



DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ
DE L'EMPLOI

Nom de naissance

- Boucetta

Nom d'usage

- Boucetta

Prénom

- Yacine

Adresse

- 6 rue marechal foch 13600 la ciotat

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.

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

DOSSIER PROFESSIONNEL (DP)

Sommaire

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

Exemples de pratique professionnelle

1 Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité p.

- Maquetter une application : Share Event application de gestion d'événement et de centralisation de photo p.
- Développer une interface utilisateur de type desktop : Segmentation client Richardson p.
- Développer des composants d'accès aux données : Share Event partie back-end p.
- Développer la partie front-end d'une interface utilisateur web : page admin share event p.
- Développer la partie back-end d'une interface utilisateur web : Share Event parti back end p.

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité p.

- Concevoir une base de données : base de données share Event p. p.
- Mettre en place une base de données : base de données Share Event p.
- Développer des composants dans le langage d'une base de données : nest js type orm p. p.

Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité p.

- *Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement* p.
- Concevoir une application: ShareEvent p.
- Développer des composants métier: ShareEvent p.
- Construire une application organisée en couches p.
- Développer une application mobile: ShareEvent p.
- Préparer et exécuter les plans de tests d'une application: p.
- Préparer et exécuter le déploiement d'une application p.

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

Déclaration sur l'honneur p.

Documents illustrant la pratique professionnelle (*facultatif*) p.

Annexes (*Si le RC le prévoit*) p.

DOSSIER PROFESSIONNEL (DP)

DOSSIER PROFESSIONNEL ^(DP)

**EXEMPLES DE PRATIQUE
PROFESSIONNELLE**

DOSSIER PROFESSIONNEL (DP)

Activité-type 1

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

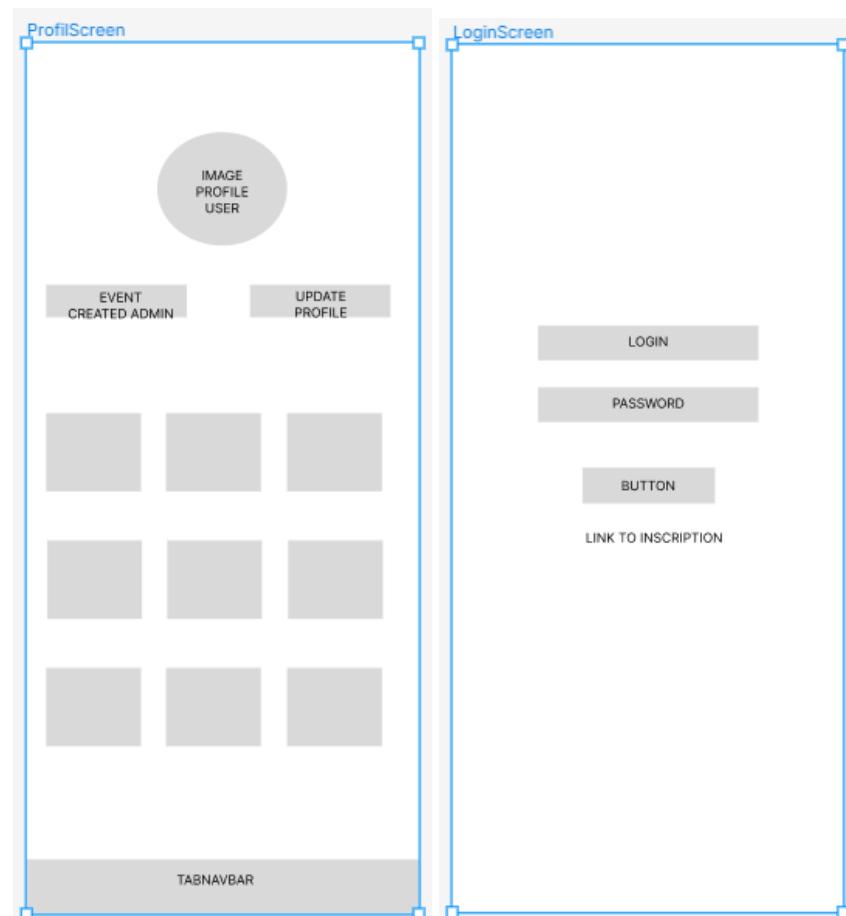
- Exemple n°1* - Maquetter une application : Share Event application de gestion d'événement et de centralisation de photo

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

Pour développer la partie front end de notre application il était primordiale de conceptualiser en amont l'application ,pour ce faire nous avons utilisé figma :

-Dans un premier temps nous avons fourni une maquette basse fidélité (zoning) qui nous permet de placer les éléments et de voir la pertinence de leur emplacement

-Dans un deuxième temps nous créons la maquette haute fidélité qui sera une image exacte de l'application qui sera créée.



DOSSIER PROFESSIONNEL (DP)

Police pour les titres et entête : CINZEL
Police pour le reste du texte : open sans
pour le code couleur:

Couleur principale : #97C17E



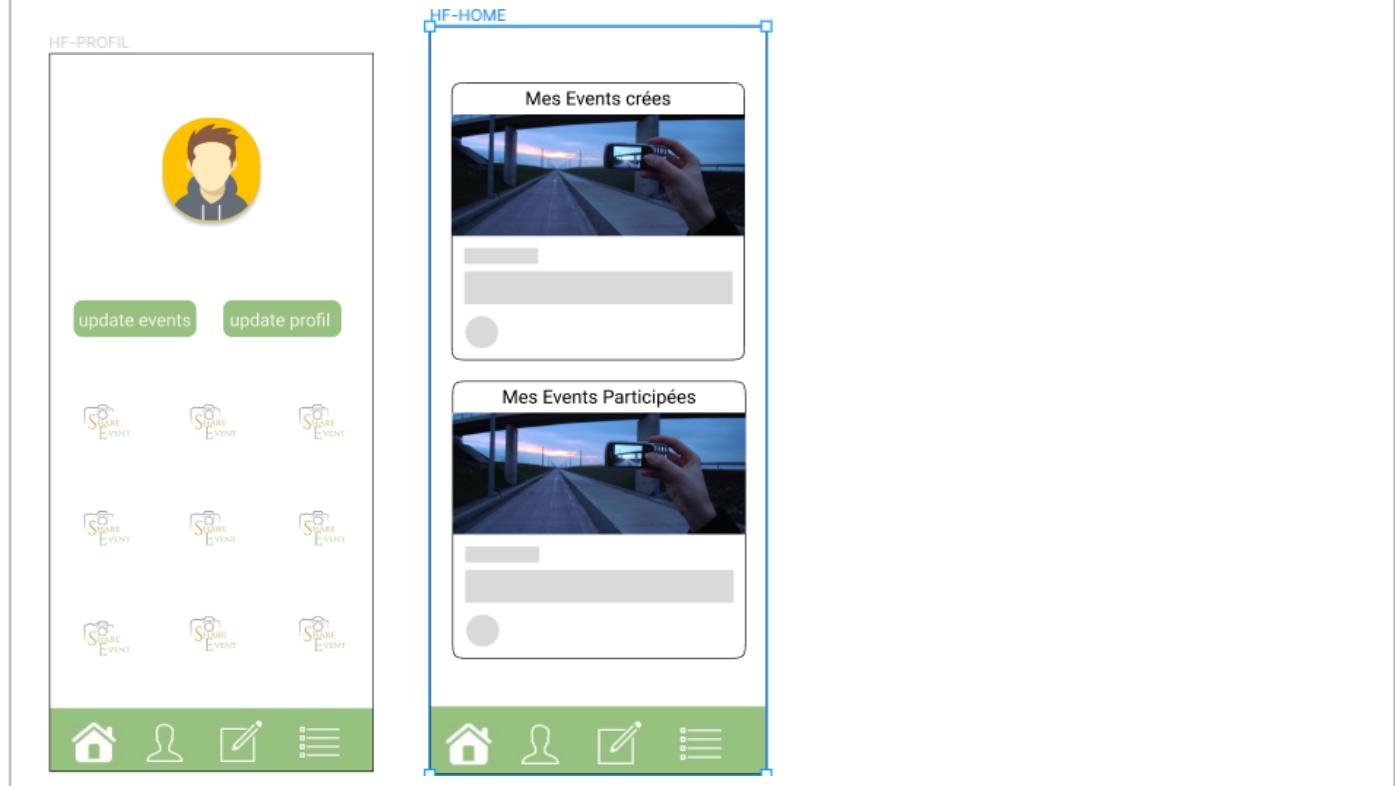
#E6C821



couleur secondaire:#FCFCFC



La charte graphique définie nous passons à la maquette haute fidélité qui sera le rendu finale de l'app



2. Précisez les moyens utilisés :

figma

3. Avec qui avez-vous travaillé ?

Raphael diop, Damien Verschaere

DOSSIER PROFESSIONNEL (DP)

4. Contexte

Nom de l'entreprise, organisme ou association ➤ *La plateforme.*

Chantier, atelier, service ➤ *Cliquez ici pour taper du texte.*

Période d'exercice ➤ Du : **06/09/2023** au : **31/08/2023**

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Activité-type 1

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n°1 - Développer une interface utilisateur de type desktop : Segmentation client Richardson

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

Dans le cadre de mon alternance il m'a été confié de créer pour le pôle marketing un client lourd (une interface utilisateur de type desktop) pour pouvoir cibler l'ensemble des client de richardson en fonction de plusieurs filtres (en fonction de leur position géographique , taille entreprise etc).

