



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL

(DP)

Nom de naissance ➤ Boujnah
Nom d'usage ➤
Prénom ➤ Nadia
Adresse ➤ 1189 avenue du Général Garbay, 06210
Mandelieu-La-Napoule

Titre professionnel visé

Cliquez ici pour entrer l'intitulé du titre professionnel visé.

MODALITÉ D'ACCÈS :

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

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.

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

Activité-type 1 : Développer la partie front-end d'une application web ou web mobile sécurisée

p. 5

- CP 1 Installer et configurer son environnement de travail en fonction du projet web ou web mobile
- CP 2 Maquetter des interfaces utilisateur web ou web mobile
- CP 3 Réaliser des interfaces utilisateur statiques web ou web mobile
- CP 4 Développer la partie dynamique des interfaces utilisateur web ou web mobile

p. 5

p. 7

p. 10

p. 13

Activité-type n° 2 : Développer la partie back-end d'une application web ou web mobile sécurisée

p. 15

- CP 5 Mettre en place une base de données relationnelle.
- CP 6 Développer les composants d'accès aux données SQL et NoSQL.
- CP 7 Développer des composants métier côté serveur.
- CP 8 Documenter le déploiement d'une application dynamique web ou web mobile.

p. 15

p. 17

p. 19

p. 21

Titres, diplômes, CQP, attestations de formation *(facultatif)*

p. 24

Déclaration sur l'honneur

p. 25

Documents illustrant la pratique professionnelle *(facultatif)*

p. 26

Annexes *(Si le RC le prévoit)*

p. 27

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1

Développer la partie front-end d'une application web ou web mobile sécurisée

CP 1

Installer et configurer son environnement de travail en fonction du projet web ou web mobile

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

Dans le cadre du développement de mon projet MeetTrip, une application web permettant à des voyageuses et voyageurs de se rencontrer autour d'activités, j'ai installée et configurée mon environnement de travail sous Windows avec Laravel, Inertia.js, React et Tailwind CSS.

J'ai d'abord réalisé les maquettes sur Figma (accueil, création d'activité, profil utilisatrice, etc.), qui m'ont servies de base pour l'intégration. Ensuite, j'ai créé un projet Laravel via Composer, intégrée Inertia.js pour relier React au backend, et installée Tailwind CSS pour obtenir une interface moderne et responsive.

J'ai utilisée le starter kit Laravel Breeze, combinant React 19, TypeScript, Tailwind CSS, shadcn/ui et Vite, afin d'assurer un développement rapide et structuré.

Pour la base de données, j'ai choisie SQLite, que j'ai configurée dans le fichier .env.

J'ai gérée les modèles et leurs relations avec Eloquent, créé les migrations et prérempli certaines données grâce aux seeders.

J'ai versionnée le projet avec Git, en travaillant sur Visual Studio Code et en utilisant les extensions Prettier, Intelephense et Tailwind CSS IntelliSense.

Concernant l'accessibilité, j'ai vérifiée la conformité aux bonnes pratiques du RGAA à l'aide de l'outil Lighthouse, intégré à Chrome DevTools. Cet audit m'a permis d'identifier plusieurs points d'amélioration, notamment le contraste des couleurs, la présence de textes alternatifs sur les images et la hiérarchie correcte des titres.

Après avoir corrigée ces éléments, le score d'accessibilité de mes pages s'est amélioré, garantissant une interface plus lisible et inclusive.

2. Précisez les moyens utilisés :

Pour réaliser ce projet, j'ai utilisée Visual Studio Code avec les extensions Prettier, PHP Intelephense, Tailwind CSS IntelliSense et Database Client.

J'ai utilisée Figma pour créer les maquettes des interfaces de l'application.

Le projet a été développé avec le starter kit React de Laravel, intégrant React 19, TypeScript, Tailwind CSS, shadcn/ui et Vite.

La base de données utilisée est SQLite, configurée via le fichier .env.

Pour le versionnement du code, j'ai utilisée Git en local, avec hébergement sur GitHub pour la sauvegarde et le suivi des dépôts.

3. Avec qui avez-vous travaillé ?

J'ai réalisée ce projet seule, dans le cadre de ma formation.

