sur Kubernetes.

Kubernetes (ou K8s) est une plateforme open-source d'orchestration de conteneurs (Docker). Elle permet de déployer, gérer, scaler et surveiller des applications conteneurisées de façon automatisée, fiable et portable.

Avantages majeurs :

- **Scalabilité automatique** : montée ou descente en charge dynamique
- **Résilience** : redémarrage automatique des services en cas de panne
- **Déploiements reproductibles** grâce à des fichiers **yaml**

DOSSIER PROFESSIONNEL (DP)

- **Séparation claire des environnements (prod/dev/staging)**
- **CI/CD facilitée** avec des outils comme GitHub Actions
- **Portabilité** : déploiement possible en cloud, on-premise ou hybride

Cette migration nous a permis de gagner en efficacité, notamment en automatisant les déploiements .

Voici les différentes étapes que nous avons suivies afin de réaliser cette mission

Etape 1: conteneurisation intégralité des applications déployées

L'ensemble des outils digitaux de La Plateforme est regroupé dans un repository nommé stack-plateforme-interne. On y trouve notamment:

- auth : serveur d'authentification
- API-LaPlateforme: back-end regroupant la base de données métier de l'école
- Intranet-Etudiant2: site internet pour l'intranet des étudiants de l'école
- Intranet-ADM-2 : site internet pour l'intranet des employés de l'école

Nous retrouvons donc dans chaque repository d'application, un dockerfile permettant de configurer et déployer l'outil dans notre infrastructure. Par exemple, L'API et le service d'authentification utilisent un Dockerfile commun basé sur Apache + PHP (avec CodeIgniter):

DOSSIER PROFESSIONNEL (DP)

```
FROM --platform=linux/x86-64 php:7.3-apache

RUN apt-get update && apt-get upgrade -y \
&& apt-get install -y libzip-dev unzip \
&& docker-php-ext-install mysqli pdo pdo_mysql zip \
&& a2enmod rewrite

RUN chown -R www-data:www-data /var/www/html \
&& chmod -R 755 /var/www/html

EXPOSE 80

CMD ["apache2ctl", "-D", "FOREGROUND"]
```

FROM php:7.4-apache:

Utilise une image officielle Docker contenant PHP 7.4 avec Apache préinstallé. Idéale pour héberger une application PHP comme CodeIgniter.

RUN docker-php-ext-install mysqli pdo pdo_mysql:

Installe les extensions PHP nécessaires pour interagir avec une base de données MySQL/MariaDB. Obligatoire pour que CodeIgniter puisse se connecter à la base.

RUN a2enmod rewrite:

Active le module mod_rewrite d'Apache, essentiel pour le routing propre (URLs sans index.php) dans CodeIgniter.

COPY ./var/www/html:

Copie le contenu du dossier du projet (API ou Auth) dans le répertoire par défaut d'Apache (/var/www/html).

WORKDIR /var/www/html:

Définit le dossier de travail par défaut dans le conteneur. Toutes les commandes suivantes seront exécutées à partir de là.

EXPOSE 80:

Indique que le conteneur expose le port HTTP standard (80). Cela permet au docker-compose de le lier à un port externe (ex: 8000 ou 8001)

DOSSIER PROFESSIONNEL (DP)

Etape 2: intégration continue

L'intégration continue, ou CI (Continuous Integration), consiste à automatiser des actions (build, tests, vérification) à chaque modification du code. Nous avons utilisé GitHub Actions afin d'automatiser ces processus dès qu'une modification est poussée sur une branche de travail.

```
1 name: Main Workflow
2 on:
3   push:
4     branches:
5       - main
6       - master-CI4
7       - master
8   workflow_call:
9     outputs:
10    repo_name:
11      description: "API URL"
12      value: ${{ jobs.get-repo-name.outputs.repo_name }}
13    timestamp:
14      description: "API URL"
15      value: ${{ jobs.get-repo-name.outputs.timestamp }}
16 jobs:
17   get-repo-name:
18     runs-on: ubuntu-latest
19     outputs:
20       repo_name: ${{ steps.get_name.outputs.repo_name }}
21       timestamp: ${{ steps.get_timestamp.outputs.timestamp }}
22     steps:
23       - name: get_name
24         id: get_name
25         run: |
26           echo "repo_name=$(basename ${{ github.repository }})" >> $GITHUB_OUTPUT
27       - name: get_timestamp
28         id: get_timestamp
29         run: |
30           echo "timestamp=$(date --utc +%FT%H-%M-%SZ)" >> $GITHUB_OUTPUT
31   run-build-job:
32     uses: laplateformeio/atelier-workflows/.github/workflows/workflow-build.yaml@main
33     needs: [get-repo-name]
34     with:
35       BUILD_FOLDER: .
36       IMAGE_NAME: ${{ needs.get-repo-name.outputs.repo_name }}
37       TAG: dev-${{ needs.get-repo-name.outputs.timestamp }}
38     secrets:
39       PAT_TOKEN: ${{ secrets.PAT_TOKEN }}
```

