



MINISTÈRE CHARGÉ  
DE L'EMPLOI

# DOSSIER PROFESSION- NEL (DP)

Nom de naissance

TALEB

Nom d'usage

TALEB

Prénom

MERYEM

Adresse

05 D RUE EUGÈNE VARLIN Montreuil 93100

## Titre professionnel visé

Concepteur développeur d'applications

### MODALITE D'ACCES :

- 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.



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

# DOSSIER PROFESSIONNEL (DP)

## Sommaire

### Exemples de pratique professionnelle

<b>Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité</b>	<b>p.</b>	<b>5</b>
1    □ <b>Maquetter une application</b> : Maquettage d'un site d'une boutique en ligne.	p.	6
2    □ <b>Développer une interface utilisateur de type desktop</b> : Gestion des produits(ITECH).	p.	7, 8
3    □ <b>Développer des composants d'accès aux données</b> pour un site d'une boutique en ligne.	p.	9, 10
4    □ <b>Développer la partie front-end d'une interface utilisateur web</b> : Réalisation du site Grey's Ban.	p.	11
5    □ <b>Développer la partie back-end d'une interface utilisateur web</b> : création d'une application de gestion des étudiants.	p.	12,13
<b>Concevoir et développer la persistance des données en intégrant les recommandations de sécurité</b>	<b>p.</b>	<b>14</b>
6    □ <b>Concevoir une base de données</b> : conception : Diagramme de classe UML pour une application de gestion des articles.	p.	15,16
7    □ <b>Mettre en place une base de données</b> : application de gestion des articles en Symfony.	p.	17,18
8    □ <b>Développer des composants dans le langage d'une base de données</b> . Gérer des tables avec des requêtes SQL.	p.	19
<b>Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité</b>	<b>p.</b>	<b>20</b>
9    □ <b>Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement.</b> pour une application de livraison de colis	p.	21,22
10    □ <b>Concevoir une application.</b> Diagramme de cas d'utilisation	p.	23
11    □ <b>Développer des composants métier.</b> Diagramme de classe et diagramme de séquence	p.	24,25
12    □ <b>Construire une application organisée en couches .MVC(réaliser une application « MyBlog »)</b>	p.	26,27
13    □ <b>Développer une application mobile.</b> Le responsive design	p.	29,30
14    □ <b>Préparer et exécuter les plans de tests d'une application.</b> Les tests de sécurités.	p.	31,32
15    □ <b>Préparer et exécuter le déploiement d'une application.</b>	p.	33,34

# DOSSIER PROFESSIONNEL (DP)

<b>Titres, diplômes, CQP, attestations de formation</b> (facultatif)	p.	<b>35</b>
<b>Déclaration sur l'honneur</b>	p.	<b>36</b>
<b>Documents illustrant la pratique professionnelle</b> (facultatif)	p.	<b>37</b>
<b>Annexes</b> (Si le RC le prévoit)	p.	<b>38,70</b>

# **EXEMPLES DE PRACTIQUE PROFESSIONNELLE**

## **Activité-type 1**

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

# 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 : Maquettage d'un site d'une boutique en ligne.

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

J'ai réalisé une maquette de mon site immobilier dans le but de visualiser les fonctionnalités. Cela m'a permis de vérifier et d'ajuster les choix fonctionnels et techniques de mes pages :

- ↳ Accueil
- ↳ Inscription
- ↳ Connexion
- ↳ Liste de produits

### 2. Précisez les moyens utilisés :

Pour le maquettage j'ai utilisé l'outil Figma.

### 3. Avec qui avez-vous travaillé ?

Avec une camarade

### 4. Contexte

Nom de l'entreprise, organisme ou association La Grande Classe

□

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

Période d'exercice Du : 22/03/2023 au : 24/03/2023

### 5. Informations complémentaires (facultatif)

Voir Annexes n° 1

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 1

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

**Exemple n°2** □ Développer une interface utilisateur de type desktop : Gestion des produits(ITECH).

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

Afin de gérer les produits, nous avons développé une application à plusieurs pages. La première page permet d'ajouter un produit, la deuxième page affiche une liste des produits en prenant en compte un filtre qui nous permet d'afficher les produits en fonction de leur catégorie. La troisième page permet de créer des utilisateurs.

Les utilisateurs non connectés ont les options suivantes :

- Se connecter à l'aide de leur compte.
- S'inscrire.

Les utilisateurs connectés ont deux rôles :

#### 1. Responsable :

- Mettre à jour les utilisateurs (ajout, édition, suppression).
- Mettre à jour les produits (ajout, édition, suppression).
- Voir tous les utilisateurs inscrits.
- Voir tous les produits.
- Afficher les produits en fonction de leur catégorie.
- Se déconnecter.

#### 2. Employé :

- Voir les produits ajoutés.
- Modifier leur profil.
- Voir tous les utilisateurs inscrits.
- Voir tous les produits.
- Afficher les produits en fonction de leur catégorie.
- Se déconnecter.

-les tâches à effectuer pour les composants liés aux produits et aux utilisateurs :

Pour les produits :

1. Page de liste des produits : Afficher une liste dynamique des produits avec leurs détails.
2. Page d'ajout de produit : Permettre aux utilisateurs d'ajouter de nouveaux produits avec des informations telles que le nom, l'image, le stock et le prix.

Pour les utilisateurs :

# DOSSIER PROFESSIONNEL (DP)

1. Page d'inscription : Permettre aux utilisateurs de créer un compte en fournissant leur nom ,pseudo , email et mot de passe.
2. Page de connexion : Permettre aux utilisateurs de se connecter en saisissant leur pseudo et mot de passe.
3. Page de profil utilisateur : Afficher les informations du profil de l'utilisateur connecté, avec la possibilité de les modifier.
4. Page de gestion des utilisateurs (pour les responsables) : Permettre aux responsables de gérer les utilisateurs en modifiant ou supprimant des utilisateurs.
5. Barre de navigation.

Ces composants permettront de gérer les produits, les utilisateurs et les fonctionnalités de connexion au sein de l'application.

## 2. Précisez les moyens utilisés :

Voici les outils et technologies que j'utilise pour faciliter le développement de mon application :

-J'utilise Visual Studio Code comme éditeur de texte pour coder en PHP. Il possède une console intégrée qui me permet de lancer mes projets en mode "Go live" pour voir rapidement les modifications apportées à l'interface.  
-Pour la création des interfaces, j'utilise le Framework Bootstrap. Il offre des outils prédéfinis et adaptés aux différents appareils, ce qui facilite le développement d'interfaces réactives pour les mobiles et les tablettes.

-Pour gérer ma base de données, j'utilise SQL, un langage standardisé pour interagir avec les bases de données relationnelles.

-Du côté du backend, j'utilise PHP procédural. C'est un langage de script côté serveur qui me permet de structurer mon code de manière claire et modulaire.

## 3. Avec qui avez-vous travaillé ?

En groupe de 4 personnes.

## 4. Contexte

Nom de l'entreprise, organisme ou association  La Grande Classe

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

Période d'exercice  Du : 19/04/2023 au : 21/04/2023

## 5. Informations complémentaires ( facultatif )

Voir Annexes n° 2 , 3 et 4

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 1 Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n° 3 □ Développer des composants d'accès aux données pour un site d'une boutique.

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

Pour mon projet de boutique en ligne, voici les étapes que j'ai suivies pour développer les composants d'accès aux données :

1. J'ai installé Node.js pour créer un serveur web local.
2. J'ai utilisé le framework Express.js en exécutant la commande npm install express --save pour gérer les routes de mon application. L'option --save a enregistré cette dépendance dans le fichier package.json.
3. Dans mon fichier app.js, j'ai défini le port du serveur et importé le module Express.js. J'ai également utilisé le module body-parser pour traiter les requêtes HTTP POST.
4. J'ai adopté le modèle MVC (Modèle-Vue-Contrôleur) pour structurer mon application. Les modèles (comme User.js et Produit.js) ont été créés pour définir la structure des objets dans la base de données.
5. J'ai créé différentes vues en utilisant le moteur de template EJS. J'ai utilisé les marqueurs <% = %> pour inclure du code serveur dans mes fichiers de vue.
6. Définition des routes : Dans mon fichier app.js, j'ai défini les routes de mon application en utilisant les méthodes app.get(), app.post(), app.put(), et app.delete(). J'ai utilisé app.get() pour récupérer des données depuis la base de données ou afficher des vues, app.post() pour envoyer des données à la base de données, app.put() pour mettre à jour les données, et app.delete() pour supprimer des données. J'ai également installé et importé le module method-override pour pouvoir utiliser les méthodes PUT et DELETE.

En suivant ces étapes, j'ai pu développer les composants d'accès aux données pour mon site de boutique en ligne en utilisant Node.js, Express.js, EJS et le modèle MVC. Cela m'a permis de gérer les routes, les modèles, les vues et les opérations sur la base de données de manière simple et efficace.

### 2. Précisez les moyens utilisés :

J'ai développé les composants d'accès aux données pour mon site de boutique en ligne en utilisant les moyens suivants :

- Environnement de développement : Node.js et Visual Studio Code.
- Framework : Express.js pour la gestion des routes.
- Dépendances : Body-parser pour simplifier la récupération et le traitement des données des requêtes POST, PUT ou DELETE dans une application Node.js/Express.js.
- Moteur de template : EJS pour la création des vues.
- Base de données : MongoDB avec MongoDB Atlas.
- Mongoose pour simplifier l'interaction avec la base de données.
- J'ai utilisé le module Dotenv pour charger les variables d'environnement à partir d'un fichier .env dans mon application.
- Method-override pour utiliser les méthodes PUT et DELETE.

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seule avec l'aide de mes formateurs pour des conseils

## 4. Contexte

Nom de l'entreprise, organisme ou association  La Grande Classe, Montreuil

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

Période d'exercice  Du : 17/05/2023 au : 23/05/2023

## 5. Informations complémentaires (facultatif)

Annexe 5 et 6

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 1

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

Exemple n° 4 □ Développer la partie front-end d'une interface utilisateur web. Réalisation du site Grey's Ban.

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

Création d'un site vitrine à partir d'une maquette.