4. Contexte

Nom de l'entreprise, organisme ou association ▶

Ecole La Plateforme

Chantier, atelier, service

▶ Projet personnel réalisé en cours de formation

Période d'exercice

▶ Du 07/04/202 au 07/07/2025
5

5. Informations complémentaires (facultatif)

Cliquez ici pour taper du texte.

Activité-type 1

Développer la partie front-end d'une application web ou web mobile sécurisée

CP 2 - Maquetter des interfaces utilisateur web ou web mobile

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

Dans le cadre du développement de mon projet MeetTrip, une application web permettant à des voyageuses et voyageurs de se rencontrer autour d'activités, j'ai commencée par concevoir les zonings et les maquettes complètes de l'application afin d'anticiper la structure des pages et l'ergonomie du site.

J'ai d'abord réalisé des zonings sur Figma, pour définir l'agencement des éléments principaux de chaque page (navbar, sections de contenu, images, boutons et pied de page).

Ces zonings m'ont permise de réfléchir à la hiérarchie visuelle et à la disposition des blocs avant de passer à la phase graphique.

Ensuite, j'ai transformé ces zonings en maquettes finales, en y intégrant la charte graphique de MeetTrip : logo, couleurs, typographies et visuels représentatifs des activités.

Je me suis inspirée d'interfaces modernes pour créer une expérience claire, conviviale et cohérente avec l'esprit du projet.

Les pages principales que j'ai maquettées sont : la page d'accueil, la liste des activités, la fiche activité, la création d'une activité et le profil utilisatrice.

J'ai déclinée chaque écran aux formats mobile, tablette et desktop, en suivant une approche mobile-first, afin d'assurer une expérience fluide sur tous les supports.

Avant de passer au maquetage, j'ai rédigé plusieurs user stories pour identifier les besoins essentiels des différents profils d'utilisatrices et d'utilisateurs.

Exemples de user stories du projet MeetTrip

Utilisateurs

En tant que	Je veux	Afin de
Voyageur	Créer un compte	Accéder à la plateforme et participer à des activités.
Voyageur	Rejoindre une activité	Rencontrer d'autres voyageurs.
Organisateur	Publier une activité	Proposer un événement local.
Organisateur	Consulter mes réservations	Gérer les participants.

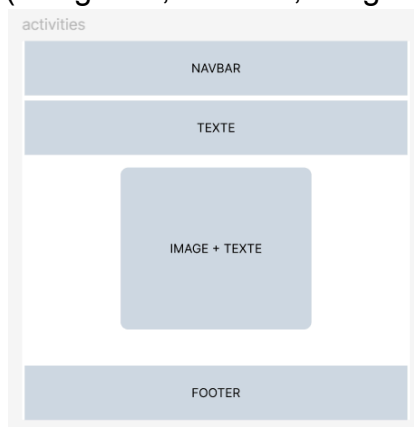
Administrateur

En tant que	Je veux	Afin de
Administrateur	Accéder au tableau de bord	Gérer les utilisateurs et les activités.
Administrateur	Supprimer une activité	Modérer le contenu de la plateforme.

Ces user stories m'ont servi de base pour concevoir des maquettes fonctionnelles et adaptées à chaque rôle, puis pour guider l'intégration avec React, Tailwind CSS et [Inertia.js](#).

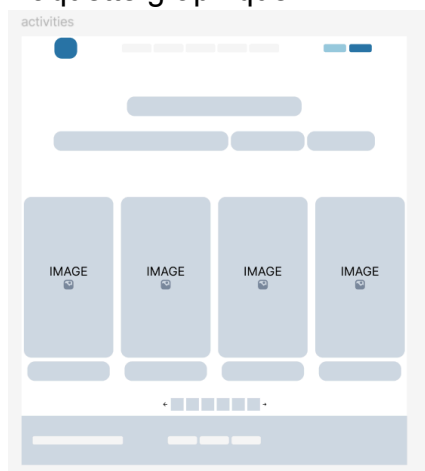
Zoning — Structure de la page “Activités”

Première ébauche servant à définir la disposition générale des zones principales (navigation, contenu, images, texte, pied de page).