DOSSIER PROFESSIONNEL (DP)

Le premier bloc, “on:”, définit où et à quel moment le workflow doit se lancer. En l’occurrence ici, à chaque push sur les branches main, master-CI4 et master.

Le bloc “workflow_call” permet d’exécuter cette tâche dans un autre workflow en exposant deux valeurs :

- repo_name : nom du repo déclencheur
- timestamp : horodatage exact de l'exécution

Le bloc “jobs” permet de récupérer dynamiquement le nom du dépôt et un timestamp, via les deux étapes suivantes:

- get_name : Extrait le nom du dépôt Git (ex : intranet-ADM-2) pour le réutiliser plus loin (comme nom d'image Docker).
- get_timestamp : Génère une date UTC formatée, utilisée comme TAG dans l'image Docker (utile pour tracer les builds).

Enfin, le bloc “run-build-jobs” permet via le “uses” de récupérer et exécuter un workflow distant en lui passant des variables. Cela permet de :

- Construire une image Docker à partir du dossier courant (./)
- Nommer l'image selon le dépôt
- Taguer l'image avec un timestamp
- Authentifier l'action avec PAT_TOKEN (Personal Access Token GitHub)

En résumé, ce fichier de workflow permet de :

- Définir à quel moment ce déclenche la tâche (push sur une branche)
- Récupérer le nom du dépôt concerné et générer un timestamp
- Utiliser un *template distant* pour lancer un build Docker avec des variables dynamiques

DOSSIER PROFESSIONNEL (DP)

Etape3: cd / déploiement

Le déploiement d'un outil sur Kubernetes se fait via un fichier de configuration YAML (deployment.yaml) contenant toutes les informations nécessaires au déploiement de l'application dans le cluster Kubernetes.

```
replicas: 1
selector:
  matchLabels:
    app: intra-adm-cybersup
template:
  metadata:
    labels:
      app: intra-adm-cybersup
  spec:
    imagePullSecrets:
      - name: dockerconfigjson-github-com
    containers:
      - name: intra-adm-cybersup
        image: ghcr.io/laplateformeio/intranet-adm-2:0.0.2 # {"$imagepolicy": "production:intra-adm-cybersup"}
        imagePullPolicy: IfNotPresent
        envFrom:
          - secretRef:
              name: intra-adm-secrets-cybersup
    ports:
      - containerPort: 80
    volumeMounts:
      - name: "config"
        mountPath: "/var/www/html/application/config/constants.php"
        subPath: "constants.php"
      - name: "logo"
        mountPath: "/var/www/html/assets/img/logo.png"
        subPath: "logo.png"
      - name: "favicon"
        mountPath: "/var/www/html/assets/img/favicon.png"
        subPath: "favicon.png"
    volumes:
      - name: "config"
        configMap:
          name: "intra-adm-constants-cybersup"
      - name: "logo"
        configMap:
          name: "intra-logo-cybersup"
      - name: "favicon"
        configMap:
          name: "intra-favicon-cybersup"
---
apiVersion: v1
kind: Service
metadata:
  name: intra-adm-service-cybersup
  namespace: production
spec:
  selector:
    app: intra-adm-cybersup
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
    type: ClusterIP
---
apiVersion: networking.k8s.io/v1
```



DOSSIER PROFESSIONNEL (DP)

Ce que définit ce fichier :

1. Deployment :

- name: intra-adm-cybersup
- image: Docker à déployer (ex: laplateformeio/intranet-adm:2.0.0.2)
- envFrom: variables d'environnement (via ConfigMap)
- volumeMounts : configuration PHP et ressources front
- replicas: 1 : nombre de pods (instances)