### 2. Précisez les moyens utilisés :

J'ai utilisé HTML et SCSS pour créer un site vitrine comprenant une barre de navigation, un élément "hero" et des "cards". J'ai appliqué des styles personnalisés pour améliorer l'apparence de chaque section. J'ai inclus correctement le fichier CSS généré à partir du SCSS pour appliquer les styles à mon site vitrine.

### 3. Avec qui avez-vous travaillé ?

J'ai travaillé seule.

### 4. Contexte

Nom de l'entreprise, organisme ou association      La Grande Classe, Montreuil

□

Chantier, atelier, service      □      Cliquez ici pour taper du texte.

/Période d'exercice      □□□□ Du :      17/03/2023      au :      17/06/2023

### 5. Informations complémentaires (facultatif)

Annexe 07

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 1

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

Exemple n° 5 □ Développer la partie back-end d'une interface utilisateur web. création d'une application de gestion des étudiants.

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

Nous avons développé une interface pour gérer les étudiants. Voici les fonctionnalités disponibles selon le type d'utilisateur :

#### 1. Utilisateur non connecté :

- Accès uniquement à la page de connexion (Login).
- Possibilité de se connecter en utilisant les identifiants appropriés.

#### 2. Utilisateur connecté avec le rôle Étudiant :

- Possibilité de se déconnecter.
- Accès à la liste des étudiants inscrits.

#### 3. Utilisateur connecté avec le rôle Scolaire :

- Possibilité de se déconnecter.
- Accès à la liste des étudiants inscrits.
- Possibilité d'inscrire de nouveaux étudiants.
- Possibilité de modifier les informations des étudiants inscrits.
- Possibilité de supprimer les étudiants inscrits.

Dans le cadre de ce projet, nous avons développé plusieurs composants d'interface :

- Une barre de navigation pour faciliter la navigation entre les différentes pages.
- Une page affichant la liste des étudiants inscrits, y compris leur nom, leur email et leur photo.
- Une page permettant la saisie des informations des étudiants, notamment leur nom, leur email et la possibilité d'importer leur photo.
- Une page de connexion pour accéder au système en utilisant les identifiants appropriés.
- Une page de déconnexion pour se déconnecter du système et être redirigé vers la page de connexion.

Ces composants ont été développés dans le but de faciliter la gestion des étudiants, en offrant une expérience utilisateur fluide et fonctionnelle.

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

Nous avons développé une interface de gestion des étudiants en utilisant les technologies suivantes :

- HTML : Utilisé pour la structure et la mise en page de l'interface utilisateur.
- CSS : Utilisé pour le style et la mise en forme des éléments de l'interface.
- PHP : Utilisé pour le développement du backend et la gestion des opérations côté serveur, telles que l'authentification, l'accès aux données des étudiants, l'inscription, la modification et la suppression des étudiants.
- MySQL: Utilisé pour stocker et récupérer les informations des étudiants, telles que leurs noms, e-mails et autres détails pertinents.

## 3. Avec qui avez-vous travaillé ?

Exercice effectué en cours.

## 4. Contexte

Nom de l'entreprise, organisme ou association

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

Période d'exercice  Du : 17/04/2023 au : 19/04/2023

## 5. Informations complémentaires (facultatif)

Annexe 08, 09, 10, et 11.

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

## **Activité-type 2 :**

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

# 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 : conception : Diagramme de classe UML pour une application de gestion des articles.

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

Voici les étapes que j'ai suivies pour réaliser le diagramme :

1. J'ai identifié les entités principales de l'application : "Article" et "User".
2. Pour chaque entité, j'ai identifié les attributs nécessaires. J'ai noté les types de données appropriés pour chaque attribut.
3. J'ai utilisé des conventions de notation généralement utilisées dans les diagrammes de classe, où les attributs privés sont représentés avec un tiret ("") devant leur nom.
4. L'utilisateur peut créer plusieurs articles, tandis qu'un article est associé à un seul utilisateur.
  - ↳ Du côté de l'entité "User", la cardinalité est "1", ce qui signifie qu'un utilisateur peut créer plusieurs articles.
  - ↳ Du côté de l'entité "Article", la cardinalité est "1", ce qui signifie qu'un article est associé à un seul utilisateur.

Veuillez noter que le symbole "1" représente une relation un-à-un (One-to-One) ou un-à-plusieurs (One-to-Many), tandis que le symbole "N" représente une relation un-à-plusieurs (One-to-Many) ou plusieurs-à-plusieurs (Many-to-Many). Dans ce cas, la relation "1 à N" est utilisée pour représenter la relation entre "User" et "Article".

- ↳ Les diagrammes de classe représentent les entités, les attributs et les relations d'un système.
- ↳ Chaque classe dans un diagramme de classe peut être associée à une table dans une base de données SQL.
- ↳ Les attributs de classe peuvent correspondre aux colonnes dans les tables de la base de données.
- ↳ Les relations entre les classes peuvent être représentées par des clés étrangères dans les tables.

### 2. Précisez les moyens utilisés :

Lucidchart

### 3. Avec qui avez-vous travaillé ?

seule

# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

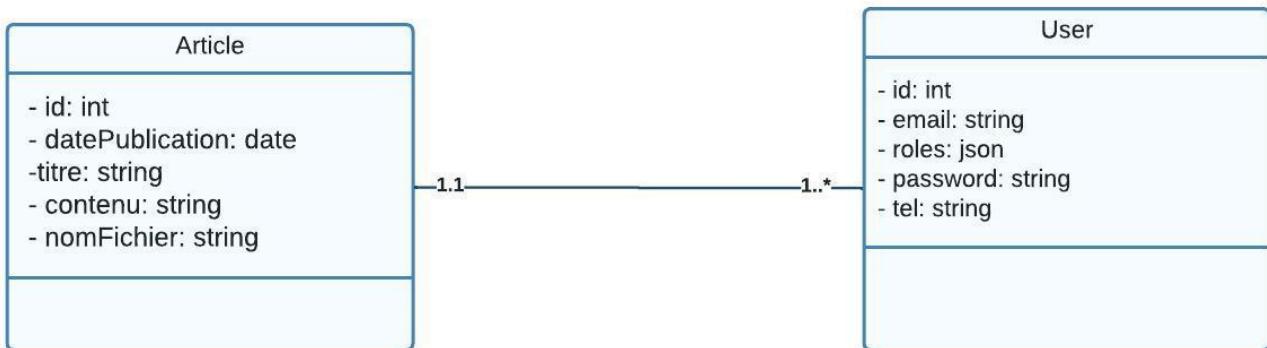
Nom de l'entreprise, organisme ou association

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

Période d'exercice  Du :  au :

## 5. Informations complémentaires (facultatif)

### Diagramme de classe



Un autre exemple de conception voir l'annexe 12.

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 2

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

Exemple n° 2 □ Mettre en place une base de données : application de gestion des articles en Symfony.

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

Pour mettre en place une base de données pour une application de gestion des articles en utilisant le framework Symfony, et en suivant les étapes suivantes :

- 1) J'ouvre le fichier `.env` à la racine de notre projet Symfony.
- 2) Je recherche les lignes qui commencent par `DATABASE_URL=`. C'est là qu'on devra spécifier les informations de connexion à la base de données.
- 3) Je modifie la valeur après `DATABASE_URL=` en fonction de votre configuration de base de données.  
`DATABASE_URL="mysql://root:@127.0.0.1:3306/gestion_articles?serverVersion=8&charset=utf8mb4"`
- 4) Créer la base de données : `symfony console doctrine:database:create`
- 5) Création des entités : Je vais créer une entité pour l'entité "Article" et une autre entité pour l'entité "User" en utilisant l'outil en ligne de commande de Symfony (console). Je vais exécuter la commande suivante pour générer les entités : `symfony console make:entity Nom_entité`
- 6) Migration de la base de données : Une fois que les entités sont définies, je vais générer le script SQL en utilisant la commande suivante : `symfony console make:migration`
- 7) Exécuter le script SQL: Je vais exécuter les migrations pour créer les tables dans la base de données en utilisant la commande suivante : `symfony console make:migrations:migrate`

### 2. Précisez les moyens utilisés :

Les moyens utilisés pour mettre en place la base de données dans Symfony :

- ↳ MySQL est le système de gestion de base de données qui stocke les données et fournit des fonctionnalités pour la gestion de la base de données à travers des requêtes SQL.
- ↳ Symfony est le framework PHP utilisé pour le développement de l'application web et fournit des fonctionnalités avancées de gestion de la base de données grâce à Doctrine, l'ORM intégré.
- ↳ phpMyAdmin est une application web qui offre une interface graphique pour la gestion de la base de données MySQL, offrant une alternative visuelle à l'utilisation de requêtes SQL pour interagir avec la base de données.

### 3. Avec qui avez-vous travaillé ?

Exercice effectué en cours

# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

Nom de l'entreprise, organisme ou association  La Grande Classe, Montreuil

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

Période d'exercice  Du : 22/06/2023 au : 23/06/2023

## 5. Informations complémentaires (facultatif)

Annexe 13.

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 2

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

Exemple n° 3 □ Développer des composants dans le langage d'une base de données.  
Crée des tables avec des requêtes SQL.