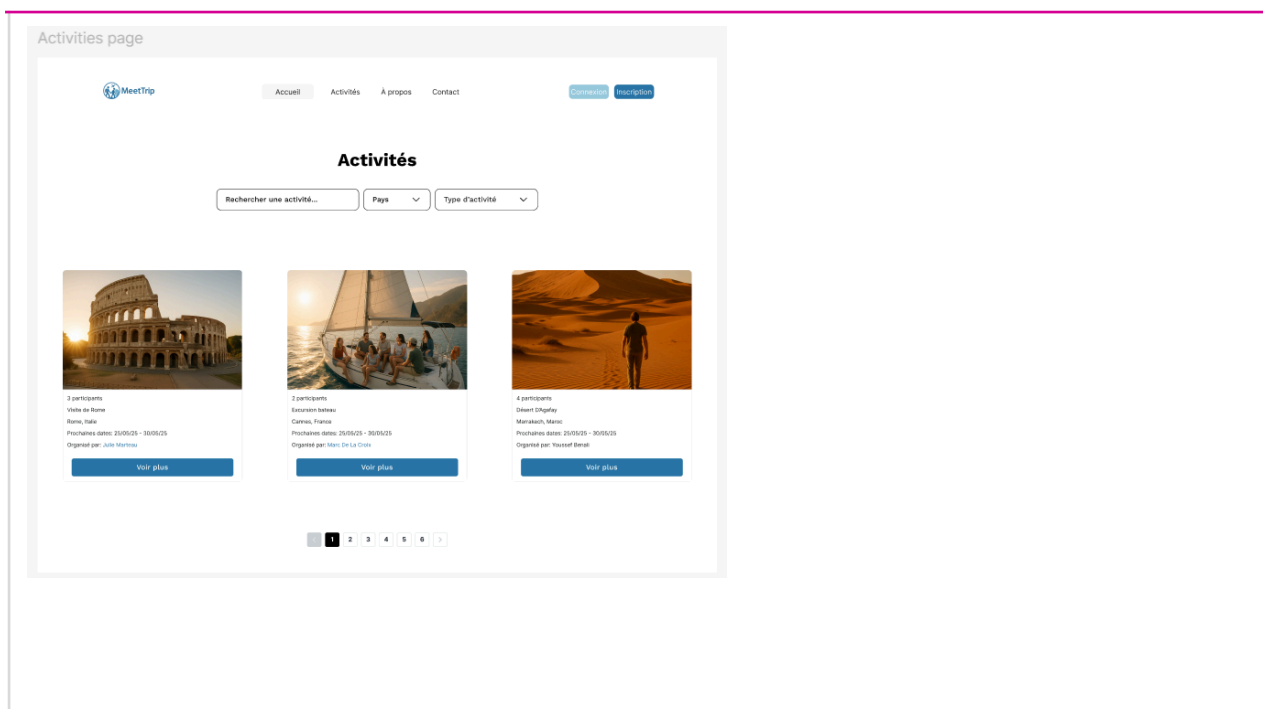
Wireframe — Mise en page fonctionnelle

Version intermédiaire permettant de préciser la hiérarchie des éléments, les interactions (filtres, boutons, pagination) et la structure visuelle avant le passage à la maquette graphique.



Maquette finale — Page “Activités”

Version finalisée intégrant la charte graphique de MeetTrip (logo, couleurs, typographies et visuels). Elle illustre le rendu complet avant intégration avec React et Tailwind CSS.



2. Précisez les moyens utilisés :

Pour réaliser les maquettes de l'application MeetTrip, j'ai utilisé Figma, un outil en ligne de conception d'interfaces. Figma m'a permis de concevoir tous les écrans de manière responsive (mobile, tablette, desktop), en appliquant directement les couleurs, typographies et éléments graphiques que je souhaitais intégrer dans le site.

J'ai également utilisé Figma pour construire une bibliothèque de composants réutilisables (boutons, cartes d'activités, champs de formulaire), ce qui m'a permis de garder une cohérence visuelle tout au long du projet.

3. Avec qui avez-vous travaillé ?

J'ai réalisée ce projet seule, dans le cadre de ma formation

4. Contexte

Nom de l'entreprise, organisme ou association		Ecole La Plateforme
Chantier, atelier, service	-	Projet personnel réalisé en cours de formation
Période d'exercice	-	Du 07/04/2025 au 07/07/2025 5

5. Informations complémentaires (facultatif)

Cliquez ici pour taper du texte.