2. Service :

- Expose l'application dans le cluster avec un nom (intra-adm-service-cybersup)
- Permet de faire le lien entre le déploiement et l'ingress

3. Ingress :

- Permet l'accès externe à l'application via un nom de domaine personnalisé (intra-adm.cybersup.ai)
- Utilise TLS (Let's Encrypt) pour un accès HTTPS sécurisé

Enfin, Nous avons utilisé Flux, un outil d'automatisation GitOps, pour gérer et déployer les fichiers deployment.yaml. Toute la configuration de l'infrastructure (déploiements, services, ingress, etc.) est versionnée dans un dépôt Git. Le cluster vient automatiquement chercher les fichiers dans ce dépôt pour appliquer les changements.

Avantages de GitOps :

- **Traçabilité** : chaque modification est historisée dans Git
- **Automatisation** : déploiement immédiat après commit
- **Centralisation** : un seul dépôt pour toute l'infra
- **Rollback simple** : retour à une version stable en 1 clic

En pratique, les traditionnelles pull requests que je faisais précédemment, sont désormais remplacées

DOSSIER PROFESSIONNEL (DP)

par:

- **La création de tags** : déploiement automatique sur l'environnement de peprod
- **La création de releases** : déploiement automatique sur l'environnement de production

Ce système garantit un **cycle de livraison contrôlé** et fiable, compatible avec l'intégration continue (CI) mise en place.

2. Précisez les moyens utilisés :

Dockerhub
Kubernetes
CodeIgniter4

3. Avec qui avez-vous travaillé ?

J'ai travaillé en collaboration avec Julien Athomas, CEO de l'Atelier.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **LA PLATEFORME**

Chantier, atelier, service ▶ **Changement du système d'hébergement des outils La Plateforme**

Période d'exercice ▶ Du : **01/09/2024** au : **31/07/2025**

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Activité-type 3 Préparer le déploiement d'une application sécurisée

Exemple n° 2 - Développement d'une application mobile d'intranet étudiant

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

La Plateforme met à disposition de ses étudiants et employés deux interfaces web principales : un intranet étudiant (consultation des absences, des projets, du profil...) et un intranet administration (création et gestion des promotions, création et gestion des projets, etc...). Ces outils, bien que fonctionnels, souffrent d'une accessibilité limitée depuis un smartphone et d'une ergonomie peu optimisée pour la navigation mobile.

Ces limites d'ergonomie ne sont pas un problème pour les employés de l'école, qui utilisent l'intranet administration sur leurs postes de travail. C'est en revanche le cas pour les étudiants, qui ont une utilisation bien différente de leur intranet.

Etant à la fois étudiant et sous-traitant de La Plateforme, je suis donc utilisateur et développeur de ces outils depuis 2 années et demi. Cette double casquette m'a donc permis d'identifier un problème et d'en penser la solution. J'ai donc, dans le cadre de mon diplôme de Concepteur Développeur d'Application, développé une application nommée Schooltool, en partenariat avec un des mes camarades de promotion.

L'objectif est de proposer une interface mobile moderne, accessible et intuitive permettant aux étudiants d'accéder à leurs outils essentiels de suivi de scolarité, actuellement disponibles uniquement sur des interfaces web peu adaptées à un usage quotidien sur smartphone.

L'application SchoolTool vise à :

- Centraliser les fonctionnalités les plus utilisées par les étudiants dans une seule interface mobile.
- Améliorer l'expérience utilisateur via une navigation fluide et intuitive.
- S'appuyer sur les systèmes d'authentification et d'API existants de La Plateforme sans les modifier.
- Offrir une base solide pour de futures évolutions (notification, mise à disposition sur stores, personnalisation).