Pour ce faire j'ai utilisé le framework.net et Windows Presentation Foundation (WPF) qui permet d'avoir une interface graphique grâce au xaml (une forme de xml amélioré)

j'ai utilisé le langage c# avec les requêtes linq pour pouvoir filtrer.

ce desktop permet de parcourir une base de donnée récupérer les information de client et avec linq de filtrer les client dont on a besoin ils sont ensuite mis dans une datagrid.

vous avez la possibilité d'extraire les données sous forme de fichier csv

Code agen	Nom agen	Code agen	Nom agen	Code clien	fermé	Nom Client	adresse cli	Ville client	Siren Client	Email	Telephone	Type Entre	Taille Entre	Phase de v	Distance	Siret	Siret Exact	nom sociét
001	AUBAGNE	001	AUBAGNE	00183	False	BLANC GEF	13830	ROQUEFOU	330243445						Indétermin		False	
001	AUBAGNE	001	AUBAGNE	55022	False	LIONEL BO	13600	LA CIOTAT	483590501	lionel.bourc	061569295				Indétermin		False	
001	AUBAGNE	001	AUBAGNE	02504	False	SARL MATK	13011	MARSEILLE	417672995	matkis@ne					Indétermin		False	
001	AUBAGNE	001	AUBAGNE	42205	True	FTEC	13420	GEMENOS	813882289	fred.richard	073287061				Indétermin		False	
001	AUBAGNE	001	AUBAGNE	00620	False	SOUDÉCOL	13120	GARDANNI	055802995	contact@sc		Société	Entreprise	Stabilisatio	Intermédiai	055802995	True	SOC PROVI
001	AUBAGNE	001	AUBAGNE	41949	True	SPECIES SA	13781	AUBAGNE	055804512	Fspecies@e	062313356	Société	Entreprise	Stabilisatio	Intermédiai	055804512	False	SPECIES
001	AUBAGNE	001	AUBAGNE	41415	False	SNEF	13015	MARSEILLE	056800659			Société	Très grande	Stabilisatio	Intermédiai	056800659	False	SNEF
001	AUBAGNE	001	AUBAGNE	46616	False	SNEF CLIM	13010	MARSEILLE	056800659			Société	Très grande	Stabilisatio	Intermédiai	056800659	False	SNEF
001	AUBAGNE	001	AUBAGNE	49681	False	SNEF TERTI	13010	MARSEILLE	056800659			Société	Très grande	Stabilisatio	Intermédiai	056800659	False	SNEF
001	AUBAGNE	001	AUBAGNE	45176	False	SNEF BERR	13130	BERRE L'E	056800659			Société	Très grande	Stabilisatio	Eloigné (>2	056800659	True	SNEF
001	AUBAGNE	001	AUBAGNE	00093	False	ETS SOUCI	13001	MARSEILLE	056806474	souchon13	062273669	Société	Entreprise	Stabilisatio	Intermédiai	056806474	True	ETABLISSEN
001	AUBAGNE	001	AUBAGNE	92004	False	STE EAUX L	13395	MARSEILLE	057806150			Société	Très grande	Stabilisatio	Intermédiai	057806150	False	SOCIETE DI
001	AUBAGNE	001	AUBAGNE	75940	False	LES FILS DE	13400	AUBAGNE	058800475	brice.willefr		Société	Entreprise	Stabilisatio	Intermédiai	058800475	True	LES FILS DE
001	AUBAGNE	001	AUBAGNE	46276	False	PROX-HYD	13456	MARSEILLE	058812397	sfile@prox		Société	Très grande	Stabilisatio	Intermédiai	058812397	False	PROX-HYD
001	AUBAGNE	001	AUBAGNE	00749	False	HOTEL DE L	13714	CASSIS	059801993	info@hotel	060997036	Société	Structure lé	En déclin	Proximité (-	059801993	True	LA PLAGE
001	AUBAGNE	001	AUBAGNE	92003	False	RTM REGIE	13235	MARSEILLE	059804062			Société	Très grande	Stabilisatio	Intermédiai	059804062	True	REGIE DES
001	AUBAGNE	001	AUBAGNE	01291	False	SODEXAL	13012	MARSEILLE	060803335	sodexal@w		Société	Entreprise	Stabilisatio	Intermédiai	060803335	True	SODEXAL
001	AUBAGNE	001	AUBAGNE	01356	False	ETS H SAIN	13013	MARSEILLE	061802070	marcel.pau		Société	Société hié	Stabilisatio	Intermédiai	061802070	True	ETABLISSEN
001	AUBAGNE	001	AUBAGNE	00976	False	COMEX SA	13275	MARSEILLE	062802459	l.negrel@cc		Société	Entreprise	Stabilisatio	Intermédiai	062802459	False	COMPAGN

2. Précisez les moyens utilisés :

net c# WPF LINQ

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

4. Contexte

Nom de l'entreprise, organisme ou association ➤ **Richardson.**

Chantier, atelier, service ➤ *Cliquez ici pour taper du texte.*

Période d'exercice ➤ Du : **06/09/2022** au : **31/08/2023**

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Activité-type 1

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n°1 ▶ Développer des composants d'accès aux données : Share Event partie back-end

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

Dans le cadre de notre formation nous avons créé une application de partage de photo nommée Share Event .Il est d'une grande importance de créer des composants d'accès au donnée . Nous avons donc créé une api avec le framework nest js qui est un framework back end qui utilise le langage typescript cela nous a permis de créer des composants d'accès au données. Nest js fonctionne avec un système de 'service' en fonction de l'url envoyé par le côté client le framework va appeler le module ,le controller et le service associé pour renvoyer les données si celle ci sont conforme , que les utilisateurs ont le droit d'accéder à ces données ou que les données envoyés par le côté client sont conforme au dto demandé.

```
@Module({
  imports: [TypeOrmModule.forFeature([User,UserEvent]),UsersModule],
  exports: [TypeOrmModule],
  providers: [UserService],
  controllers: [UsersController],
})
export class UsersModule {}
```

```
import { Controller, Post, Body, Delete } from '@nestjs/common';
import { Get, Patch, Put } from '@nestjs/common/decorators';
import { User } from './model/entities/users.entity';
import { UserService } from './users.service';
import { Param } from '@nestjs/common/decorators';
import { UpdateUserDto } from './dto/update-user.dto';
import { CreateUserData } from './dto/create-user.dto';

@Controller('users')
export class UsersController {
  constructor(private readonly userService: UserService) {}

  @Post('register')
  async register(@Body() user: CreateUserData) {
    return this.userService.create(user);
  }

  @Get()
  findAll() {
    return this.userService.findAll();
  }

  @Get()
  findOneByEmail(@Param('email') email:string){
    return this.userService.findOneByEmail(email)
  }

  @Get(':id')
  findOne(@Param('id') id: string) {
    return this.userService.findOne(+id);
  }
}

async findAll(): Promise<User[]> {
  return await this.usersRepository.find();
}

async findOneByEmail(email:string){
  return await this.usersRepository.findOneBy({email:email});
}

async findOne(id: number) {
  return await this.usersRepository.findOneBy({ id });
}

async updateUser(id: number, user: UpdateUserDto): Promise<any> {
  var passwordHash= user.password;
  if(user.password){
    passwordHash = await bcrypt.hash(user.password,10);
  }

  const updateuserDatabase = await this.usersRepository.findOneBy({id:id} );
  updateuserDatabase.id= id;
  updateuserDatabase.email=user.email;
  updateuserDatabase.isAdmin=false;
  updateuserDatabase.name=user.name;
  updateuserDatabase.url=user.url;
  updateuserDatabase.lastname=user.lastname;
  updateuserDatabase.password=passwordHash;
```

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

typescript , nest js

3. Avec qui avez-vous travaillé ?

Raphael diop, Damien Verschaere

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *La plateforme.*

Chantier, atelier, service ▶ *Cliquez ici pour taper du texte.*

Période d'exercice ▶ Du : **06/09/2023** au : **31/08/2023**

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Activité-type 1

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

- Exemple n°1** - Développer la partie front-end d'une interface utilisateur web :page admin share event

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

Dans le cadre de notre formation nous avons créé une application de partage de photos nommée Share Event .