Activité-type 1 Développer la partie front-end d'une application web ou web mobile sécurisée

CP3 - Réaliser des interfaces utilisateur statiques web ou web mobile

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

En reprenant les maquettes réalisées sur Figma ainsi que les user stories définies en amont, je suis passée à la phase d'intégration statique de l'interface de l'application MeetTrip. L'objectif de cette étape était de reproduire fidèlement l'interface utilisatrice, en respectant les contraintes de responsive design, de lisibilité et de cohérence graphique. Le site devait être utilisable sur mobile, tablette et desktop, tout en suivant une approche mobile-first. J'ai utilisée React pour structurer l'interface en composants réutilisables (Header, Footer, Carte d'activité, etc.). Pour le style, j'ai intégrée Tailwind CSS, un framework utilitaire qui permet de gérer facilement la réactivité, les espacements, les typographies et les couleurs, tout en maintenant un code clair et organisé. À ce stade, l'interface était statique, sans logique dynamique. L'objectif principal était de valider l'ergonomie, la hiérarchie des contenus et l'aspect visuel. Chaque composant a été testé dans le navigateur afin d'en vérifier le rendu et le comportement sur différents écrans. Cette phase a été réalisée en autonomie, à partir des maquettes Figma conçues précédemment, et a servi de base pour la suite du développement dynamique.

1. Structure des pages

Pour structurer les pages statiques du projet MeetTrip, j'ai utilisée React en combinant des composants organisés de manière claire, selon une logique sémantique respectant l'accessibilité.

Chaque composant correspond à une section de l'application (accueil, carte activité, profil, formulaire, etc.) et utilise des balises structurantes comme <header>, <main>, <section> et <article>.

La structure générale de chaque page a été pensée pour séparer la navigation, le contenu principal et les appels à l'action, ce qui facilite la lecture du code et améliore l'expérience utilisatrice.

2. Responsive et media queries

Pour garantir un affichage adapté à tous les supports (mobile, tablette, desktop), j'ai utilisée Tailwind CSS avec ses classes responsives intégrées (sm:, md:, lg:).

J'ai adopté une approche mobile-first, en commençant l'intégration par les petits écrans avant d'ajuster progressivement la mise en page pour les écrans plus grands.

Les tests ont été effectués sur Chrome, Firefox et Edge afin de valider le bon rendu des composants et la fluidité du design sur différents navigateurs.

3. Accessibilité et navigation

J'ai également portée une attention particulière à l'accessibilité, en suivant les recommandations du RGAA.

J'ai utilisé Lighthouse (Chrome DevTools) pour auditer les pages et détecter les points d'amélioration.

Les correctifs appliqués concernent :

- l'ajout d'attributs alt sur toutes les images,
- la structure logique des titres (h1, h2, h3),
- le contraste des couleurs et la visibilité du focus clavier.

La navigation principale est fixée en haut de l'écran et optimisée pour les petits écrans grâce à un menu hamburger.

Ce menu peut être ouvert ou fermé à l'aide d'une petite logique React, qui modifie l'état du composant selon l'interaction de l'utilisatrice.