DOSSIER PROFESSIONNEL (DP)

L'une des contraintes de ce projet fût de s'adapter aux outils existants de l'école. Schooltool devait donc respecter certaines règles pour venir s'intégrer dans les outils de l'école, notamment:

- L'outil d'Authentification de La Plateforme se base sur le protocole Oauth2 via Google.
- L'API de La Plateforme, codée en CodeIgniter

Conteneurisation

Nous avons donc créé un environnement docker comprenant

- L'API existante de l'école
- L'authentification de l'école
- Notre application front-end

```
db:
  image: mariadb
  command: --default-authentication-plugin=mysql_native_password
  restart: always
  ports:
    - 3307:3306
  environment:
    MYSQL_ROOT_PASSWORD: root

D-Ran Service
back:
  build: ./back
  container_name: schooltool-back
  volumes:
    - ./back:/var/www/html
  ports:
    - 8000:80
  depends_on:
    - db
  environment:
    TAB_ROLES: '[ {"id": 1, "name": "student", "access": "Student_Access"}, {"id": 2, "name": "hors-parcours"}, {"id": 3, "name": "alumni"}, {"id": 4, "name": "admin"}, {"id": 5, "name": "crm", "access": "CRM_Access"} ]'
    DB_HOST: db
    DB_PORT: 3306
    DB_PWD: root
    DB_NAME: schooltool_api
    LPTF_JWT_KEY: toto
    GOOGLE_CLIENT_ID: '604347883543-cu73up3fqo5r9gn18tqpkf3tu9ud4ls4.apps.googleusercontent.com'
    GOOGLE_DRIVE_ABSENCE_FOLDER: 'toFill'
    GOOGLE_CLIENT_EMAIL: ''
    GOOGLE_PRIVATE_KEY: ''
    IS_DEV: true

D-Ran Service
auth:
  build: ./auth
  container_name: schooltool-auth
  volumes:
    - /auth:/var/www/html
  ports:
    - 8001:80
  depends_on:
    - db
  environment:
    TAB_ROLES: '[ {"id": 1, "name": "student", "access": "Student_Access"}, {"id": 2, "name": "hors-parcours"}, {"id": 3, "name": "alumni"}, {"id": 4, "name": "admin"}, {"id": 5, "name": "crm", "access": "CRM_Access"} ]'
    DB_HOST: db
    DB_USER: root
    DB_PWD: root
    DB_NAME: schooltool_auth
    LPTF_JWT_KEY: toto
    GOOGLE_CLIENT_ID: '604347883543-cu73up3fqo5r9gn18tqpkf3tu9ud4ls4.apps.googleusercontent.com'
    IS_DEV: true
    OAUTH_URL: 'http://localhost:8001/oauth'

D-Ran Service
front:
  build: ./front
  container_name: schooltool-front
  tty: true
  stdin_open: true
  volumes:
    - ./front:/app
  ports:
    - 8082:8081
    - 19000:19000 # Metro Bundler
    - 19001:19001 # WebSocket (live/hot reload)
    - 19002:19002 # React Native Debugger
  depends_on:
    - back
    - auth
  environment:
    API_URL: 'http://localhost:8000'
    AUTH_URL: 'http://localhost:8001'

volumes:
  db_data:
```

DOSSIER PROFESSIONNEL (DP)

Authentification Google OAuth

Une de nos tâches pour SchoolTool fût donc de développer un système de connexion qui puisse fonctionner avec l'authentification La Plateforme.

La première étape était donc d'envoyer à l'API Google une requête comprenant l'identifiant client La Laplateforme.

```
const [request, response, promptAsync] = AuthSession.useAuthRequest(  
  {  
    clientId: googleClientId,  
    redirectUri,  
    usePKCE: true,  
    scopes: [  
      'openid',  
      'profile',  
      'email',  
      'https://www.googleapis.com/auth/calendar.readonly',  
    ],  
    extraParams: {  
      access_type: 'offline',  
      prompt: 'consent',  
      response_type: 'code',  
    },  
    { authorizationEndpoint: 'https://accounts.google.com/o/oauth2/v2/auth' },  
  }  
);
```

Nous avons également intégré l'API Google Calendar, puisque c'est l'outil utilisé pour la gestion des calendriers de formation.

Cette étape permet de récupérer un code Google nécessaire à l'étape suivante, qui est de récupérer une token_id requise par l'authentification La Plateforme.