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

Pour les deux tables "article" et "user", nous avons développé des composants dans le langage SQL pour effectuer différentes opérations :

- 1) Création des tables
- 2) Insertion de données
- 3) Mise à jour de données
- 4) Suppression de données
- 5) Sélection de données

### 2. Précisez les moyens utilisés :

-le langage SQL  
-phpMyAdmin

### 3. Avec qui avez-vous travaillé ?

seule

### 4. Contexte

Nom de l'entreprise, organisme ou association La Grande Classe, Montreuil

□

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

Période d'exercice Du : 14/04/2023 au : 14/04/2023

### 5. Informations complémentaires (facultatif)

Annexe 14

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

## **Activité-type 3**

**Concevoir et développer une application multi-couche répartie en intégrant les recommandations de sécurité**

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 3

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

Exemple n° 1 □ Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement (application livraison de colis)

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

Le diagramme de Gantt pour le projet de livraison de colis :

Tâches	Durée estimée	Début	Fin
Analyse des besoins	1 semaine	29/05/2023	02/06/2023
Conception du schéma de la base de données	1 semaine	05/06/2023	09/06/2023
<b>RDV Client</b>			
Mise en place de la base de données	3 jours	13/06/2023	15/06/2023
Création des formulaires d'ajout et d'édition des colis et des annonces.	6 jours	16/06/2023	23/06/2023
Implémentation de la gestion des utilisateurs	1 semaine	26/06/2023	30/06/2023
Création des commentaires et des messages	1 semaine	03/07/2023	07/07/2023
<b>RDV Client</b>			
Intégration de l'authentification et de l'autorisation	2 jours	11/07/2023	12/07/2023
Création des vues pour l'affichage des colis et des annonces	3 jours	13/07/2023	17/07/2023
Tests et débogage	1 jour	18/07/2023	18/07/2023
<b>RDV Client</b>			
Finalisation et livraison du projet	-	20/07/2023	-

### 2. Précisez les moyens utilisés :

Gantt

### 3. Avec qui avez-vous travaillé ?

Seule

# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

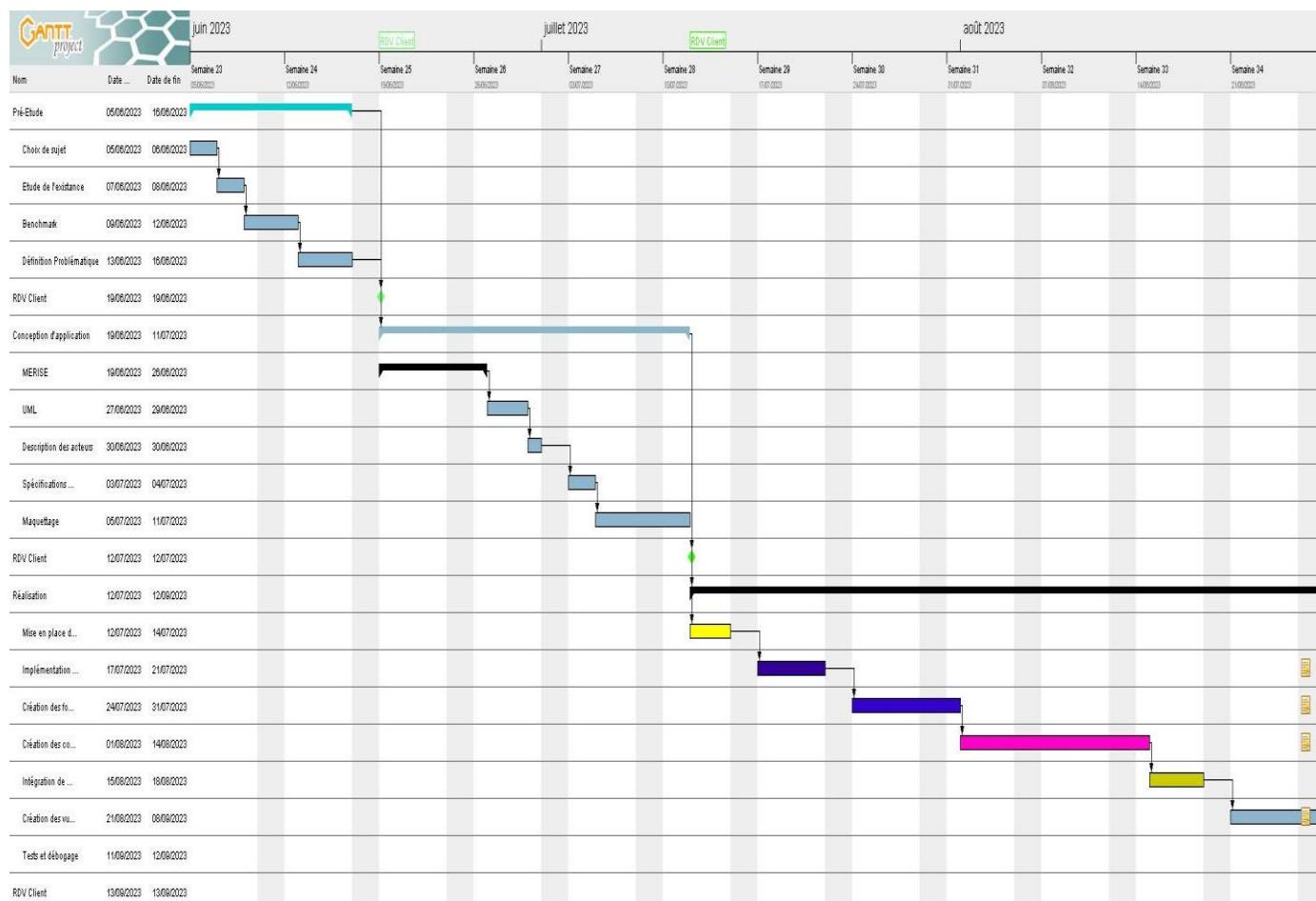
Nom de l'entreprise, organisme ou association La Grande Classe, Montreuil

□

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

Période d'exercice Du : 05/07/2023 au : 06/07/2023

## 5. Informations complémentaires (facultatif)



Voir Annexe 15.

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 3

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

**Exemple n° 2 □ Concevoir une application.** Diagramme cas d'utilisation (entreprise)

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

Nous avons élaboré le diagramme des cas d'utilisation du système suivant :

- Tous les personnels de l'entreprise peuvent consulter le système. Toute consultation doit être précédée d'une authentification légère dans laquelle la personne précise son nom et son service à des fins de statistiques ultérieures.
- Les ingénieurs peuvent effectuer différentes opérations de mise à jour pour les produits dont ils sont responsables : ajout, retrait, modification des informations sur les produits. Ces opérations doivent être précédées d'une authentification plus approfondie lors de laquelle l'ingénieur précise son nom, son service et donne un mot de passe qui est vérifié en contactant le système de gestion des personnels.
- Toutes les opérations (consultation et mise à jour) peuvent optionnellement s'accompagner d'une impression des documents accédés

### 2. Précisez les moyens utilisés :

Lucidchart

### 3. Avec qui avez-vous travaillé ?

Exercice effectué en cours.

### 4. Contexte

Nom de l'entreprise, organisme ou association **La Grande Classe, Montreuil**

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

Période d'exercice **00000** Du : **04/05/2023** au : **04/05/2023**

### 5. Informations complémentaires (facultatif)

Annexe 16

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 3

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

Exemple n° 3 □ **Développer des composants métier.** Diagramme de classe et diagramme de séquence

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

#### Diagramme de classe :

10°) Une académie souhaite gérer les cours dispensés dans plusieurs collèges. Pour cela, on dispose des renseignements suivants :

- Chaque collège possède d'un site Internet
- Chaque collège est structuré en départements, qui regroupent chacun des enseignants spécifiques. Parmi ces enseignants, l'un d'eux est responsable du département.
- Un enseignant se définit par son nom, prénom, tél, mail, date de prise de fonction et son indice.
- Chaque enseignant ne dispense qu'une seule matière.
- Les étudiants suivent quant à eux plusieurs matières et reçoivent une note pour chacune d'elle.
- Pour chaque étudiant, on veut gérer son nom, prénom, tél, mail, ainsi que son année d'entrée au collège.
- Une matière peut être enseignée par plusieurs enseignants mais a toujours lieu dans la même salle de cours (chacune ayant un nombre de places déterminé).
- On désire pouvoir calculer la moyenne par matière ainsi que par département
- On veut également calculer la moyenne générale d'un élève et pouvoir afficher les matières dans lesquelles il n'a pas été noté
- Enfin, on doit pouvoir imprimer la fiche signalétique (, prénom, tél, mail) d'un enseignant ou d'un élève.

*Elaborez le diagramme de classes correspondant. Pour simplifier l'exercice, on limitera le diagramme à une seule année d'étude*

#### Diagramme de séquence

# DOSSIER PROFESSIONNEL (DP)

## Diagramme de séquence

### Exercice – Caisse d'un supermarché

Le déroulement normal d'utilisation d'une caisse de supermarché est le suivant :

Un client arrive à la caisse avec ses articles à payer

Le caissier enregistre le numéro d'identification de chaque article, ainsi que la quantité si elle est supérieure à 1

La caisse affiche le prix de chaque article et son libellé

Lorsque tous les achats sont enregistrés, le caissier signale la fin de la vente

La caisse affiche le total des achats

Le caissier annonce au client le montant total à payer

Le client choisit son mode de paiement

- ❖ Liquide : le caissier encaisse l'argent, la caisse indique le montant à rendre au client
- ❖ Chèque : le caissier note le numéro de pièce d'identité du client
- ❖ Carte de crédit : la demande d'autorisation est envoyée avant la saisie

La caisse enregistre la vente et l'imprime

Le caissier donne le ticket de caisse au client

Modéliser cette situation à l'aide d'un diagramme de séquence en ne prenant en compte que le cas du paiement en liquide.

## 2. Précisez les moyens utilisés :

LucidChart

## 3. Avec qui avez-vous travaillé ?

Exercice effectué en cours.

## 4. Contexte

Nom de l'entreprise, organisme ou association

La Grande Classe, Montreuil

Chantier, atelier, service

Cliquez ici pour taper du texte.

Période d'exercice

0000 Du : 05/05/2023 au : 05/05/2023

## 5. Informations complémentaires (facultatif)

Annexe 17

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 3

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

**Exemple n° 4** □ **Construire une application organisée en couches .MVC (réaliser une application « MyBlog »)**

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

J'ai développé une application Myblog en utilisant l'architecture MVC avec Node.js (Express) et React ! L'architecture MVC, qui signifie Modèle-Vue-Contrôleur, est une méthode efficace pour organiser le code et séparer les différentes responsabilités, ce qui améliore la maintenabilité et l'évolutivité de l'application.

Voici une représentation simplifiée de l'architecture MVC de mon application Myblog, qui repose sur un modèle "Blog.js", un contrôleur "app.js", et trois vues : "myblog", "new post" et "edit post".

#### 1. Modèle (Model) :

- ↳ Blog.js : Dans ce fichier, j'ai défini le modèle pour les blogs. Il spécifie la structure des données des blogs dans la base de données .