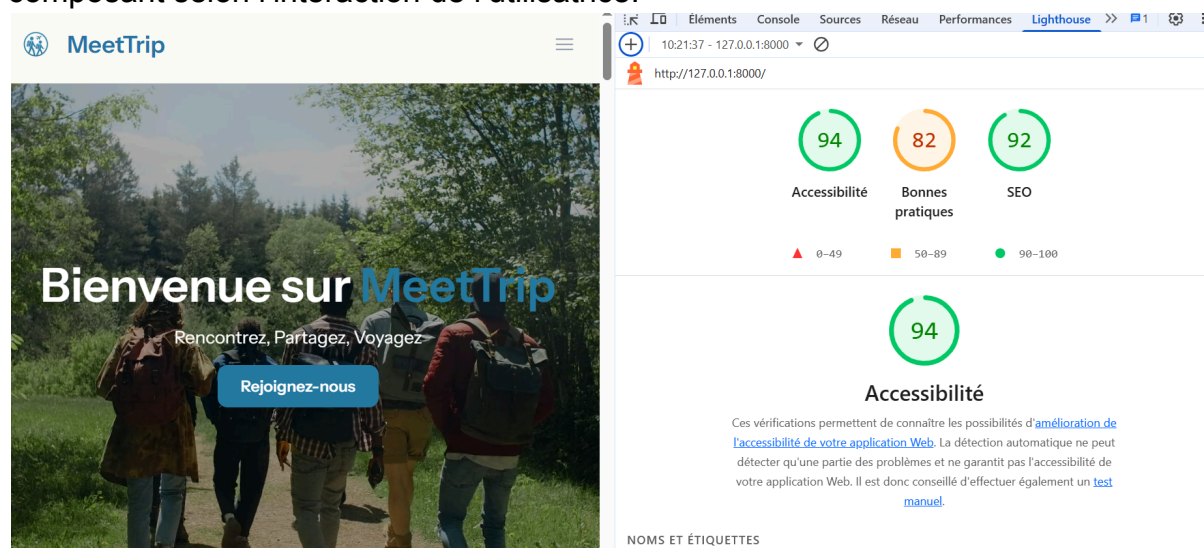


Illustration : maquette de la page d'accueil "MeetTrip" (à gauche) et résultat de l'audit Lighthouse montrant un score d'accessibilité de 94 % (à droite).

2. Précisez les moyens utilisés :

Pour réaliser ce projet, j'ai utilisé Visual Studio Code avec les extensions Prettier (pour le formatage), Tailwind CSS IntelliSense, PHP Intelephense et Database Client.

Le code a été structuré sous forme de composants React, stylisés avec Tailwind CSS.

J'ai utilisé la console de développement du navigateur pour vérifier le rendu des composants, tester l'affichage responsive, et ajuster les styles en direct.

Pour m'assurer du bon affichage sur différents formats (mobile, tablette, desktop), j'ai simulé les résolutions via les outils intégrés à Chrome.

Je me suis également appuyé sur la documentation officielle de Tailwind CSS et de React, ainsi que sur MDN Web Docs pour les bonnes pratiques HTML et l'accessibilité.

3. Avec qui avez-vous travaillé ?

J'ai réalisée ce projet seule, dans le cadre de ma formation

4. Contexte

Nom de l'entreprise, organisme ou association - Ecole la plateforme

Chantier, atelier, service - Projet personnel réalisé en cours de formation

Période d'exercice - Du : 07/04/2025 au : 07/07/2025

5. Informations complémentaires (facultatif)

Activité-type 1 Cliquez ici pour entrer l'intitulé de l'activité

CP 4 - Développer la partie dynamique des interfaces utilisateur web ou web mobile

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

Dans le cadre de mon projet professionnel, j'ai développée la partie dynamique de l'application MeetTrip, qui permet aux utilisatrices et utilisateurs connectés d'interagir avec les principales fonctionnalités de la plateforme : consulter les activités, créer leur propre activité et rejoindre celles proposées par d'autres.

Cette logique dynamique repose sur la communication entre le front-end (React avec Inertia.js) et le back-end Laravel.

Pour réaliser cette partie, j'ai utilisée plusieurs outils et bibliothèques adaptés au projet :

- Gestion des routes : les routes sont définies côté Laravel, mais accessibles depuis React via Inertia, assurant une navigation fluide sans rechargement de page.
- Gestion des requêtes : envoi de requêtes POST / GET / DELETE à l'aide des helpers Inertia pour créer, modifier ou supprimer des activités et enregistrer des utilisatrices.
- Gestion des formulaires : les formulaires de création d'activité, d'inscription et de connexion sont gérés avec des composants contrôlés React. Les erreurs de validation sont récupérées depuis Laravel et affichées automatiquement dans le formulaire.
- Affichage conditionnel : certains boutons ou éléments (comme "Participer à une activité") ne s'affichent que si l'utilisatrice est connectée.

2. Exemple : formulaire de création d'activité

J'ai créée un formulaire dynamique permettant à une utilisatrice connectée de publier une activité (lieu, description, nombre de participantes, dates, etc.).

Lors de la soumission du formulaire, une requête POST est envoyée à Laravel via Inertia.

En cas d'erreur, les messages renvoyés par le serveur sont affichés automatiquement dans les champs concernés.