DOSSIER PROFESSIONNEL (DP)

```
const exchangeCodeForToken = async (code: string) => {
  try {
    const response = await fetch('https://oauth2.googleapis.com/token', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/x-www-form-urlencoded',
      },
      body: new URLSearchParams({
        code,
        client_id: googleClientId,
        client_secret: isExpoGo ? undefined : googleSecret,
        redirect_uri: redirectUri,
        grant_type: 'authorization_code',
        code_verifier: request?.codeVerifier || '',
      }).toString(),
    });
    const tokenData = await response.json();

    if (!tokenData.access_token) {
      Alert.alert('Erreur', 'Token Google non reçu');
      return;
    }
  }
```

Une fois la token_id Google récupérée, nous avons pu mettre en place une requête vers l'authentification La Plateforme.

```
const authToken = await fetch(`${authUrl}/oauth`, {
  method: 'POST',
  headers: {
    'Content-Type': 'application/x-www-form-urlencoded',
  },
  body: `token_id=${tokenData.id_token}`,
});

const apiToken = await authToken.json();

const decoded = decodeJWT(apiToken.token);
if (!decoded) {
  Alert.alert('Erreur', 'Impossible de décoder le token utilisateur');
  return;
}
```

Enfin, une fois le token d'authentification et le token d'accès à l'API reçus, nous avons pu stocker:

- Les tokens La Plateforme et Google, nécessaires pour les requêtes vers l'API et le refresh des tokens

DOSSIER PROFESSIONNEL (DP)

- Les informations utilisateur

```
const userSession = {
    accessToken: apiToken.token,
    authToken: apiToken.authtoken,
    googleAccessToken: tokenData.access_token,
    googleExpiresIn: tokenData.expires_in,
    googleExpiresAt: Date.now() + tokenData.expires_in * 1000,
    googleScope: tokenData.scope,
    googleRefreshToken: tokenData.refresh_token,
};

const userData = {
    id: decoded.user_id,
    email: decoded.user_email,
    role: decoded.role,
    scope: decoded.scope,
};

await Session.set(userSession, userData);
setUser(userData);
```

Requêtes API

Notre application devait, afin de pouvoir communiquer avec l'API de l'école, respecter certaines règles établies. Il fallait donc imaginer un système qui:

- Inclut toujours les données dans data (même pour les DELETE), car CodeIgniter utilise \$this->request->getRawInput() qui lit le corps brut de la requête.
- Utilise multipart/form-data pour le rafraîchissement (comme l'exige le endpoint /refresh), et application/x-www-form-urlencoded pour les autres requêtes.
- Construit dynamiquement l'URL en fonction de la méthode fournie et enchaîne les paramètres dans l'URL (GET) ou dans le corps (POST, PUT, DELETE) selon les besoins.
- Ajoute manuellement un header Token (non standard), attendu par l'API pour la vérification d'accès.

Nous avons donc mis en place, dans un dossier nommé Service, un fichier nommé ApiActions. Dans l'architecture de SchoolTool, le module ApiActions agit comme un intermédiaire universel entre le frontend React Native et l'API backend CodeIgniter. Il centralise la logique des appels API

DOSSIER PROFESSIONNEL^(DP)

HTTP (GET, POST, PUT, DELETE) tout en respectant les contraintes spécifiques de sécurité, de structure, et de gestion de session imposées par le backend.

Structure de la requête

Chaque méthode (get, post, put, delete) suit le même schéma :

- Vérification/rafraîchissement du token
- Construction de l'URL finale
- Encodage des paramètres (URLSearchParams)
- Requête via axios.request(...)

DOSSIER PROFESSIONNEL (DP)

```
export const ApiActions = {
    async get(payload: ApiPayload): Promise<AxiosResponse | null> {
        let route = payload.route;
        let url = buildUrl(payload.params);

        const token = await getApiToken();
        if (!token) {
            router.push('/');
            return null;
        }

        let params = {
            method: 'get',
            maxBodyLength: Infinity,
            url: `${ENV.LPTF_API_URL}/${route}?${url}`,
            headers: {
                Token: token || '',
            },
        };
        try {
            const response = await axios.request(params);
            return response;
        } catch (error) {
            console.error('GET request error: ', error);
            throw error;
        }
    },

    async post(payload: ApiPayload): Promise<AxiosResponse | null> {
        let route = payload.route;
        const body = payload.params;

        const token = await getApiToken();
        if (!token) {
            router.push('/');
            return null;
        }

        let params = {
            method: 'post',
            maxBodyLength: Infinity,
            url: `${ENV.LPTF_API_URL}/${route}?`,
            data: new URLSearchParams(body).toString(),
            headers: {
                Token: token || '',
            },
        };
        try {
            const response = await axios.request(params);
            return response;
        } catch (error) {
            console.error('POST request error: ', error);
            throw error;
        }
    },
}
```