#### 2. Vue (View) :

- ↳ myblog : Cette vue représente la page d'accueil de mon application Myblog. C'est là que j'affiche une liste de blogs aux utilisateurs.
- ↳ New post : Dans cette vue, j'ai créé un formulaire pour permettre aux utilisateurs de créer de nouveaux articles. Le formulaire comprend des champs pour le titre, le contenu et d'autres informations liées à chaque article.
- ↳ Edit post : Cette vue présente un formulaire d'édition qui permet aux utilisateurs de modifier un article existant.

#### 3. Contrôleur (Controller) :

- ↳ "app.js : Mon fichier "app.js" joue le rôle de contrôleur principal dans mon application Myblog. Il est responsable de la gestion des routes, de la réception des demandes HTTP et de la coordination des interactions entre les modèles et les vues. Grâce à cela, je peux traiter les actions des utilisateurs et effectuer les opérations appropriées sur les données en utilisant le modèle

Grâce à cette architecture MVC, j'ai pu développer une application bien organisée, ce qui facilite la maintenance et l'ajout de nouvelles fonctionnalités. Chaque partie de l'architecture a un rôle bien défini, ce qui rend mon code plus clair et évite les problèmes de couplage entre les différents composants de mon application Myblog.

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

Node.js avec Express  
Reactjs  
MongoDB

## 3. Avec qui avez-vous travaillé ?

Seule

## 4. Contexte

Nom de l'entreprise, organisme ou association  La Grande Classe, Montreuil

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

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

## 5. Informations complémentaires (facultatif)

Annexe 18,19 et 20

# **DOSSIER PROFESSIONNEL (DP)**

---

---

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 3

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

Exemple n° 5 □ Développer une application mobile. Le responsive design

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

Le responsive design consiste à concevoir et à développer un site web de manière à ce qu'il s'adapte automatiquement à différentes tailles d'écran, qu'il s'agisse d'un ordinateur de bureau, d'une tablette ou d'un smartphone. Voici quelques étapes et astuces pour rendre notre site web responsive :

↳ en utilisant CSS (les Media Queries) :

```
/* Styles pour les écrans larges (desktop) */
@media screen and (min-width: 1200px) {
    /* Ajoutez ici vos styles pour les écrans larges */
}

/* Styles pour les tablettes */
@media screen and (max-width: 1199px) and (min-width: 768px) {
    /* Ajoutez ici vos styles pour les tablettes */
}

/* Styles pour les smartphones */
@media screen and (max-width: 767px) {
    /* Ajoutez ici vos styles pour les smartphones */
}
```

↳ Utilisation de la grille Bootstrap :

Bootstrap propose un système de grille flexible basé sur les classes "row" et "col". Nous pouvons créer une mise en page réactive en utilisant ces classes pour diviser l'écran en colonnes.

### 2. Précisez les moyens utilisés :

Bootstrap  
Html et css

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

Exercice effectué en cours

## 4. Contexte

Nom de l'entreprise, organisme ou association

Chantier, atelier, service

Période d'exercice

## 5. Informations complémentaires (facultatif)

Annexe 21,22

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 3

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

**Exemple n° 6** □ Préparer et exécuter les plans de tests d'une application. Les tests de sécurité

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

voici un résumé des middlewares et des tests que nous avons ajoutés dans notre application Express :

1. **toobusy** : Ce middleware, basé sur le module toobusy-js, permet de détecter si l'application est trop occupée (overloaded). Si l'application est surchargée, il peut renvoyer une réponse d'erreur appropriée pour éviter des temps de réponse lents. Cela aide à gérer les périodes de forte demande et à garantir une meilleure expérience utilisateur.
2. **Captcha avec express-session** : Le middleware svgCaptcha génère un captcha sous forme de SVG pour être utilisé dans vos formulaires. Les captchas aident à prévenir les attaques par robots et assurent que les actions effectuées sur votre site sont bien réalisées par des utilisateurs légitimes.
3. **hpp** (HTTP Parameter Pollution) : Le middleware.hpp aide à prévenir la pollution des paramètres HTTP dans les requêtes. En s'assurant que les noms de paramètres sont uniques, il empêche les tentatives de contournement des contrôles de sécurité basés sur les paramètres.
4. **helmet** : helmet est un ensemble de middlewares pour sécuriser votre application Express en configurant différents en-têtes HTTP. Vous avez utilisé {crossOriginResourcePolicy: false} pour désactiver la politique Cross-Origin Resource Sharing (CORS) pour les ressources statiques. Cela permet de limiter l'accès aux ressources statiques et de protéger contre les vulnérabilités de sécurité liées à CORS.
5. **nocache** : nocache est un middleware qui configure les en-têtes HTTP pour désactiver la mise en cache des réponses. Cela garantit que les réponses envoyées par votre application ne sont pas mises en cache par les navigateurs des utilisateurs, ce qui peut parfois entraîner des problèmes de mise à jour du contenu.

↳ En combinant ces middlewares dans notre application Express, nous mettons en place des mécanismes de sécurité, de protection contre les attaques et d'optimisation des performances. Ces mesures aident à améliorer la fiabilité, la sécurité et la convivialité de notre application, en fournissant une meilleure expérience utilisateur tout en minimisant les risques liés à la sécurité.

### 2. Précisez les moyens utilisés :

**Node.js(Express).**

# DOSSIER PROFESSIONNEL (DP)

## 3. Avec qui avez-vous travaillé ?

-Exercice effectué en cours

## 4. Contexte

Nom de l'entreprise, organisme ou association

Chantier, atelier, service

Période d'exercice

## 5. Informations complémentaires (facultatif)

Annexe 23

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 3

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

Exemple n° 7 □ Préparer et exécuter le déploiement d'une application.

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

Nous avons utilisé Glitch pour développer notre application Myblog. Glitch est une plateforme de développement web qui prend en charge l'hébergement et le déploiement d'applications directement sur leur plateforme. Voici les étapes qu'on suivie pour déployer notre application Myblog sur Glitch :

1. **Créez un compte Glitch** : Si vous n'avez pas déjà de compte sur Glitch, rendez-vous sur leur site web (<https://glitch.com/>) et créez un compte gratuitement.
2. **Ouvrez votre projet Myblog sur Glitch** : Connectez-vous à votre compte Glitch et ouvrez le projet Myblog que vous avez développé. Si vous avez créé votre projet à partir de zéro, vous pouvez créer un nouveau projet en cliquant sur "New Project" et en choisissant "Import from GitHub" ou "New Project" selon votre préférence.
3. **Vérifiez et configurez les dépendances** : Assurez-vous que toutes les dépendances nécessaires pour votre application Myblog sont correctement spécifiées dans le fichier package.json. Glitch détecte automatiquement les dépendances de votre projet et les installe pour vous.
4. **Effectuez des tests** : Avant de déployer, assurez-vous que votre application Myblog fonctionne correctement sur Glitch en effectuant des tests sur leur plateforme. Vous pouvez utiliser leur console de développement pour afficher les erreurs et les logs.
5. **Déployez votre application** : Pour déployer votre application sur Glitch, il vous suffit de cliquer sur le bouton "Share" en haut à gauche de l'interface Glitch, puis de sélectionner "Live App" pour rendre votre application publique.
6. **Accédez à votre application déployée** : Une fois le déploiement terminé, Glitch vous fournira un lien direct vers votre application déployée. Vous pouvez le partager avec d'autres ou l'ouvrir dans un navigateur pour voir votre application en direct.

### 2. Précisez les moyens utilisés :

Glitch

### 3. Avec qui avez-vous travaillé ?

Exercice effectué en cours

# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

Nom de l'entreprise, organisme ou association  La Grande Classe, Montreuil

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

Période d'exercice  Du : 20/07/2023 au : 20/07/2023

## 5. Informations complémentaires (facultatif)

Annexe 24

## **DOSSIER PROFESSIONNEL (DP)**

## **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.

# 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)

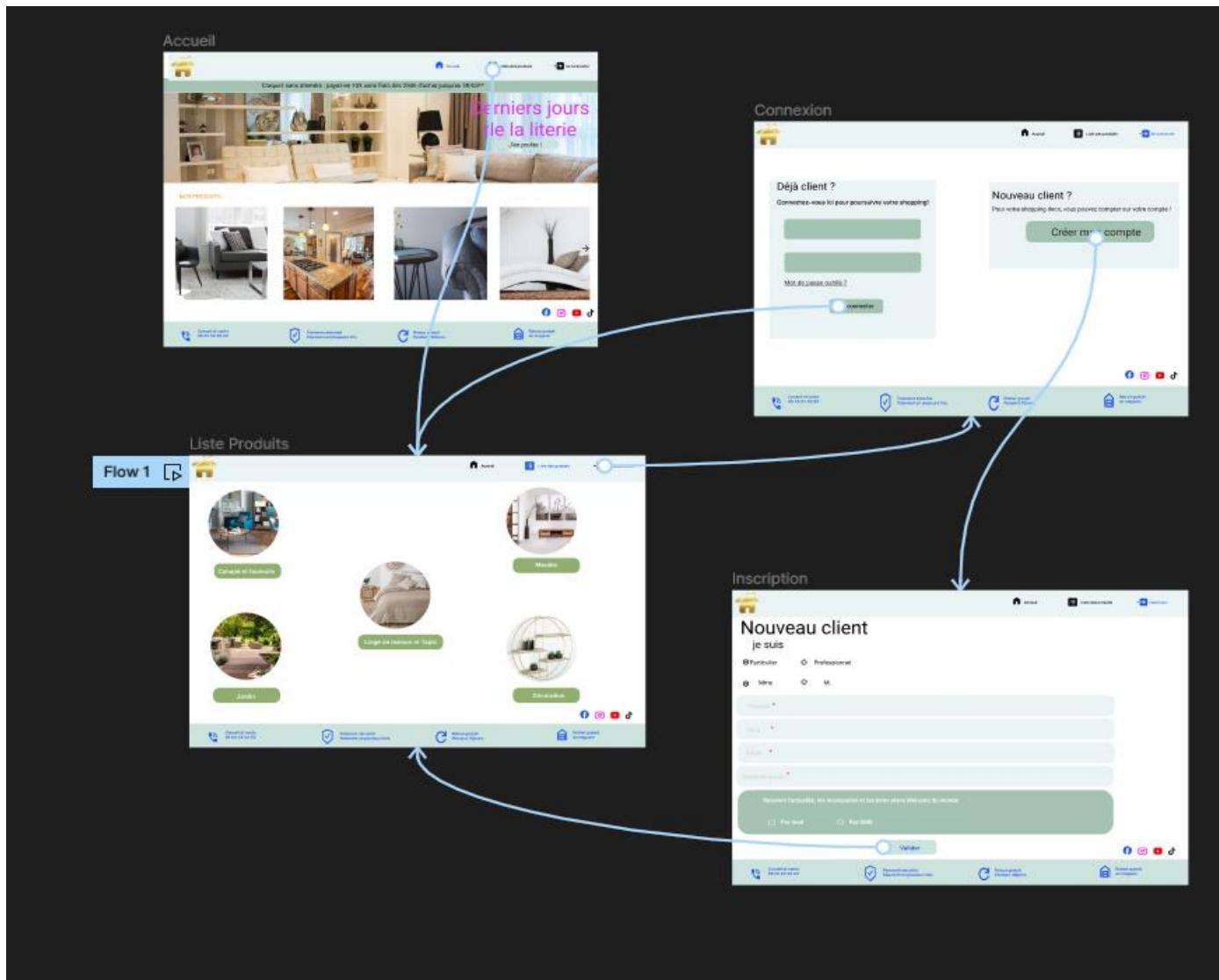
### Intitulé

Cliquez ici pour taper du texte.

# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

### Annexes 1: Maquetter une application : Maquettage d'un site d'une boutique en ligne.



### Annexes 2 : Développer une interface utilisateur de type desktop : Gestion des produits(ITECH).

Page Accueil

# DOSSIER PROFESSIONNEL (DP)



ITECH COMPANY

Catégorie ▾

Tous Les Produits

			
<b>ASUS ROG STRIX</b> Prix : 1800€ Stock : 10 Categorie : Ordinateur <a href="#">Modifier</a> <a href="#">Supprimer</a>	<b>MACBOOK AIR</b> Prix : 1200€ Stock : 25 Categorie : Ordinateur <a href="#">Modifier</a> <a href="#">Supprimer</a>	<b>DELL XPS</b> Prix : 2000€ Stock : 20 Categorie : Ordinateur <a href="#">Modifier</a> <a href="#">Supprimer</a>	<b>GALAXY TAB S</b> Prix : 900€ Stock : 20 Categorie : Tablette <a href="#">Modifier</a> <a href="#">Supprimer</a>
			
<b>IPAD PRO</b> Prix : 1100€ Stock : 30 Categorie : Tablette <a href="#">Modifier</a> <a href="#">Supprimer</a>	<b>SURFACE TAB</b> Prix : 1300€ Stock : 20 Categorie : Tablette <a href="#">Modifier</a> <a href="#">Supprimer</a>	<b>XIAOMI 13</b> Prix : 1200€ Stock : 15 Categorie : Mobile <a href="#">Modifier</a> <a href="#">Supprimer</a>	<b>SAMSUNG Z FLIP</b> Prix : 1400€ Stock : 10 Categorie : Mobile <a href="#">Modifier</a> <a href="#">Supprimer</a>

## Annexe 3 :

### Page liste des utilisateurs

# DOSSIER PROFESSIONNEL (DP)



La liste des utilisateurs

Modifier mes infos

Pseudo	Nom	Prénom	Email	Rôle	Modifier	Supprimer
willy	smith	will	will@email.com	responsable	<button>Modifier</button>	<button>Supprimer</button>
wick	wick	john	wick@gmail.com	responsable	<button>Modifier</button>	<button>Supprimer</button>
Chris	Bale	Christian	chris@gmail.com	employé	<button>Modifier</button>	<button>Supprimer</button>
leo	Dicaprio	Leonardo	leo@gmail.com	employé	<button>Modifier</button>	<button>Supprimer</button>
Morgan	Freeman	Morgan	morgan@gmail.com	employé	<button>Modifier</button>	<button>Supprimer</button>

## Page ajouter un produit



Ajouter un produit

Catégorie

Choisissez une catégorie du produit

Nom du produit

Prix

Stock restant

Photo

Choisir un fichier Aucun fichier n'a été sélectionné

Ajouter

## Annexe 4 :

### Page connexion et inscription :

# DOSSIER PROFESSIONNEL (DP)

[Se connecter](#) [S'inscrire](#)  [Rechercher](#)

## Connectez vous !

Entrez votre Pseudo

Entrez Le Mot de passe

[Se connecter](#)[Se connecter](#) [S'inscrire](#)  [Rechercher](#)

## S'inscrire

Pseudo

Nom

Prénom

Email

Mot de passe

[S'inscrire](#)

# DOSSIER PROFESSIONNEL (DP)

## Développer des composants d'accès aux données pour un site d'une boutique.

### Annexe 05 :

#### 1-Définir le port du serveur :

```
//server*****
var server = app.listen(5000, function () {
  console.log("Server listening on port 5000");
});
```

#### 2-Configuration d'Express, Dotenv, bodyParser, EJS et Method-override dans mon application :

```
//Express
var express = require('express');
var app = express();

//Dotenv
require('dotenv').config();

//bodyParser*****
var bodyParser = require('body-parser');
app.use(bodyParser.urlencoded({ extended: false }));

//EJS*****
app.set('view engine', 'ejs');

//Method-override
const methodOverride = require('method-override');
app.use(methodOverride('_method'));

//Bcrypt*****
const bcrypt = require('bcrypt');
```

### Annexe 06 :

#### 3-Models : Produit.js

# DOSSIER PROFESSIONNEL (DP)

Produit.js X

Projet\_Nodejs > models > Produit.js > ...

```
1 const mongoose = require('mongoose');
2 const produitSchema = mongoose.Schema({
3     nom : {type : 'String'},
4     categories : {type : 'String'},
5     prix : {type : 'Number'},
6     stock : {type : 'Number'},
7 }
8 module.exports = mongoose.model('Produit',produitSchema);
```

# DOSSIER PROFESSIONNEL (DP)

## 4-Ajouter PUT et DELETE

```
<form class="w-50 m-auto" action= "/produit/edit/<%= data.id %>?_method=PUT" method="post">
    <input type="hidden" name="_method" value="PUT">

    <form action="/produit/delete/<%= produit.id%>?_method=DELETE" method="post">
        <input type="hidden" name="_method" value="DELETE">
        <input class="btn btn-warning" onclick="return confirm('Vous voulez supprimer le post ?')" type="submit" value="Supprimer">
    </form>
```

## 5-Afficher tous les produits :

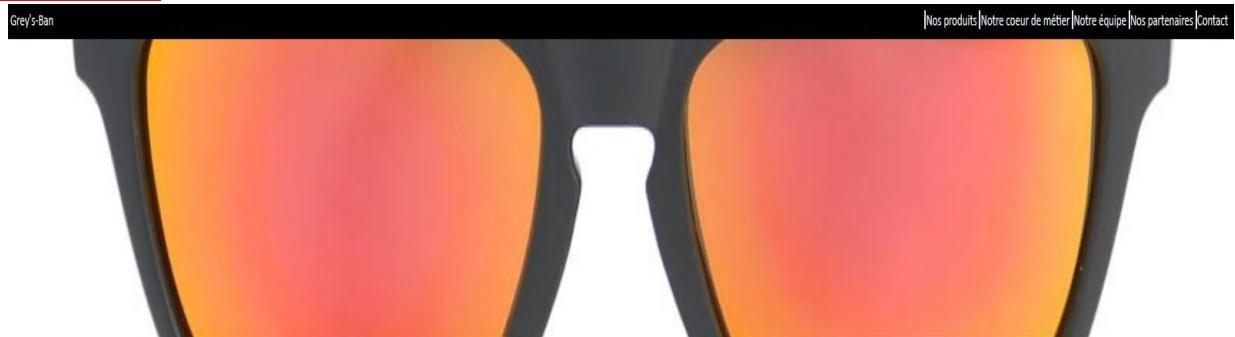
```
const Produit = require('./models/Produit');
app.get('/produits', function (req, res) {
    Produit.find().then((data) => {
        console.log(data);
        res.render('Produits',{ data : data});
    }).catch(err => console.log(err));
})

<table class="table">
    <thead>
        <tr>
            <th scope="col">Nom Produit</th>
            <th scope="col">Catégories</th>
            <th scope="col">Stock</th>
            <th scope="col">Prix</th>
            <th scope="col">Modifier</th>
            <th scope="col">Supprimer</th>
        </tr>
    </thead>
    <tbody>
        <% data.forEach((produit)=>{ %>
            <tr>
                <td>
                    | <%= produit.nom %>
                </td>
                <td>
                    | <%= produit.categories %>
                </td>
                <td>
                    | <%= produit.stock %>
                </td>
                <td>
                    | <%= produit.prix %>€
                </td>
            </tr>
        <% } %>
    </tbody>
</table>
```

# DOSSIER PROFESSIONNEL (DP)

Développer la partie front-end d'une interface utilisateur web. Réalisation du site Grey's Ban.

Annexe 07 :



Nos produits les plus populaires



©Grey's Ban 2023

Développer la partie back-end d'une interface utilisateur web. création d'une application de gestions des étudiants.

Annexe 08 :

# DOSSIER PROFESSIONNEL (DP)

[Se connecter](#) [S'inscrire](#)

Authentification

Login:

Mot de passe:

[Se connecter](#)

## Annexe 09 :

[Etudiants](#) [Saisie](#) [Se deconnecter](#)

Ajouter un étudiant

Nom:

Email:

Photo: [Parcourir...](#) Aucun fichier sélectionné.