L'état du formulaire est géré avec useState, et la redirection se fait vers la page d'accueil ou une page de confirmation après succès.

Ce fonctionnement offre une expérience fluide sans rechargement complet de la page.

3. Mise à jour des données affichées

Après certaines actions (création ou suppression d'une activité), les composants React sont mis à jour automatiquement grâce au système de props d'Inertia, qui recharge les données depuis Laravel sans rechargement complet.

Cette approche permet de maintenir une expérience utilisatrice fluide, réactive et moderne.

2. Précisez les moyens utilisés :

Pour ce projet, j'ai utilisé Vite avec React et Inertia.js, ainsi que la documentation officielle des dépendances utilisées, notamment celle de Laravel, React et Inertia. Le projet a été développé dans VSCode avec les extensions Prettier, PHP Intelephense, et Tailwind CSS IntelliSense.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seule sur cette partie du projet, en autonomie, avec un appui ponctuel de mon formateur en cas de blocage.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **Ecole la plateforme**

Chantier, atelier, service ▶ **Projet personnel réalisé en cours de formation**

Période d'exercice ▶ **Du : 07/04/2025 au : 07/07/2025**

5. Informations complémentaires (facultatif)

Activité-type 2 Cliquez ici pour entrer l'intitulé de l'activité

CP 5 - Mettre en place une base de données relationnelle

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

Dans le cadre du projet MeetTrip, j'ai conçu une base de données relationnelle afin de structurer les données des utilisatrices, des activités proposées, des lieux et des participations.

J'ai commencé par réaliser un MCD (Modèle Conceptuel de Données) pour identifier les entités principales et leurs relations :

- Une utilisatrice peut créer plusieurs activités.
- Une activité est liée à un lieu et peut accueillir plusieurs participantes.
- La participation d'une utilisatrice à une activité est gérée via une table pivot.

À partir du MCD, j'ai ensuite définie un MLD (Modèle Logique de Données) pour traduire les relations sous forme de tables et de clés étrangères.

Plutôt que de créer un MPD séparé, j'ai directement implémentée la base dans Laravel grâce aux migrations, en générant un fichier pour chaque table :

users, activities, locations, activity_user (table pivot), etc.

Cette démarche s'appuie sur une logique de modélisation inspirée de la méthode MERISE, adaptée ici aux outils modernes de Laravel.

Elle permet de maintenir la cohérence du schéma relationnel tout en facilitant les futures évolutions du projet.

Pour tester le bon fonctionnement de la base, j'ai utilisée des seeders afin d'insérer des données fictives (utilisatrices, activités, lieux).

Ces données de test m'ont permis de vérifier les relations entre les tables et la cohérence du modèle dans mon environnement local (SQLite).

2. Précisez les moyens utilisés :

Pour concevoir la base de données, j'ai utilisée Laravel avec ses outils de migrations et de seeders, et la base a été gérée en SQLite pour le développement local.

J'ai réalisé les MCD et MLD sur Figma, puis je les ai traduits en fichiers de migration Laravel pour construire la structure finale des tables et leurs relations.

J'ai également utilisée l'extension Database Client dans Visual Studio Code pour visualiser les tables, exécuter des requêtes SQL et vérifier la cohérence des données pendant le développement.

3. Avec qui avez-vous travaillé ?

J'ai travaillée seule sur cette partie, en autonomie, tout en sollicitant ponctuellement mon formateur pour valider la structure de la base de données.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **Ecole la plateforme**

Chantier, atelier, service ▶ **Projet personnel réalisé en cours de formation**

Période d'exercice ▶ **Du : 07/04/2025 au : 07/07/2025**

5. Informations complémentaires *(facultatif)*

Activité-type 2 Cliquez ici pour entrer l'intitulé de l'activité

CP 6 - Développer les composants d'accès aux données SQL et NoSQL.

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

Pour cette partie du projet, j'ai développé les composants d'accès aux données en utilisant Eloquent, l'ORM intégré à Laravel.

L'objectif était de gérer facilement les interactions avec la base de données relationnelle (SQLite) sans écrire directement de requêtes SQL.

J'ai créé les modèles Laravel pour les différentes entités : User, Activity, Location, et mis en place les relations entre elles (hasMany, belongsTo, belongsToMany).