DOSSIER PROFESSIONNEL (DP)

Les requêtes GET suivent un cheminement légèrement différent, puisque l'API, pour cette méthode, lit les paramètres directement dans l'url.

```
function buildUrl(  
  params: Record<string, string | number | (string | number)[]>,  
): string {  
  let url = '';  
  Object.keys(params).forEach((key) => {  
    if (Array.isArray(params[key])) {  
      params[key].forEach((element) => {  
        url += `${key}[]=$element&`;  
      });  
    } else {  
      url += `${key}=${params[key]}&`;  
    }  
  });  
  return url;  
}
```

Par exemple, la requête suivante:

```
ApiActions.get({  
  route: 'absence/Absence',  
  params: { student_id: '123' }  
});
```

génèrera : http://localhost:8000/absence/Absence?student_id=123

Gestion du token d'authentification

Avant chaque requête :

- Le module tente d'obtenir le token accessToken via la session stockée (Session.getSession()).
- Il vérifie si ce token est expiré via un décodage local JWT.
- Si le token est expiré, une requête de rafraîchissement est envoyée (/refresh), exploitant le authToken.

DOSSIER PROFESSIONNEL (DP)

- En l'absence de token valide, l'utilisateur est automatiquement redirigé vers la page d'accueil (router.push('/')).

```
async function getApiToken() {
  try {
    const session = await Session.getSession();

    if (!session) {
      router.push('/');
      return null;
    }

    if (isTokenExpired(session.accessToken)) {
      const newToken = await refreshToken(session);
      if (newToken) {
        await Session.updateAccessToken(newToken);
        return newToken;
      } else {
        await Session.clear();
        router.push('/');
        return null;
      }
    }

    return session.accessToken;
  } catch (error) {
    console.error('Error fetching session: ', error);
    return null;
  }
}
```

Tests et qualité logicielle

Des tests de composants ont été mis en place afin de valider le bon fonctionnement des interfaces utilisateurs. Ces tests s'appuient sur la bibliothèque Jest associée à React Native Testing Library, ce qui permet de simuler l'affichage, les interactions utilisateur et les effets déclenchés par le cycle de vie du composant (useEffect, useState, etc.).

Ces tests permettent de s'assurer que :

- Les composants réagissent correctement aux données entrantes.
- Les interactions utilisateur fonctionnent comme prévu.

DOSSIER PROFESSIONNEL (DP)

- Les éventuelles régressions soient détectées automatiquement.

Exemple ci-dessous, test du composant JobsAvailable

```
1 import React from 'react';
2 import { render, waitFor, fireEvent } from '@testing-library/react-native';
3 import JobsAvailable from '../jobs/JobsAvailable';
4 import { ApiActions } from '@services/ApiServices';
5
6 jest.mock('@services/ApiServices', () => ({
7   ApiActions: {
8     get: jest.fn(),
9   },
10 }));
11
12 const mockJobs = [
13   {
14     job_id: '1',
15     job_name: 'Développeur Frontend',
16     job_unit_name: 'Unit 1',
17   },
18   {
19     job_id: '2',
20     job_name: 'Développeur Backend',
21     job_unit_name: 'Unit 2',
22   },
23 ];
24
25 describe('JobsAvailable', () => {
26   beforeEach(() => {
27     (ApiActions.get as jest.Mock).mockResolvedValue({
28       status: 200,
29       data: mockJobs,
30     });
31   });
32
33   it('affiche les jobs disponibles après chargement', async () => {
34     const { getByText } = render(<JobsAvailable />);
35
36     await waitFor(() => {
37       expect(getByText('Développeur Frontend')).toBeInTheDocument();
38       expect(getByText('Développeur Backend')).toBeInTheDocument();
39     });
40   });
41
42   it('filtre les jobs par unité via le picker', async () => {
43     const { getByText, getByDisplayValue, queryByText } = render(
44       <JobsAvailable />,
45     );
46
47     await waitFor(() => {
48       expect(getByText('Développeur Frontend')).toBeInTheDocument();
49     });
50
51     const picker = getByDisplayValue('Toutes les Units');
52     fireEvent(picker, 'valueChange', 'Unit 1');
53
54     await waitFor(() => {
55       expect(getByText('Développeur Frontend')).toBeInTheDocument();
56       expect(queryByText('Développeur Backend')).toBeNull();
57     });
58   });
59
60   it('ouvre la modale en cliquant sur un job', async () => {
61     const { getByText, queryByText } = render(<JobsAvailable />);
62
63     await waitFor(() => {
64       expect(getByText('Développeur Frontend')).toBeInTheDocument();
65     });
66
67     fireEvent.press(getByText('Développeur Frontend'));
68
69     await waitFor(() => {
70       expect(queryByText(/Groupe/i)).toBeInTheDocument();
71     });
72   });
73});
```