Nous avons dû dans le cadre de ce projet créer une page admin pour gérer certains aspects de l'application.

L'interface graphique devait être créé sur une page web , c'est pour cela qu'on a utilisé angular avec la librairie material , cette librairie permet de créer des composants avec un style de manière très rapide . Nous avons utilisé angular car la logique et l'architecture des projets angular est similaire à celle de nest js .

Nous avons un component.html qui appelle le component.ts qui appelle le service associé .

Nous avons ajouté des alertes pour confirmer la suppression des événements.

The diagram illustrates the architecture of the event deletion process. It shows four main components:

- event**: A table view showing a list of events with columns: id, Name, start_date, end_date, userid, description, url_event, isPrivate, and suppression.
- component.html**: The template file containing an ng-container with matColumnDef="suppression". It includes a th for suppression and a td for each row with a button labeled "supprimer" that triggers a click event to delete the row's id.
- component.ts**: The logic file that handles the deletion. It contains a delete(event: Event) method. This method sets isWaitingForServerResponse to true, calls deleteEvent on the eventsService, pipes the result through catchError and subscribe, and then handles the response by setting isWaitingForServerResponse to false and calling handleSuccess.
- event service.ts**: The service file that performs the actual deletion. It has a deleteEvent(event: Event) method. This method constructs a full URL using the baseURL and event ID, prompts a confirmation dialog, shows an alert if confirmed, and then returns a promise for the httpClient.delete call.

Red arrows indicate the flow of data and control between these components: from the table to the template, from the template to the component.ts file, and from component.ts to the event service.ts file.

id	Name	start_date	end_date	userid	description	url_event	isPrivate	suppression
85	Evg	2023-07-11	2023-07-11	37	C'est la fin de la vie de dann	file:///data/user/0/hostexp/exponent/cache/ExperienceData/%2540anonymous%252fapp-mobile-forum-7569de2e-838e-4185-a8b6-65c333eb652a/ImagePickerController/c6ab5910-c8eb-472c-ab2b-de1c9d77f1e4.jpeg	true	<button>supprimer</button>
86	Mariage de l'année	2023-07-11	2023-07-11	37	Il y a à boire et à manger	file:///data/user/0/hostexp/exponent/cache/ExperienceData/%2540anonymous%252fapp-mobile-forum-7569de2e-838e-4185-a8b6-65c333eb652a/ImagePickerController/b8b549115-476a-4f68-afaa-dcfcce2ee7b42.jpeg	false	<button>supprimer</button>
87	Fête de fin d'année	2023-07-13	2023-07-13	37	C'est la fin	file:///data/user/0/hostexp/exponent/cache/ExperienceData/%2540anonymous%252fapp-mobile-forum-7569de2e-938e-4185-a8b6-65c333eb652a/ImagePickerController/63a55963-368b-4f68-afaa-dcfcce2ee7b42.jpeg	true	<button>supprimer</button>
88	Vide grenier	2023-07-10	2023-07-10	37	Vide grenier	file:///data/user/0/hostexp/exponent/cache/ExperienceData/%2540anonymous%252fapp-mobile-forum-7569de2e-838e-4185-a8b6-65c333eb652a/ImagePickerController/b8404202-c8d2-4f11-818b-a9fc2e741999.jpeg	false	<button>supprimer</button>
89	Anniversaire de Jean Pierre	2023-07-10	2023-07-10	37	C'est la fête	file:///data/user/0/hostexp/exponent/cache/ExperienceData/%2540anonymous%252fapp-mobile-forum-7569de2e-838e-4185-a8b6-65c333eb652a/ImagePickerController/b8404202-c8d2-4f11-818b-a9fc2e741999.jpeg	true	<button>supprimer</button>

DOSSIER PROFESSIONNEL (DP)

ID	Name	start_date	end_date	userId	description	url_event	isPrivate	suppression
85	Evg	2023-07-11	2023-07-11	37	C'est la fin de la vie de dam	file:///data/user/0/host.exp.exponent/cache/ExperienceData/%2540anonymous%252Fapp-mobile-forum-7569de2a-838e-4185-ab6-65c333eb652a/ImagePicker/c0a059f0-c8eb-472d-ab2b-de1c9d7d7fa4.jpeg	true	<button>supprimer</button>
86	Mariage de l'année	2023-07-11	2023-07-11	37	Il y a à boire et à manger	file:///data/user/0/host.exp.exponent/cache/ExperienceData/%2540anonymous%252Fapp-mobile-forum-7569de2a-838e-4185-ab6-65c333eb652a/ImagePicker/bb549115-476a-40dc-a73-84fe8cc0dd1b.icea	false	<button>supprimer</button>
87	Fête de fin d'année	2023-07-13	2023-07-13	37	C'est la fin	file:///data/user/0/host.exp.exponent/cache/ExperienceData/%2540anonymous%252Fapp-mobile-forum-7569de2a-838e-4185-ab6-65c333eb652a/ImagePicker/63a55963-3688-dfbf0-a0aa-8ccce2ee7b42.jpeg	true	<button>supprimer</button>
88	Vide grenier	2023-07-13	2023-07-13	37	Vide grenier	file:///data/user/0/host.exp.exponent/cache/ExperienceData/%2540anonymous%252Fapp-mobile-forum-7569de2a-838e-4185-ab6-65c333eb652a/ImagePicker/bb8404202-c0d2-4fb1-818b-29d297d74199.jpeg	false	<button>supprimer</button>
89	Anniversaire de Jean Pierre	2023-07-13	2023-07-13	37	C'est la fête	file:///data/user/0/host.exp.exponent/cache/ExperienceData/%2540anonymous%252Fapp-mobile-forum-7569de2a-838e-4185-ab6-65c333eb652a/ImagePicker/bb8404202-c0d2-4fb1-818b-a90e28741999.jpeg	true	<button>supprimer</button>

2. Précisez les moyens utilisés :

typescript , angular,material

3. Avec qui avez-vous travaillé ?

Raphael diop, Damien Verschaere

4. Contexte

Nom de l'entreprise, organisme ou association ➔ *La plateforme.*

Chantier, atelier, service ➔ *Cliquez ici pour taper du texte.*

Période d'exercice ➔ Du : *06/09/2023* au : *31/08/2023*

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL ^(DP)

DOSSIER PROFESSIONNEL (DP)

Activité-type 1

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n°1 - Développer la partie back-end d'une interface utilisateur web : Share Event parti back end

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

Pour la partie web la même api est utilisé

Cependant la connexion à été faite via une base url

grâce à la classe httpclient et la méthode get par exemple nous pouvons connecté la partie front du back

```
export class EventService {
  private baseURL = 'http://localhost:3000/events';
  private httpHeaders = {
    headers: new HttpHeaders({ 'Content-Type': 'application/json' })
  };

  constructor(private httpClient: HttpClient) {}

  getEvent() {
    const fullURL = `${this.baseURL}`;
    return this.httpClient.get<Event>(fullURL, this.httpHeaders);
  }

  deleteEvent(event: Event) {
    const fullURL = `${this.baseURL}/${event}`;
    if(window.confirm("êtes vous sur de vouloir supprimer cette evenement?"))
    {
      window.alert("l'évenement à bien été supprimé")
      return this.httpClient.delete<Event>(fullURL, this.httpHeaders);
    }
    return this.httpClient.get<Event>(fullURL, this.httpHeaders);
  }
}
```