[Sauvgarder](#)

## Annexe 10 :

# DOSSIER PROFESSIONNEL (DP)

Etudiants Saisie Se deconnecter

## La liste des étudiants

CODE	NOM	EMAIL	PHOTO	Modifier	Supprimer
18	Maxime	Max@gmail.com		<button>Modifier</button>	<button>Supprimer</button>
15	Sarah	Ben@gmail.com		<button>Modifier</button>	<button>Supprimer</button>
16	Jonathan	Jon@gmail.com		<button>Modifier</button>	<button>Supprimer</button>
19	Ily	ily@gmail.com		<button>Modifier</button>	<button>Supprimer</button>

## Annexe 11 :

### Enregistre un étudiant :

```
① ecole >  saveEtudiant.php > ...
1  <?php require_once("securite.php");?>
2  <?php require_once("rolescolarite.php");?>
3  <?php
4  // récupere les données de formulaire saisie par les utilisateurs.
5  $nom = $_POST['nom']; //
6  $email = $_POST['email'];
7  $nomPhoto = $_FILES['photo']['name'];
8  $fichierTempo = $_FILES['photo']['tmp_name'];
9  // Deplacer le fichier temporel vers le dossier img de notre application.
10 move_uploaded_file($fichierTempo, './img/' . $nomPhoto); // $nomPhoto pour recuperer le nom de l'image.
11 //envoie des données vers BD.
12 //Se connecter à la base de données.
13 require_once("connexion.php");
14 //cree objet prepare statement ($ps)
15 $ps = $pdo->prepare("INSERT INTO etudiants(NOM,EMAIL,PHOTO) VALUES(?, ?, ?)");
16 //stocker les parametre (dans l'ordre)dans un tableau
17 $params = array($nom, $email, $nomPhoto);
18 //executer la requete +transfere les parametre dans l'ordre
19 $ps->execute($params);
20 //rediriger :revenir vers l'apache pour afficher les etudiants.
21 header("location:etudiant.php");
??
```

# DOSSIER PROFESSIONNEL (DP)

## Saisie un étudiant :

```
① ecole >  saisieEtudiant.php >  html >  body
1   <?php require_once("securite.php");
2   require_once("rolescolarite.php");?>
3   <html lang="en">
4   > <head>...
14  </head>
15  <body>
16      <?php require_once("entete.php"); ?>
17      <div class="col-md-5 m-auto col-xs-5 spacer">
18          <div class="card spacer ">
19              <div class="card-header text-center fs-2">Ajouter un étudiant</div>
20              <div class="card-body m-auto">
21                  <!-- Contenu du formulaire -->
22                  <form action="saveEtudiant.php" method="post" enctype="multipart/form-data">
23                      <div class="form-group ">
24                          <label for="">Nom:</label>
25                          <input type="text" name="nom" id="form-control">
26                      </div>
27                      <div class="form-group mt-3">
28                          <label for="">Email:</label>
29                          <input type="email" name="email" id="form-control">
30                      </div>
31                      <div class="form-group mt-3">
32                          <label for="">Photo:</label>
33                          <input type="file" name="photo" id="form-control">
34                      </div>
35                      <div class="mt-3 text-center">
36                          <button class="btn btn-success " type="submit">Sauvgarder</button>
37                      </div>
38                  </form>
39          </div>
40      </div>
41  </div>
```

## Barre de navigation

# DOSSIER PROFESSIONNEL (DP)

④ ecole >  entete.php >  nav.navbar.navbar-expand-lg.bg-dark.navbar-dark.fixed-top

```
1  <nav class="navbar navbar-expand-lg bg-dark navbar-dark fixed-top">
2      <ul class="navbar-nav">
3          <li class="nav-item" <?php if (isset($_SESSION['PROFILE'])) { ?>>
4              <a href="etudiant.php" class="nav-link">Etudiants</a>
5          </li>
6          <li class="nav-item">
7              <a href="saisieEtudiant.php" class="nav-link">Saisie</a>
8          </li>
9          <li class="nav-item">
10             <a href="logOut.php" class="nav-link">Se deconnecter</a>
11         </li>
12     <?php } ?>
13     <li class="nav-item" <?php if (!isset($_SESSION['PROFILE'])) { ?>>
14         <a href="etudiant.php" class="nav-link">Se connecte</a>
15     </li>
16     <li class="nav-item">
17         <a href="saisieEtudiant.php" class="nav-link">S'inscrire</a>
18     </li><?php } ?>
19   </ul>
20
21 </nav>
```

## Annexe 12 : Concevoir une base de données : Conception.

# DOSSIER PROFESSIONNEL (DP)

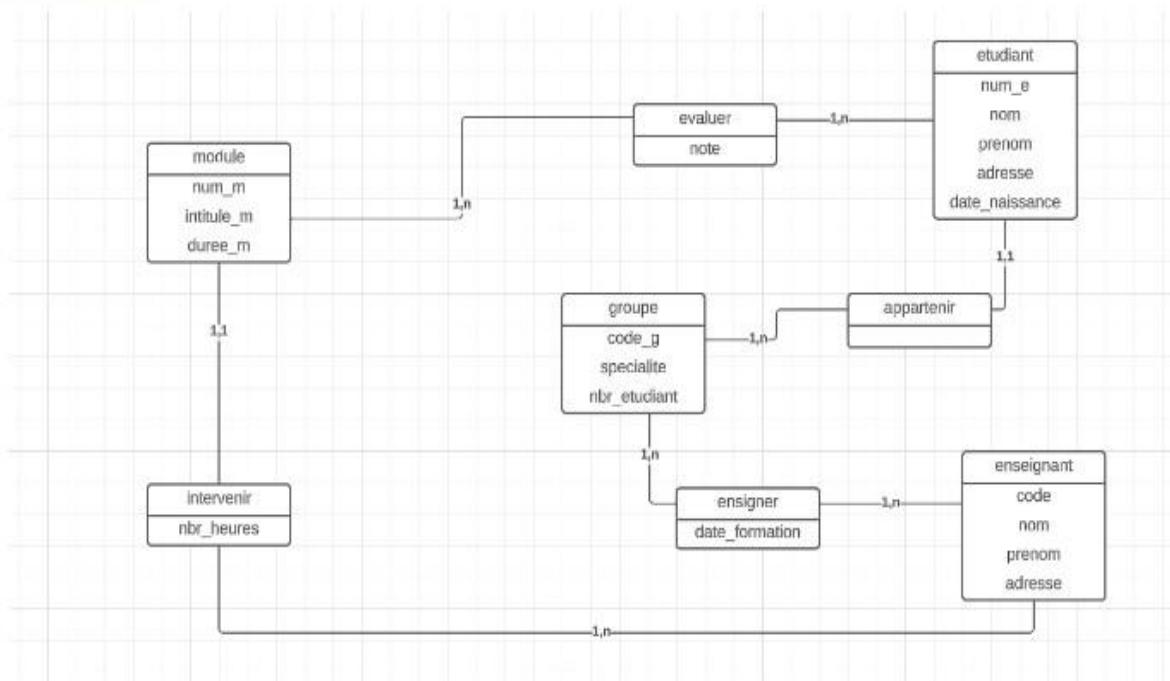
## **2-MCD et MLD pour un institut de formation**

- Les cours sont organisés en modules, chaque module est caractérisé par un numéro de module, un intitulé, une durée en
  - heures et un type.
- Les étudiants suivent des enseignements portant sur plusieurs modules. Chaque étudiant est caractérisé par un numéro
  - d'inscription unique, un nom, un prénom et une adresse et une date de naissance, un étudiant est évalué trois fois pour chaque
- module et possède une note de fin de module.
- Chaque étudiant appartient à un groupe caractérisé par un code, une spécialité et le nombre d'étudiants qu'il comporte.
- Un enseignant intervient dans un module pour un groupe donné à une date donnée, chaque enseignant est caractérisé par un
  - code, un nom, un prénom et une adresse.
- Un enseignant intervient habituellement dans plusieurs modules.
- On désire aussi mémoriser le nombre d'heures effectué par chaque enseignant dans un module donné pour un groupe donné

# DOSSIER PROFESSIONNEL (DP)

## Institut de formation

### 1-MCD



### 2-MLD

- **Evaluer**(num\_m#, num\_e#, note)
- **Etudiant**(num\_e, nom, prenom, adresse, date\_naissance, code\_g#)
- **Enseigner**(code\_g#, code#, date\_formation)
- **Groupe**(code\_g, specialite, nbr\_etudiant)
- **Enseignant**(code, nom, prenom, adresse)
- **Intervenir**(nbr\_heures)
- **Module**(num\_m, intitule\_m, duree\_m, code#)

# DOSSIER PROFESSIONNEL (DP)

## Annexe 13 : Mettre en place une base de données : application de gestion des articles en Symfony.

### ↳ Le fichier .env :

```
| DATABASE_URL="mysql://root:@127.0.0.1:3306/gestion_articles?serverVersion=8&charset=utf8mb4"
```

### ↳ Crée la base de données : **symfony console doctrine:database:create**

```
▶ PS C:\Users\merye\Downloads\gestion_articles\gestion_articles\gestion_articles> symfony console doctrine:database:create
Created database `gestion_articles` for connection named default
```

### ↳ Créeation des entités : **symfony console make:entity Nom\_entité**

```
PS C:\Users\merye\Downloads\gestion_articles\gestion_articles\gestion_articles> symfony console make:entity Articles
created: src/Entity/Articles.php
created: src/Repository/ArticlesRepository.php

Entity generated! Now let's add some fields!
You can always add more fields later manually or by re-running this command.

New property name (press <return> to stop adding fields):
> DatePublication

Field type (enter ? to see all types) [string]:
> date

Can this field be null in the database (nullable) (yes/no) [no]:
> yes

updated: src/Entity/Articles.php

Add another property? Enter the property name (or press <return> to stop adding fields):
```

### ↳ générer le script SQL: **symfony.exe console make:migration**

```
▶ PS C:\Users\merye\Downloads\gestion_articles\gestion_articles\gestion_articles> symfony.exe console make:migration
```

```
created: migrations/Version20230718120108.php
```

Success!

```
Review the new migration then run it with symfony.exe console doctrine:migrations:migrate
See https://symfony.com/doc/current/bundles/DoctrineMigrationsBundle/index.html
```