Par exemple, une activité est liée à un utilisateur (organisateur) et peut avoir plusieurs participants via une table pivot.

J'ai utilisé les méthodes d'Eloquent pour :

- Récupérer les activités créées par un utilisateur (\$user->activities)
- Ajouter ou supprimer des participants (\$activity->participants()->attach(\$user_id))
- Charger des relations avec with() pour optimiser les requêtes

Tout ce traitement des données a été fait en backend Laravel, puis affiché côté React via Inertia.js..

2. Précisez les moyens utilisés :

Pour cette partie, j'ai utilisé Laravel avec Eloquent ORM pour accéder aux données.

J'ai utilisé SQLite comme base de données locale, et les relations entre modèles ont été définies directement dans les fichiers User.php et Activity.php.

Les requêtes ont été testées via les contrôleurs Laravel et la console de développement du navigateur pour vérifier le rendu côté React via Inertia.js.

3. Avec qui avez-vous travaillé ?

J'ai travaillée seule sur cette partie, en autonomie, tout en m'appuyant sur les documentations officielles de Laravel et Inertia.js.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **Ecole la plateforme**

Chantier, atelier, service ▶ **Projet personnel réalisé en cours de formation**

Période d'exercice ▶ **Du : 07/04/2025 au : 07/07/2025**

5. Informations complémentaires *(facultatif)*

Activité-type 2 Cliquez ici pour entrer l'intitulé de l'activité

CP 7 - Développer des composants métier côté serveur.

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

Dans le cadre du projet MeetTrip, j'ai développée plusieurs composants métier côté serveur à l'aide du framework Laravel.

Ces composants permettent de gérer la logique d'inscription, de création et d'affichage des activités, ainsi que les règles métier liées aux rôles des utilisateurs.

J'ai mis en place des contrôleurs Laravel comme ActivityController, pour gérer la création, la mise à jour et la suppression d'activités.

Chaque méthode intègre :

- La validation des données avec `$request->validate()`
- Les actions métier (ex. : enregistrer une activité, associer un utilisateur, gérer les participations)
- La redirection ou la réponse avec message selon le résultat

Pour sécuriser les accès, j'ai utilisé des middleware Laravel, notamment :

- `auth` : pour vérifier qu'un utilisateur est connecté avant d'accéder à certaines routes
- `verified` : pour s'assurer que l'email a été vérifié
- Et un contrôle de rôle dans les contrôleurs pour restreindre certaines actions aux organisateurs

2. Précisez les moyens utilisés :

Pour cette partie, j'ai utilisé Laravel pour structurer la logique métier dans les contrôleurs, modèles et middleware.

Les validations ont été faites avec `$request->validate()`, et les accès sécurisés via les middleware `auth` et `verified`.

Le tout a été développé dans VSCode, avec les extensions Prettier, PHP Intelephense et testé via Inertia.js côté front.

3. Avec qui avez-vous travaillé ?

J'ai travaillée seule sur cette partie du projet, en autonomie, tout en échangeant avec mon formateur pour valider certains choix de logique métier et de sécurité.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **Ecole la plateforme**

Chantier, atelier, service ▶ **Projet personnel réalisé en cours de formation**

Période d'exercice ▶ **Du : 07/04/2025 au : 07/07/2025**

5. Informations complémentaires *(facultatif)*

Activité-type 2 Cliquez ici pour entrer l'intitulé de l'activité

CP 8 - Documenter le déploiement d'une application dynamique web ou web mobile.

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

Le déploiement de l'application MeetTrip a été effectué à l'aide de Docker afin de reproduire un environnement de production complet.

J'ai configuré plusieurs conteneurs indépendants pour isoler les différents services nécessaires au bon fonctionnement du projet :

- PHP / Laravel pour la partie back-end,
- MySQL pour la base de données,
- Nginx pour le serveur web,
- Node.js pour la partie front-end avec Inertia.js.

L'ensemble est automatisé avec Docker Compose, ce qui permet de lancer ou d'arrêter tous les conteneurs via une seule commande.

Cette solution garantit une application stable, portable et facilement exécutable sur n'importe quel système, en reproduisant des conditions réelles d'hébergement.

1^{re} étape : Guide d'installation et de lancement

Le déploiement de MeetTrip s'effectue à l'aide de Docker Compose.