2. Précisez les moyens utilisés :

typescript , nest js

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

Raphael diop, Damien Verschaere

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *La plateforme.*

Chantier, atelier, service ▶ *Cliquez ici pour taper du texte.*

Période d'exercice ▶ Du : *06/09/2023* au : *31/08/2023*

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Activité-type 2

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

Exemple n° 1 - Concevoir une base de données

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

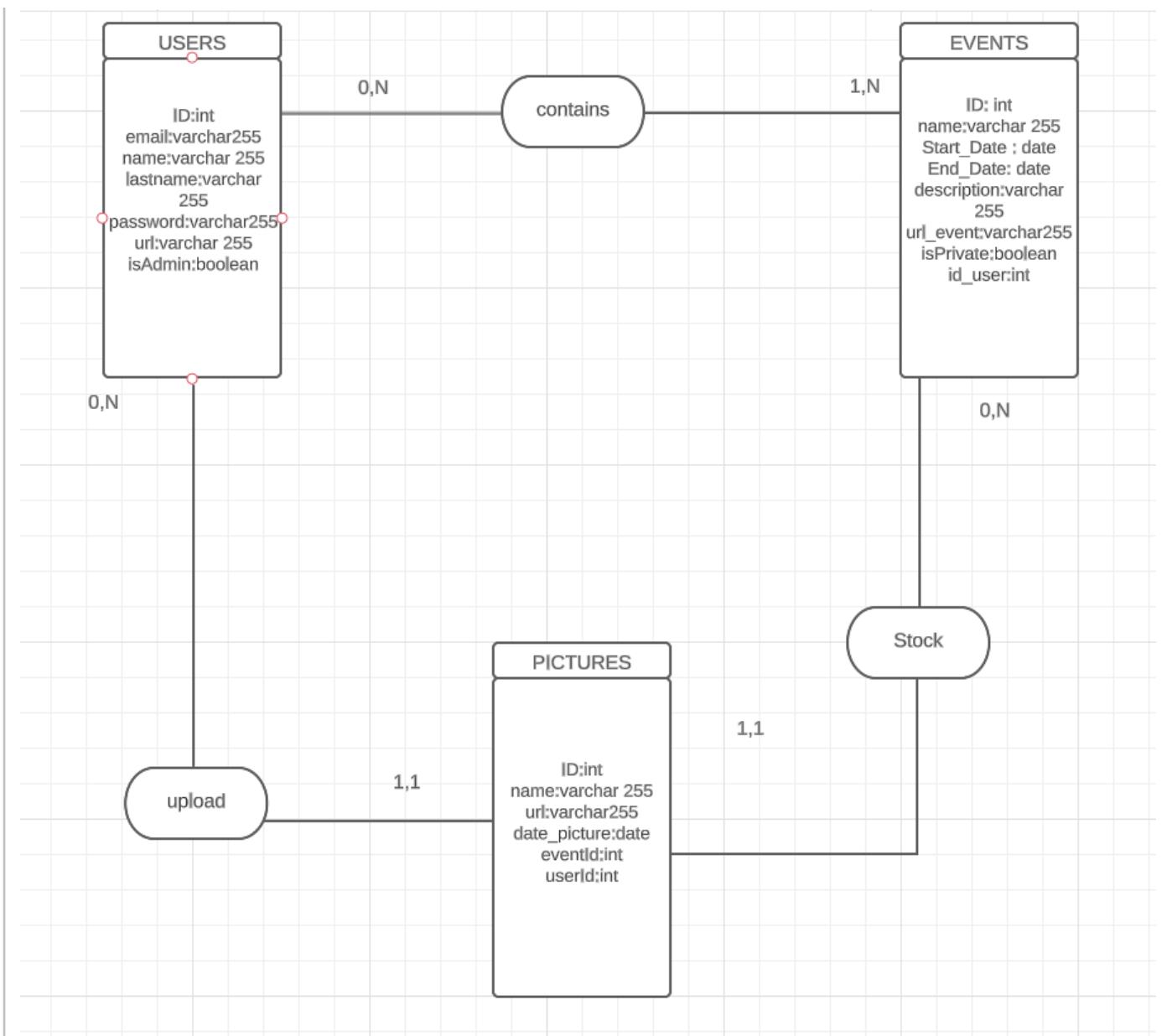
Dans le cadre de notre projet d'application mobile ShareEvent, nous avons utilisé l'outil de modélisation Lucidchart pour créer nos **modèles conceptuels de données (MCD)**, **modèles logiques de données (MLD)** et **modèles physiques de données (MPD)**.

Modèle Conceptuel de Données (MCD):

Le Modèle Conceptuel de Données est la première étape de la conception d'une base de données. Il permet de représenter de manière abstraite et simplifiée la structure des données que l'on souhaite stocker.

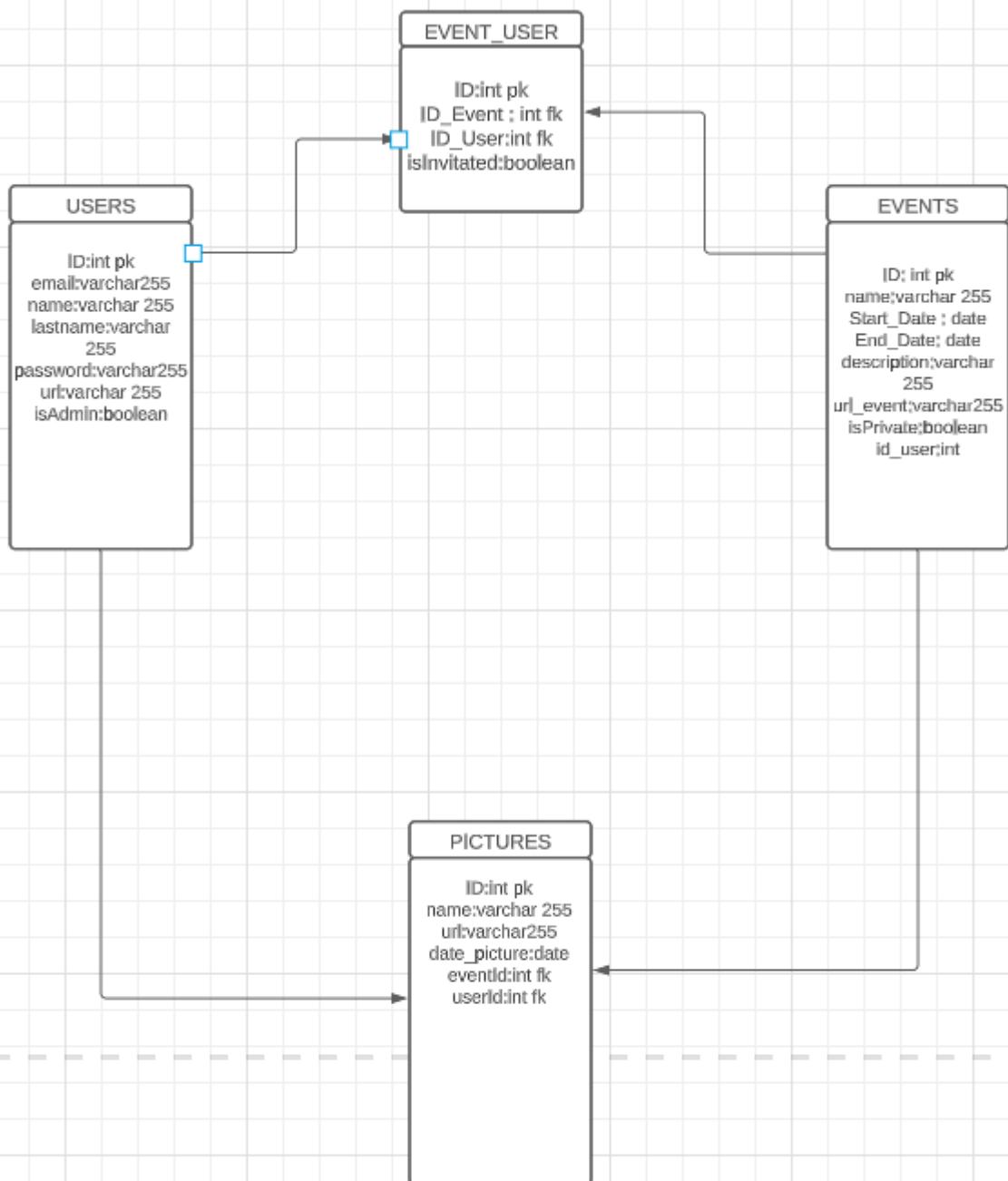
Dans un MCD, nous avons identifié les différentes entités de notre projet (par exemple, Utilisateur, Événement, etc.), ainsi que les relations entre ces entités.