### ↳ Exécuter le script SQL: **symfony.exe console make:migrations:migrate**

# DOSSIER PROFESSIONNEL (DP)

```
PS C:\Users\merrye\Downloads\gestion_articles\gestion_articles\gestion_articles> symfony.exe console doctrine:migrations:migrate
WARNING! You are about to execute a migration in database "gestion_articles" that could result in schema changes and data loss. Are you sure you wish to continue? (yes/no) [yes]:
> yes

[notice] Migrating up to DoctrineMigrations\Version20230718120108
[notice] finished in 5662ms, used 20M memory, 1 migrations executed, 5 sql queries
[OK] Successfully migrated to version : DoctrineMigrations\Version20230718120108
```

```
PS C:\Users\merrye\Downloads\gestion_articles\gestion_articles\gestion_articles>
```

## Annexe 14 : Développer des composants dans le langage d'une base de données. Crée des tables avec des requêtes SQL.

### **1) Création des tables**

```
1 CREATE TABLE article (
2   id INT PRIMARY KEY,
3   date_publication DATE,
4   contenu TEXT,
5   nom_fichier VARCHAR(255),
6   user_id INT,
7   FOREIGN KEY (user_id) REFERENCES user(id)
8 );
9
10 CREATE TABLE users (
11   id INT PRIMARY KEY,
12   email VARCHAR(180) UNIQUE,
13   roles JSON,
14   password VARCHAR(255),
15   tel VARCHAR(13)
16 );
```

### **2) Insertion de données**

```
1 INSERT INTO article (id, date_publication, contenu, nom_fichier, user_id)
2 VALUES (1, '2023-07-18', 'Contenu de l\'article 1', 'article1.jpg', 1);
3
4 INSERT INTO user (id, email, roles, password, tel)
5 VALUES (1, 'exemple@email.com', '[{"ROLE_ADMIN"}]', 'password123', '0123456789');
```

### **3) Mise à jour de données**

# DOSSIER PROFESSIONNEL (DP)

```
1 UPDATE article SET contenu = 'Nouveau contenu de l'article 1' WHERE id = 1;
```

## 4) Suppression de données

```
1 DELETE FROM article WHERE id = 1;
```

## 5) Sélection de données

```
1 SELECT * FROM user;
```

```
1 SELECT * FROM article;
```

## Annexe 15 : Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement (application livraison de colis)

### ↳ User stories :

- En tant que Julien, je souhaite pouvoir créer un compte sur la plateforme afin d'accéder aux fonctionnalités de livraison de colis en avion.
- En tant que Maxime je souhaite pouvoir publier une offre de livraison de colis en avion avec les détails spécifiques, tels que l'aéroport de départ, l'aéroport d'arrivée, la taille du colis et la date limite de livraison, afin que des voyageurs comme Marie puissent trouver mon offre.
- En tant que Marie, je souhaite pouvoir consulter les offres de livraison de colis en avion disponibles, y compris les informations sur le colis, les aéroports de départ et d'arrivée, et les dates de voyage prévues, afin de choisir une offre qui correspond à mes préférences.
- En tant que Sara, je souhaite pouvoir sélectionner Pierre pour la livraison de mon colis en avion, en établissant un accord sur les détails de la livraison, tels que la rémunération, l'assurance et

### ↳ MOSCOW (Must have, Should have, Could have, Won't have)

#### 1. Must :

# DOSSIER PROFESSIONNEL (DP)

- Les utilisateurs doivent pouvoir créer un compte et se connecter à la plateforme.

## 2. **should**

- Les expéditeurs devraient pouvoir publier des offres de livraison de colis en avion.
- Les voyageurs devraient pouvoir rechercher et consulter les offres de livraison disponibles.

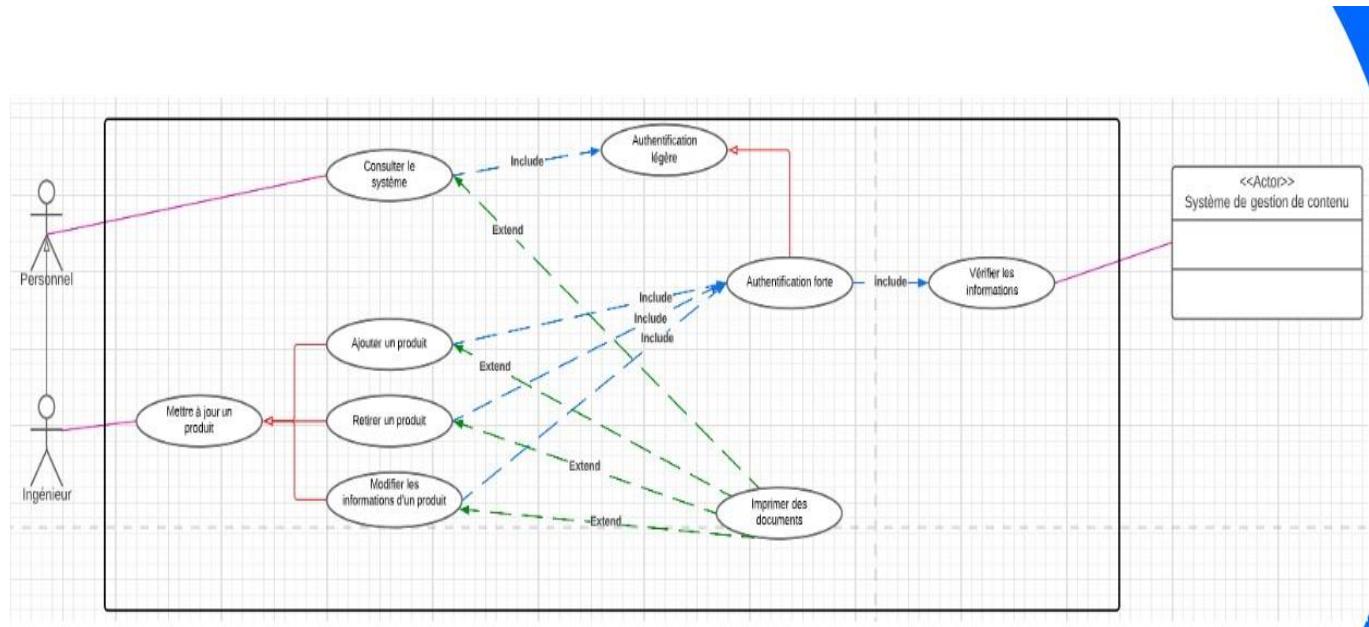
## 3. **Could :**

- Les utilisateurs pourraient recevoir des notifications lorsqu'une nouvelle offre correspondant à leurs préférences est publiée.
- Les utilisateurs pourraient pouvoir consulter les évaluations et les commentaires des voyageurs précédents.

## 4. **Won't :**

- L'application ne prendra pas en charge d'autres modes de transport que l'avion pour la livraison de colis.

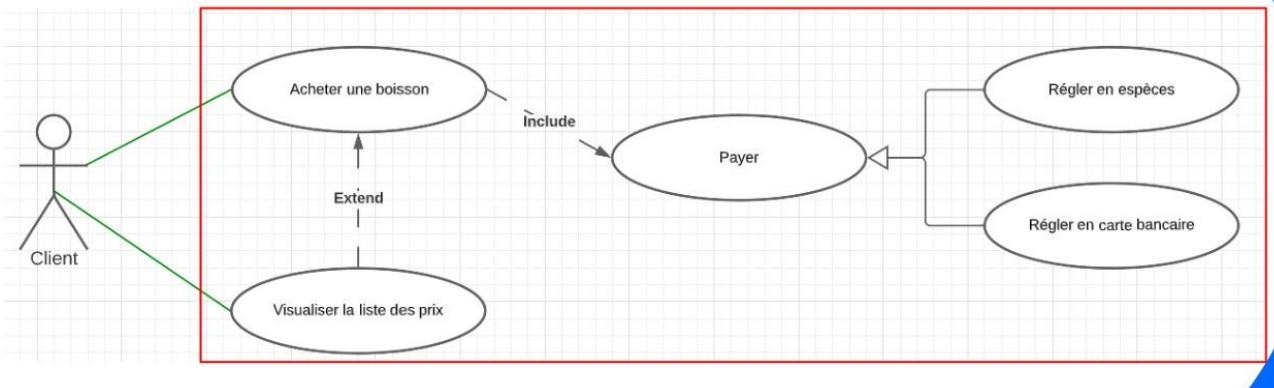
## Annexe 16 Concevoir une application. Diagramme de cas d'utilisation (entreprise)



## Exemple 2 :

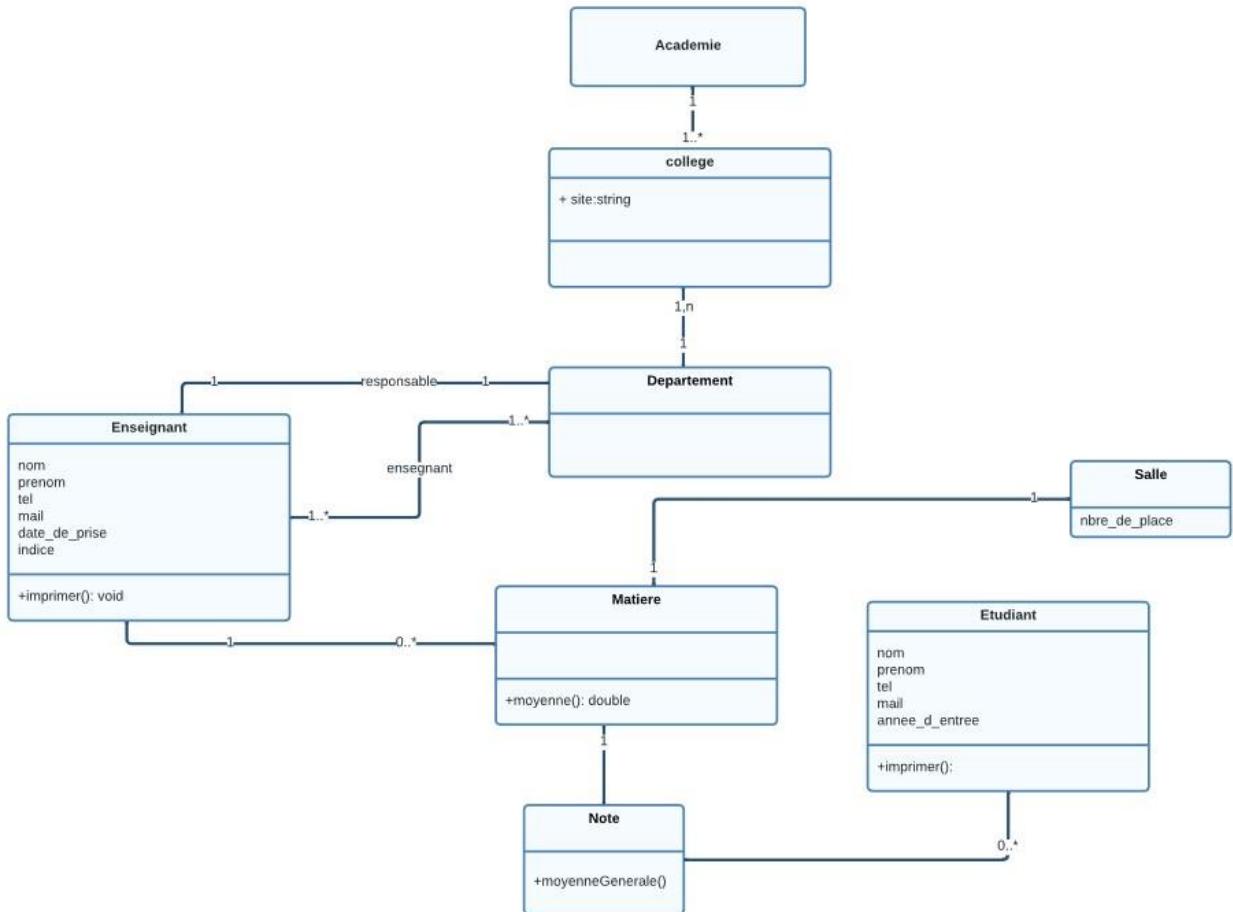
## Diagramme des cas d'utilisation : Distributeurs de boissons chaudes et machines à café