DOSSIER PROFESSIONNEL (DP)

Affichage des données après chargement

Ce test vérifie que les projets sont correctement affichés après que la requête API (mockée ici) a renvoyé les données. L'utilisation de waitFor permet de s'assurer que le rendu s'effectue après la fin du chargement.

Filtrage dynamique via le picker

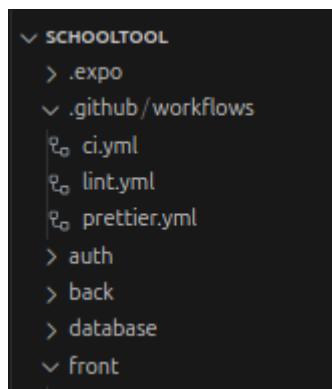
Ce test simule l'interaction avec le menu déroulant (Picker) permettant de filtrer les projets. En sélectionnant "Unit 1", le composant ne doit plus afficher les jobs associés à "Unit 2". Cela vérifie que la logique de filtrage est correctement implémentée côté client.

Ouverture de la modale au click sur un projet

Ce test valide l'interaction utilisateur : lorsqu'on appuie sur un projet (Pressable), une modale d'information doit s'ouvrir. On vérifie sa présence à l'aide d'un mot-clé contenu dans la modale (Groupe).

Automatisation des tests

À la racine de notre projet, nous avons ajouté un dossier .github, dans lequel sont stockés nos fichiers permettant de paramétriser les github actions.



C'est notamment ici que l'on retrouve le fichier permettant de lancer le test de nos composants à chaque pull request effectuée.

DOSSIER PROFESSIONNEL (DP)

```
1 name: CI/CD Pipeline
2
3 on:
4   push:
5     branches:
6       - main
7   pull_request:
8
9 jobs:
10   build:
11     runs-on: ubuntu-latest
12
13     services:
14       docker:
15         image: docker:24.0.2
16         options: --privileged
17
18     steps:
19       - name: Checkout code
20         uses: actions/checkout@v3
21
22       - name: Set up Node.js
23         uses: actions/setup-node@v3
24         with:
25           node-version: 18
26
27       - name: Install frontend dependencies
28         run: |
29           cd front
30           npm install
31
32       - name: Run frontend tests
33         run: |
34           cd front
35           npm test -- --ci
36
37       - name: Set up Docker Buildx
38         uses: docker/setup-buildx-action@v2
39
40       - name: Build Docker Compose
41         run: docker compose -f docker-compose.yml build
42
43       - name: Send email notification
44         uses: dawidd6/action-send-mail@v3
45         with:
46           server_address: smtp.gmail.com
47           server_port: 465
48           username: ${{ secrets.EMAIL_USERNAME }}
49           password: ${{ secrets.EMAIL_PASSWORD }}
50           subject: Nouvelle action sur le repo!
51           body: "Une action a été déclenchée sur la branche main."
52           to: ${{ secrets.EMAIL_USERNAME }}
53           from: ${{ secrets.EMAIL_USERNAME }}
54           secure: true
```

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Docker
CodeIgniter
React Native + Expo
Expo Go
GitHub Projects
Figma

3. Avec qui avez-vous travaillé ?

J'ai travaillé en collaboration avec Hervé BEZIAT, étudiant en alternance à La Plateforme

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **LA PLATEFORME**

Chantier, atelier, service ▶ *Projet de fin d'année pour le diplôme CDA*

Période d'exercice ▶ Du : **01/09/2024** au : **31/07/2025**

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Déclaration sur l'honneur

Je soussigné(e) **ALOESODE Alexandre** ,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis
l'auteur(e) des réalisations jointes.

Fait à **Marseille** le **30/07/2025**

pour faire valoir ce que de droit.

Signature : ALOESODE Alexandre