DOSSIER PROFESSIONNEL (DP)



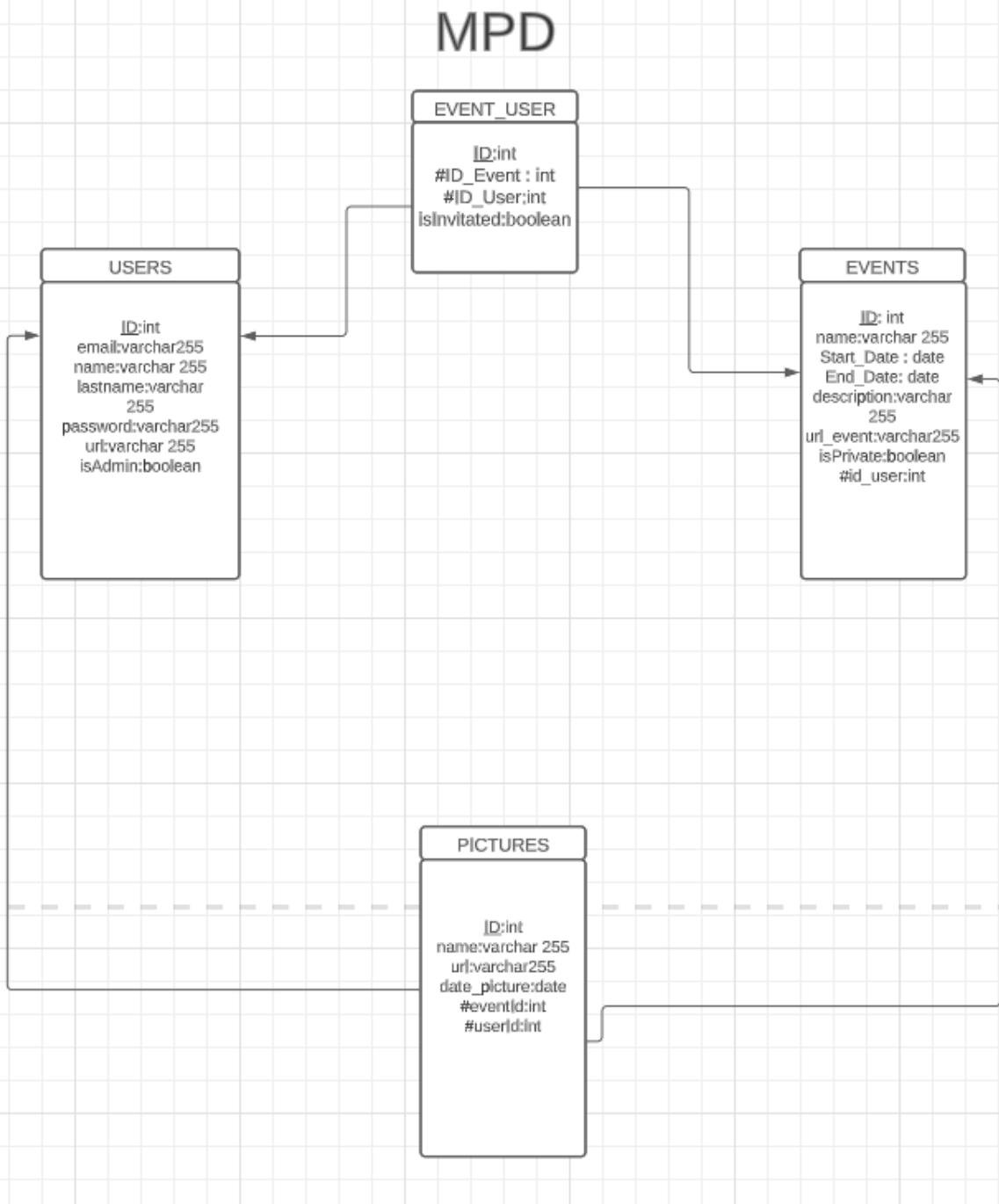
DOSSIER PROFESSIONNEL (DP)

MLD



DOSSIER PROFESSIONNEL (DP)

MPD



2. Précisez les moyens utilisés :

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

J'ai travaillé en groupe pour ce projet :

Raphael Diop

Damien Verschaere

4. Contexte

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

Chantier, atelier, service ▶ *Cliquez ici pour taper du texte.*

Période d'exercice ▶ Du : *06/09/2023* au : *31/08/2023*

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Activité-type 2

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n°1 - Mettre en place une base de données : base de données Share Event

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

Suite à la création du MCD MLD MPD nous avons créer la base de donnée a l'aide de PHPmyadmin , le MPD et la representation finale de notre base de donnée donc nous appuyons sur celui- ci pour créer les tables et leurs attributs .

dans la capture ci-dessous nous pouvons voir la table user et ses attributs :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id	int			Non	Aucun(e)		AUTO_INCREMENT	
2	email	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			
3	name	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			
4	lastname	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			
5	password	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			
6	url	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			
7	isAdmin	tinyint			Non	Aucun(e)			

puis la table Event :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id	int			Non	Aucun(e)		AUTO_INCREMENT	
2	name	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			
3	start_date	datetime(6)			Non	CURRENT_TIMESTAMP(6)		DEFAULT_GENERATED	
4	end_date	datetime(6)			Non	CURRENT_TIMESTAMP(6)		DEFAULT_GENERATED	
5	description	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			
6	url_event	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			
7	isPrivate	tinyint			Non	Aucun(e)			
8	id_user	int			Non	Aucun(e)			

DOSSIER PROFESSIONNEL (DP)

et comme vu lors de la présentation de la conception d'une bdd une table de relation entre la table user et event

table user_event :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 eventId	int			Oui	NULL			Modifier Supprimer Plus
<input type="checkbox"/>	2 id	int			Non	Aucun(e)	AUTO_INCREMENT		Modifier Supprimer Plus
<input type="checkbox"/>	3 isInvitated	tinyint			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	4 userId	int			Oui	NULL			Modifier Supprimer Plus

2. Précisez les moyens utilisés :

Pour la réalisation de ce projet nous avons utilisé **Phpmyadmin**

PhpMyAdmin fournit une interface web conviviale qui facilite la gestion des bases de données MySQL, même pour les utilisateurs non techniques. Il offre une navigation intuitive, des fonctionnalités d'édition de données, de création de tables, de requêtes SQL et d'administration des priviléges.

Gestion complète des bases de données : PhpMyAdmin permet de gérer tous les aspects d'une base de données, y compris la création, la modification et la suppression des bases de données, des tables, des colonnes, des index, des clés étrangères, etc. Il offre également des fonctionnalités avancées telles que l'importation et l'exportation de données, la sauvegarde et la restauration de bases de données.

Pour la réalisation de ce projet nous avons utilisé **Phpmyadmin**

PhpMyAdmin fournit une interface web conviviale qui facilite la gestion des bases de données MySQL, même pour les utilisateurs non techniques. Il offre une navigation intuitive, des fonctionnalités d'édition de données, de création de tables, de requêtes SQL et d'administration des priviléges.

Gestion complète des bases de données : PhpMyAdmin permet de gérer tous les aspects d'une base de données, y compris la création, la modification et la suppression des bases de données, des tables, des colonnes, des index, des clés étrangères, etc. Il offre également des fonctionnalités avancées telles que l'importation et l'exportation de données, la sauvegarde et la restauration de bases de données.

3. Avec qui avez-vous travaillé ?

Raphael diop, Damien Verschaere

DOSSIER PROFESSIONNEL (DP)

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *La plateforme.*

Chantier, atelier, service ▶ *Cliquez ici pour taper du texte.*

Période d'exercice ▶ Du : *06/09/2023* au : *31/08/2023*

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Activité-type 2

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

- Exemple n°1* - Développer des composants dans le langage d'une base de données : nest js type orm

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

Dans le cadre de notre travail sur une API de gestion d'événements, j'ai pu démontrer ma compétence à développer des composants métier en utilisant **NestJS** et **TypeORM**.