Voici la procédure complète permettant de lancer le projet en local ou sur un serveur de préproduction :

1) Récupérer le projet

```
git clone <url_du_repo> && cd meettrip
```

2) Choisir l'environnement (ex : local)

```
cp .env.local .env
```

3) Lancer l'infrastructure

```
docker compose up -d --build
```

4) Préparer l'application

```
docker compose exec app composer install
```

```
docker compose exec app php artisan key:generate
```

5) Créer la base et insérer les données de test

```
docker compose exec app php artisan migrate --force --seed
```

6) Finaliser la configuration

```
docker compose exec app php artisan storage:link
```

docker compose exec app php artisan optimize:clear

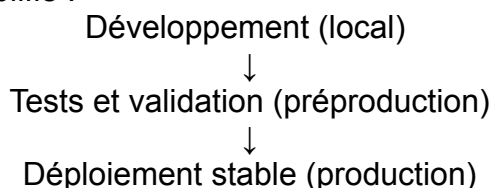
Ces commandes permettent de cloner le projet, lancer les conteneurs, installer les dépendances, exécuter les migrations et alimenter la base de données avec des données de test automatiques.

2^e étape : Schéma des environnements et critères de validation

Afin d'assurer la fiabilité, la sécurité et la continuité du projet, MeetTrip fonctionne sur trois environnements distincts :

Environnement	Objectif	Accès / URL	Particularités techniques
Local (dev)	Développement individuel	http://localhost:8080	Docker, base MySQL locale, APP_DEBUG=true
Préproduction (preprod)	Tests avant mise en ligne	https://preprod.meettrip.test	Données de test, cache activé, debug désactivé
Production (prod)	Environnement final	https://meettrip.fr	HTTPS, APP_DEBUG=false, monitoring et backup actifs

Cycle de déploiement simplifié :



3^e étape : Critères “Go / No-Go” et procédure de rollback

Avant chaque déploiement, plusieurs points de contrôle garantissent la stabilité du projet :

Étape	Critère “Go” (OK pour déploiement)	Critère “No-Go” (bloquant)
Tests unitaires / fonctionnels	100 % des tests critiques passent (php artisan test)	Tests échoués sur routes principales ou formulaires
Accessibilité & performances	Score Lighthouse ≥ 90	Score < 80 ou erreurs ARIA
Migration & seed	Aucune erreur sur php artisan migrate --seed	Conflits ou rollback forcé
Interface / affichage	Aucune anomalie visuelle majeure	Bugs d'affichage ou crash JS
Rollback possible	Dump SQL de secours disponible	Pas de backup récent

Procédure de rollback en cas d'échec du déploiement :

Restaurer la version stable

```
git checkout main && git reset --hard <commit_stable>
```

Réappliquer la base de données sauvegardée

```
docker compose exec db sh -c "mysql -umeettrip -psecret meettrip <  
/backups/meettrip_backup.sql"
```

Relancer l'application

```
docker compose up -d --build
```

Ce processus de déploiement progressif assure la stabilité du projet MeetTrip, tout en permettant de tester, corriger et valider chaque version avant sa mise en production finale.

2. Précisez les moyens utilisés :

Outils : Docker, Docker Compose, Git, GitHub, Visual Studio Code

Technologies : PHP (Laravel), MySQL, Node.js, Inertia.js, Nginx

Environnement : Windows 10, WSL2

Commandes clés :

- docker-compose up -d pour le lancement des services
- docker exec -it app bash pour accéder au conteneur applicatif
- php artisan migrate pour exécuter les migrations de la base de données

3. Avec qui avez-vous travaillé ?

J'ai travaillée de manière autonome, avec l'appui ponctuel de mes formateur pour la configuration du fichier docker-compose.yml et la gestion des volumes persistants.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Ecole la plateforme

Chantier, atelier, service ▶ Projet personnel réalisé en cours de formation

Période d'exercice ▶ Du : 07/04/2025 au : 07/07/2025

5. Informations complémentaires (facultatif)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.

Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] **Boujnah Nadia** ,
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 **Cannes** le **07/07/2025**
pour faire valoir ce que de droit.

Signature :

Boujnah Nadia

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé
Cliquez ici pour taper du texte.

ANNEXES

(Si le RC le prévoit)