Injection de dépendances

Nous avons utilisé l'injection de dépendances pour intégrer les référentiels nécessaires à notre service. Par exemple, dans `EventsService`, nous avons injecté les référentiels pour `Event`, `User`, et `UserEvent`.

```
constructor(
  @InjectRepository(Event)
  private eventsRepository: Repository<Event>,
  @InjectRepository(User)
  private usersRepository: Repository<User>,
  @InjectRepository(UserEvent)
  private userEventRepository: Repository<UserEvent>
) {}
```

Gestion des opérations CRUD

Notre API gère un ensemble de scénarios de base de données. Par exemple, elle peut créer de nouveaux événements et utilisateurs (`create`), mettre à jour les informations des événements (`updateEvent`), récupérer la liste des événements (`findAll`), rechercher un événement spécifique (`findOne`) et supprimer des événements (`remove`).

```
async create(event: CreateEventDto) {
  const reqsUser=await this.usersRepository.findOneBy({id:event.id_user});
  const newUser =new User();
  newUser.id=reqsUser.id;
  this.usersRepository.save(newUser);

  const newEvent = new Event();
  newEvent.name = event.name,
  newEvent.url_event = event.url_event,
  newEvent.id_user = event.id_user,
  newEvent.start_date = new Date(),
  newEvent.end_date = new Date(),
  newEvent.isPrivate = event.isPrivate,
  newEvent.description = event.description,
  await this.eventsRepository.manager.save(newEvent);
}
```

Gestion des relations entre les tables

DOSSIER PROFESSIONNEL (DP)

Notre API gère les relations entre les utilisateurs et les événements. Par exemple, elle peut ajouter des utilisateurs à un événement (updateEventUsers), supprimer des utilisateurs d'un événement (deleteEventUsers), et rechercher tous les utilisateurs participant à un événement (findAllUserInEvent).

```
async updateEventUsers(eventId: string, User: GetUserDto): Promise<any> {
  const user = await this.usersRepository.findOneBy({email:User.email});
  if (user) {
    const event = await this.eventsRepository.findOneBy({ id:+eventId});
    if (event) {
      const userEvent = new UserEvent();
      userEvent.user = user;
      userEvent.event = event;
      userEvent.isInvitated=false;
```

Utilisation de requêtes complexes

Nous utilisons des requêtes SQL complexes pour récupérer des informations spécifiques. Par exemple, pour obtenir la liste de tous les utilisateurs participant à un événement, j'utilise une jointure entre les tables user_event, event, et user :

```
return await this.usersRepository
  .createQueryBuilder()
  .select("user.name,user.email,user.url,user.id")
  .innerJoin("user_event","user_event","user_event.userId = user.id")
  .innerJoin("event","event","user_event.eventId = event.id")
  .where("user_event.eventId = "+ id_event)
  .execute()
```

En conclusion, le travail que j'ai accompli sur cette API démontre **ma capacité à développer des composants métier efficaces et robustes.**

2. Précisez les moyens utilisés :

typescript , nest js

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

Raphael diop, Damien Verschaere

4. Contexte

Nom de l'entreprise, organisme ou association ➔ *La plateforme.*

Chantier, atelier, service ➔ *Cliquez ici pour taper du texte.*

Période d'exercice ➔ Du : *06/09/2023* au : *31/08/2023*

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Activité-type 3 Cliquez ici pour entrer l'intitulé de l'activité

Exemple n° 1 - Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement

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

Au cours de notre projet, nous avons adopté une approche collaborative pour gérer et organiser nos tâches et notre environnement de développement. Voici une liste de quelques outils que nous avons utilisés et comment ils ont contribué à notre processus de développement :

Google Docs : Nous avons utilisé Google Docs pour centraliser toutes les informations essentielles et les comptes-rendus de nos réunions. Cela nous a aidé à garder une trace des décisions importantes, des tâches attribuées et des prochaines étapes du projet. Cela nous a également permis de collaborer en temps réel sur les documents et de maintenir une source unique de vérité pour notre projet.

REUNION 09/01/2023 (Scrum master: Raphaël)

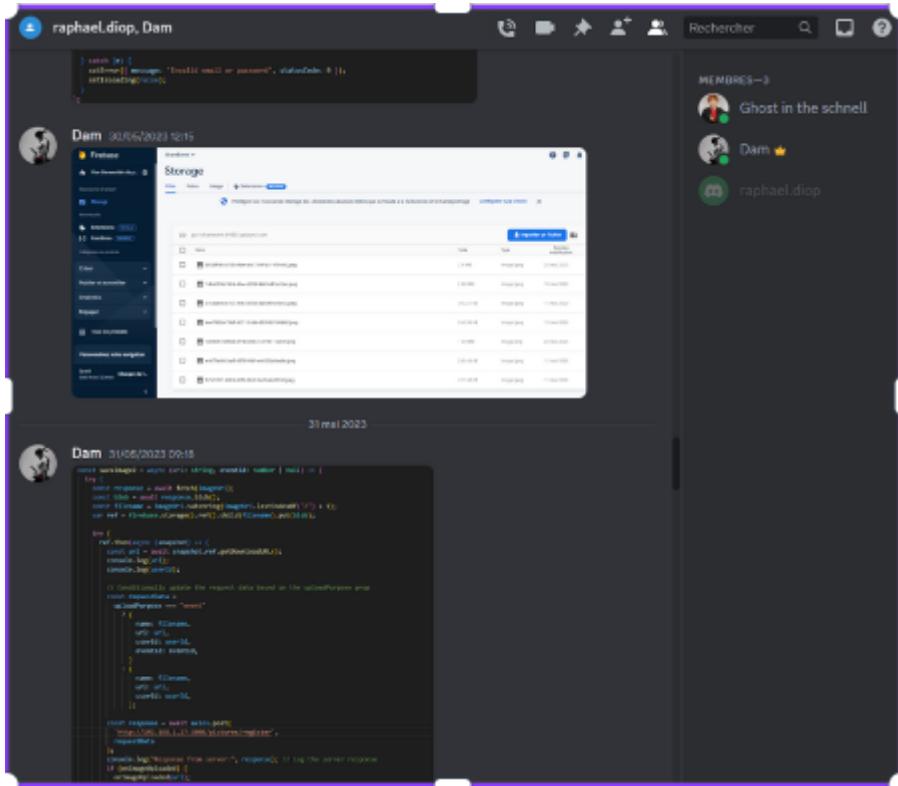
- Fix des pb de formats de date
- Poursuite de L'API :
 - Finition de routes (admin essentiellement)
 - Création de la vérification par JWT
 - Création des entités de liaison ?
- Finir de design le chemin utilisateur sur l'application

RÉUNION 10/01/2023 (Scrum master: Raphaël)

- Modification API, utilisation DTO.
- Debut dev application:
 - implémentation de la navigation dans l'appli via la librairie react-navigation (à installer depuis expo). [Getting started | React Navigation](#)
 - Dvl Écran SInIn.

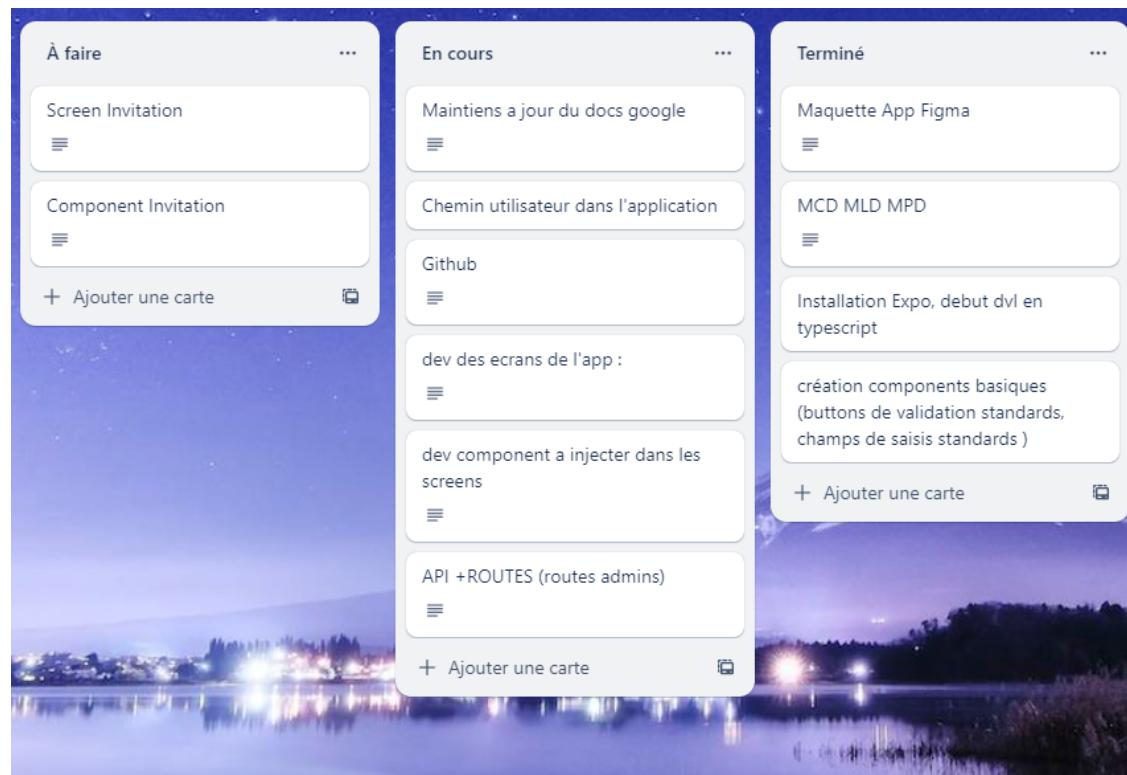
DOSSIER PROFESSIONNEL (DP)

Discord : Discord a été notre plateforme de choix pour les réunions à distance. En plus des discussions vocales et vidéo, nous avons utilisé les canaux de texte pour des mises à jour rapides, des clarifications et des discussions sur les problèmes ou tâches en cours. Discord a aidé à maintenir une communication fluide entre les membres de l'équipe malgré le travail à distance.



Trello : Nous avons utilisé Trello pour la gestion des tâches. Nous avons organisé nos tâches en listes et en cartes, en utilisant le système de statut "À faire", "En cours" et "Terminé" pour suivre l'avancement de chaque tâche. Trello nous a aidé à visualiser l'état d'avancement du projet et à identifier rapidement les goulets d'étranglement ou les problèmes potentiels.

DOSSIER PROFESSIONNEL (DP)



GitHub : Pour le contrôle de version, nous avons utilisé GitHub. Cela nous a permis de travailler simultanément sur différents aspects du projet sans interférer les uns avec les autres. En utilisant ces outils, nous avons pu travailler efficacement en tant qu'équipe, malgré les défis de la collaboration à distance. Nous avons appris à nous adapter à différents outils et processus, et à en tirer le meilleur parti pour notre projet.

ce projet m'a permis d'acquérir **Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement**

2. Précisez les moyens utilisés :

trello , discord

3. Avec qui avez-vous travaillé ?

DOSSIER PROFESSIONNEL (DP)

4. Contexte

Nom de l'entreprise, organisme ou association ➤ *Cliquez ici pour taper du texte.*

Chantier, atelier, service ➤ *Cliquez ici pour taper du texte.*

Période d'exercice ➤ Du : *Cliquez ici* au : *Cliquez ici*

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Activité-type **3** Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n°1 - Concevoir une application: ShareEvent

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

Au cours de notre projet Share Event, nous avons travaillé sur le développement de la partie front-end de l'interface utilisateur en utilisant React Native et TypeScript. Notre but était de fournir une interface utilisateur claire et intuitive pour nos utilisateurs.

Pour commencer, nous avons utilisé le système de navigation de React Navigation pour mettre en place la navigation entre les différents écrans de notre application. Nous avons principalement utilisé deux types de navigation : le Stack Navigator et le Bottom Tab Navigator.

```
const Tab = createBottomTabNavigator();
const Stack = createStackNavigator();

import { createStackNavigator } from 'react-navigation';
import { createBottomTabNavigator } from 'react-navigation'
```

Le Stack Navigator nous a permis de gérer une pile d'écrans où chaque nouvel écran est placé sur le dessus de la pile. Par exemple, nous l'avons utilisé pour la navigation entre l'écran d'accueil, l'écran de l'événement et l'écran de téléchargement :

createStackNavigator nous a permis de mettre en place une navigation entre les différents écrans de l'application. Cette bibliothèque a rendu possible la création de piles de navigation, où l'utilisateur peut naviguer en avant vers un nouvel écran, puis en arrière vers l'écran précédent.

```
function ProfilStack() {
  return (
    <Stack.Navigator initialRouteName="Profile">
      <Stack.Screen name="Profile" component={ProfilScreen} options={{ headerShown: false }} />
      <Stack.Screen name="MyEvents" component={MyEventsScreen} />
      <Stack.Screen name="MyProfil" component={UpdateProfilScreen} />
      <Stack.Screen name='ManageEvent' component={AdminEventScreen} />
    </Stack.Navigator>
  );
}
```

D'autre part, le Bottom Tab Navigator a été utilisé pour faciliter la navigation entre les différents écrans principaux de notre application. Voici un exemple de comment nous avons mis en place le Bottom Tab Navigator :

DOSSIER PROFESSIONNEL (DP)

```
function TabNavigator() {
  return (
    <Tab.Navigator
      initialRouteName="Home"
      screenOptions={({ route }) => ({
        headerShown: false,
        tabBarIcon: ({ focused, color, size }) => {
          let iconName;
          if (route.name === 'Home') {
            iconName = (
              <Entypo name="home" size={24} color={focused ? '#CBA85F' : 'white'} />
            );
          } else if (route.name === 'Profil') {
            iconName = (
              <AntDesign name="user" size={24} color={focused ? '#CBA85F' : 'white'} />
            );
          } else if (route.name === 'PublicEvents') {
            iconName = (
              <Entypo name="list" size={24} color={focused ? '#CBA85F' : 'white'} />
            );
          } else if (route.name === 'CreateEvent') {
            iconName = (
              <Ionicons name="create" size={24} color={focused ? '#CBA85F' : 'white'} />
            );
          }
          return iconName;
        },
        tabBarActiveTintColor: '#CBA85F',
        tabBarInactiveTintColor: 'white',
        tabBarStyle: {
          backgroundColor: '#98A68F',
        },
      })}
    >
      <Tab.Screen name="Home" component={EventsStack} />
      <Tab.Screen name="Profil" component={ProfilStack} />
      <Tab.Screen name="PublicEvents" component={PublicEventsScreen} />
      /* <Tab.Screen name="CreateEvent" component={CreateEventScreen} /> */
    </Tab.Navigator>
  );
}
```

Nous avons également personnalisé les icônes de la barre de navigation en utilisant des bibliothèques d'icônes externes, à savoir AntDesign, Entypo, et Ionicons. Chaque onglet de la barre de navigation a été associé à une icône spécifique, rendant l'interface utilisateur plus attrayante et intuitive.

Dans l'ensemble, notre travail sur le développement de l'interface utilisateur a impliqué l'utilisation de nombreux aspects de React Native et de TypeScript, nous permettant de fournir une application avec une expérience utilisateur de qualité.

Nous avons également accordé une attention particulière à la gestion des données utilisateur, en particulier en ce qui concerne l'authentification des utilisateurs.

Pour ce faire, nous avons utilisé **SecureStore**, une fonctionnalité d'Expo qui fournit un moyen sécurisé de stocker des données sensibles localement sur l'appareil de l'utilisateur. En utilisant **SecureStore**, nous avons pu stocker le jeton d'accès d'un utilisateur après son authentification réussie. Ce jeton est ensuite utilisé pour valider les requêtes subséquentes de l'utilisateur, permettant ainsi une expérience utilisateur fluide tout en maintenant la sécurité.

Voici un exemple de comment nous avons utilisé **SecureStore** pour stocker le jeton d'accès :

DOSSIER PROFESSIONNEL (DP)

```
const [isLoggedIn, setIsLoggedIn] = useState(false);

useEffect(() => {

  const checkIfLoggedIn = async () => {
    const token = await SecureStore.getItemAsync('acess_token');

    if (token) {
      setIsLoggedIn(true);
      console.log(token);
    } else {
      setIsLoggedIn(false);
    }
  };
  checkIfLoggedIn();
}, []);

return (

```

Dans notre projet, nous avons utilisé les **hooks** de React, qui sont des fonctions qui utilisent des callback dans la bibliothèque React, pour gérer l'état de l'application. Nous avons fait des **useState** et **useEffect**. Dans cet exemple, **isLoggedIn** est une variable d'état que nous avons initialisée à **false**. **setIsLoggedIn** est la fonction que nous avons utilisée pour modifier l'état de **isLoggedIn**.

Nous avons également utilisé le **hook useEffect** pour vérifier si un utilisateur est connecté ou non. Cela a été réalisé en vérifiant la présence d'un **token stocké localement**. Si un **token** est présent, cela signifie que l'utilisateur est connecté, et par conséquent, nous avons utilisé **setIsLoggedIn** pour mettre à jour l'état **isLoggedIn** à **true**. Sinon, nous avons réinitialisé l'état **isLoggedIn** à **false**.

La gestion de **l'état** est cruciale dans une application React car elle permet à l'interface utilisateur de réagir dynamiquement aux changements d'état. Grâce à l'utilisation efficace des **hooks** de React, nous avons pu créer une interface utilisateur dynamique et réactive qui améliore considérablement l'expérience utilisateur.

Ce projet m'a permis de valider la compétence :

Développer la partie front-end d'une interface utilisateur web

2. Précisez les moyens utilisés :

typescript , nest js

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

Raphael diop, Damien Verschaere

4. Contexte

Nom de l'entreprise, organisme ou association ➤ *La plateforme.*

Chantier, atelier, service ➤ *Cliquez ici pour taper du texte.*

Période d'exercice ➤ Du : *06/09/2023* au : *31/08/2023*

5. Informations complémentaires (*facultatif*)

DOSSIER PROFESSIONNEL (DP)

Activité-type 3

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n°1 - Construire une application organisée en couches

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

Nous avons utilisé un modèle de conception couramment utilisé: **l'architecture en couches (ou architecture n-tiers)**. Dans ce modèle, une application est divisée en plusieurs couches qui ont chacune des responsabilités distinctes.

Cette **séparation des responsabilités** permet une plus grande **modularité du code**, facilitant ainsi la **maintenance et l'évolution de l'application**.

Elle permet également une meilleure réutilisabilité du code et peut améliorer la performance en distribuant les tâches entre différentes couches.

La couche de présentation de notre application est entièrement construite à l'aide de **React Native** et **Expo**, deux technologies permettant de développer des applications mobiles multiplateformes (iOS et Android).

Dans cette section, nous allons examiner deux composants clés de notre application : le EventScreen et le EventImage.

1. Le composant EventScreen :

EventScreen est une page qui présente des informations sur un événement spécifique. Il reçoit l'identifiant (id) et le nom (nameevent) de l'événement à travers les paramètres de navigation (route). Voici un extrait de code de ce composant :

```
const EventScreen = ({ route }: EventScreenProps) => {
  const { id, nameevent } = route.params;

  return (
    <View style={styles.view}>
      <View style={styles.header}>
        <Text style={styles.headerText}>{nameevent}</Text>
      </View>
      <EventImage eventId={id} />
    </View>
  );
};
```

DOSSIER PROFESSIONNEL (DP)

Dans ce composant, nous utilisons le paramètre id pour rendre le composant EventImage qui affiche toutes les images liées à l'événement en question. Le nom de l'événement est affiché dans le header.

2. Le composant EventImage :

EventImage est un composant responsable de la récupération et de l'affichage des images associées à un événement spécifique. Il reçoit l'identifiant de l'événement et effectue une requête à l'API pour obtenir les images correspondantes.

De plus, il propose une option d'affichage différente des images grâce à un bouton de switch, et un bouton pour naviguer vers une autre page d'upload d'images.

```
const EventImage = ({ eventId }: EventImageProps) => {
  const [events, setEvents] = useState<any>([]);
  const [isEnabled, setIsEnabled] = useState(false);
  const toggleSwitch = () => setIsEnabled(previousState => !previousState);
  const navigation = useNavigation();

  useEffect(() => {
    async function getEvents() {
      try {
        const response = await axios.get(
          route + `/pictures/find/` + eventId
        );
        setEvents(response.data);
      } catch (error) {
        console.error(error);
      }
    }
    getEvents();
  }, [eventId]);
```

Ces deux composants montrent comment la couche de présentation est conçue pour interagir avec l'utilisateur et afficher les données fournies par la couche de logique métier. Leur seul rôle est de présenter les données pour l'utilisateur. Ils n'ont pas à se soucier de savoir comment les données sont récupérées ou où elles sont stockées, ce qui illustre la séparation des responsabilités entre les différentes couches de notre application.

DOSSIER PROFESSIONNEL (DP)

La couche logique métiers :

Elle abrite la logique opérationnelle de l'application, gère les interactions entre **la couche de présentation** et **la couche d'accès aux données**, et met en œuvre les règles métier. Dans le cadre de notre application, **cette couche est gérée par NestJS**.

Contrôleur des événements :

La classe **EventsController** est le point d'entrée de notre application pour toutes **les requêtes HTTP** liées à la gestion des événements. Elle définit les routes pour créer de nouveaux événements (register), mettre à jour des événements existants (updateEvent), supprimer des événements (remove), et bien plus encore. Cette classe s'appuie sur EventsService pour traiter les requêtes reçues.

```
@Controller('events')
export class EventsController {
  constructor(private readonly eventsService: EventsService,
```

Service des événements :

La classe **EventsService** gère la logique métier spécifique aux événements. Elle interagit avec la base de données via le repository **eventsRepository** pour effectuer des opérations telles que la création d'un nouvel événement (create), la mise à jour d'un événement existant (updateEvent), ou la suppression d'un événement (remove). Cette classe est essentiellement une couche d'abstraction qui simplifie l'interaction avec la base de données pour le contrôleur.

```
@Injectable()
export class EventsService {
  constructor(
    @InjectRepository(Event)
    private eventsRepository: Repository<Event>,
    @InjectRepository(User)
    private usersRepository: Repository<User>,
    @InjectRepository(UserEvent)
    private userEventRepository: Repository<UserEvent>
  ) {}
```

DOSSIER PROFESSIONNEL (DP)

Entité de l'événement :

L'entité Event représente un événement dans notre application. Elle définit la structure des événements dans notre base de données et fournit une interface pour interagir avec les données d'événements dans notre code. L'entité Event est utilisée par EventsService pour créer, mettre à jour et supprimer des événements.

```
@Entity()
export class Event extends BaseEntity {
    @PrimaryGeneratedColumn()
    id: number;
```

En structurant notre application de cette manière, nous parvenons à une séparation des responsabilités entre le contrôleur, le service et l'entité. Cela rend notre code plus lisible, plus maintenable et plus facile à tester.

De plus, en définissant la logique métier dans le service plutôt que dans le contrôleur, nous pouvons facilement réutiliser cette logique dans d'autres parties de notre application si nécessaire.

La couche d'accès au données :

La couche d'accès aux données est une composante essentielle, permettant d'interagir avec le système de stockage de données, que ce soit une base de données SQL, NoSQL, un système de fichiers, un service cloud, etc.

Dans le cas de notre application, nous utilisons une base de données SQL et interagissons avec elle via un ORM (Object-Relational Mapping), en particulier TypeORM. TypeORM est un ORM très populaire pour TypeScript et Node.js, offrant une approche basée sur les entités et les décorateurs pour définir les modèles de données et les relations entre eux.

Voici comment l'entité Event est définie :

DOSSIER PROFESSIONNEL (DP)

```
@OneToMany(() => User, (user) => user.events)
@JoinTable({
    name: 'user_event',
    joinColumn: {
        name: 'user',
        referencedColumnName: 'id',
    },
    inverseJoinColumn: {
        name: 'event',
        referencedColumnName: 'id',
    },
})
users:User[];

@OneToMany(() => Picture, (picture) => picture.event)
picture: Picture[]
}
```

Chaque entité est mappée à une table dans la base de données. Ici, Event est une table qui contient des événements. Les décorateurs comme @Entity, @PrimaryGeneratedColumn, @Column, @OneToMany, etc., sont utilisés pour décrire la structure de la table et les relations entre les tables.

En plus des entités, nous utilisons également des **Data Transfer Objects** (DTOs) pour modéliser les données transférées entre le client et le serveur. Les DTO sont essentiels pour assurer la sécurité, la validation et la cohérence des données.

Par exemple, lorsque le client crée un nouvel événement, nous n'utilisons pas directement l'objet reçu dans la requête HTTP pour créer un nouvel enregistrement dans la base de données. Au lieu de cela, nous utilisons un DTO pour définir la forme que ces données devraient avoir. Le DTO correspondant est CreateEventDto:

```
export class CreateEventDto {
    name: string;
    url_event: string;
    id_user: number;
    isPrivate: boolean;
    description: string;
}
```

Cela garantit que seul un ensemble spécifique de champs peut être utilisé pour créer un nouvel événement, évitant ainsi l'injection de champs indésirables ou potentiellement malveillants.

En conclusion, la couche d'accès aux données en utilisant un **ORM** comme **TypeORM** et des

DOSSIER PROFESSIONNEL (DP)

DTOs permet de maintenir un code propre, sécurisé et maintenable. Elle offre également une manière abstraite de travailler avec la base de données, facilitant la transition vers un autre système de base de données si nécessaire.

ce projet m'a permis d'acquérir la compétence **Construire une application organisée en couches**

2. Précisez les moyens utilisés :

typescript , nest js

3. Avec qui avez-vous travaillé ?

Raphael diop, Damien Verschaere

4. Contexte

Nom de l'entreprise, organisme ou association ➔ *La plateforme.*

Chantier, atelier, service ➔ *Cliquez ici pour taper du texte.*

Période d'exercice ➔ Du : **06/09/2023** au : **31/08/2023**

5. Informations complémentaires (facultatif)

Titres, diplômes, CQP, attestations de formation

DOSSIER PROFESSIONNEL (DP)

(facultatif)

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

DOSSIER PROFESSIONNEL (DP)

Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] Cliquez ici pour taper du texte.,

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 Cliquez ici pour taper du texte.

le Cliquez ici pour choisir une date

pour faire valoir ce que de droit.

Signature :

DOSSIER PROFESSIONNEL ^(DP)

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé

Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL ^(DP)

ANNEXES

(Si le RC le prévoit)