# DOSSIER PROFESSIONNEL (DP)



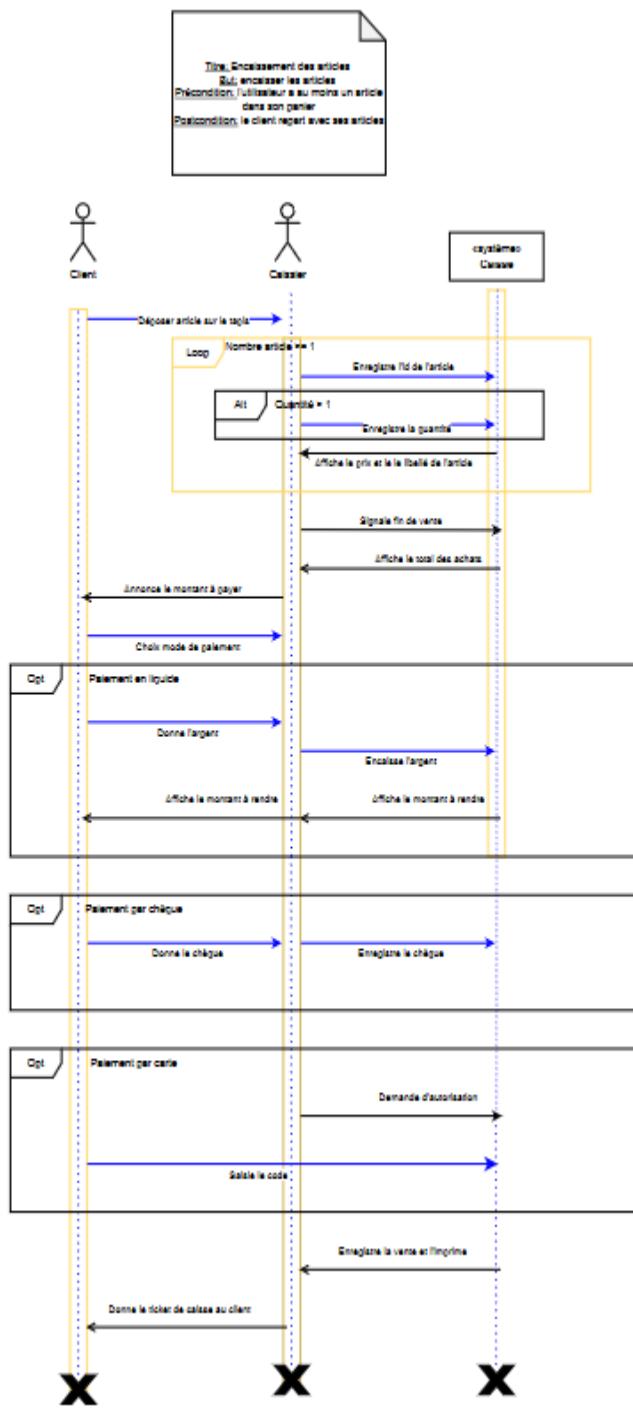
## Annexe 17 : Développer des composants métier. Diagramme de classe et diagramme de séquence.

### Diagramme de classe



# DOSSIER PROFESSIONNEL (DP)

## Diagramme de séquence



Construire une application organisée en couches .MVC (réaliser une application « MyBlog »)

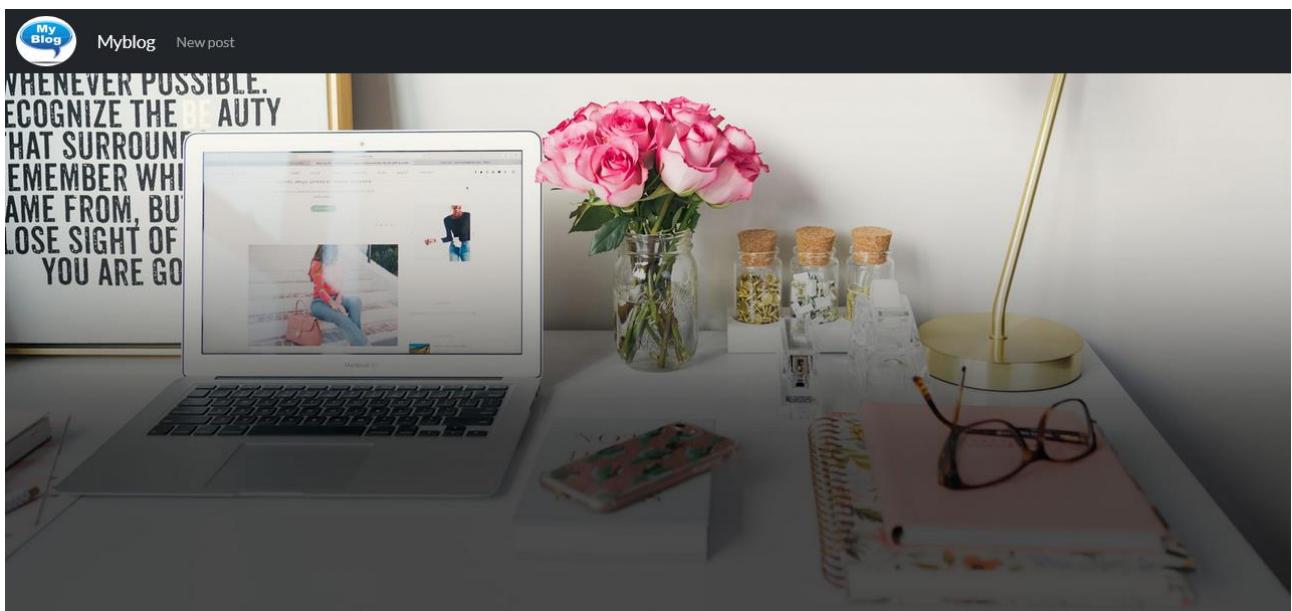
# DOSSIER PROFESSIONNEL (DP)

## Annexe 18 : Models

```
Blog.js      X
Backend > models > Blog.js > blogSchema
1  const mongoose = require('mongoose')
2
3  const blogSchema = mongoose.Schema({
4    titre: {type : 'String'},
5    username: {type : 'String'},
6    imagename: {type : 'String'},
7    content: {type : 'String'},
8
9  })
10
11
12  module.exports = mongoose.model('Blog', blogSchema);
```

## Annexe 19 :Views

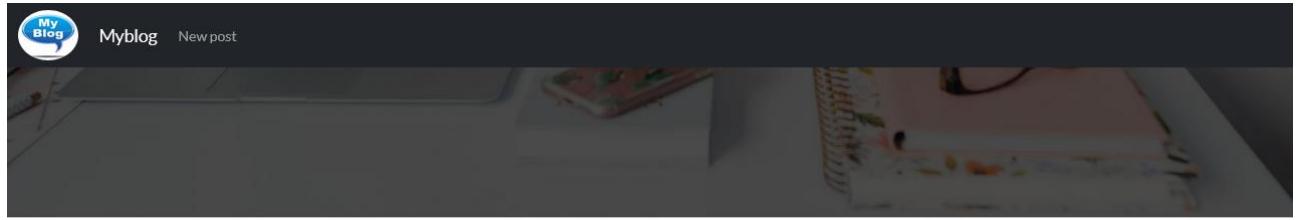
### Page My Blog :



My Blog : actualités et conseils.



## **DOSSIER PROFESSIONNEL (DP)**



## My Blog : actualités et conseils.

		
<h2>Emploi</h2> <p><b>Sarah Ben</b></p> <p>Le marché de l'emploi subit autant de bouleversements que notre société. Comment s'adapter aux nouvelles demandes des employeurs ? Comment réinventer sa carrière chaque jour, évoluer et gagner en compétences ? Des conseils de coaching d'entreprise sont disponibles sur notre site.</p> <p><a href="#">Edit</a> <a href="#">Delete</a></p>	<h2>Education</h2> <p><b>Will Smith</b></p> <p>L'éducation des plus jeunes et la formation de plus grands est un enjeu d'avenir : école, études, formations, universités, apprentissage, soutien scolaire, orientation, stages. Découvrez nos retours d'expériences et les tendances dans ce domaine</p> <p><a href="#">Edit</a> <a href="#">Delete</a></p>	<h2>Entreprise</h2> <p><b>Maxime Lerither</b></p> <p>Le monde de l'entreprise est en perpétuelle mutation. Carrière, salaire, adaptabilité, relation entre collègues, bien-être au travail, droit des salariés, création d'entreprise, e-commerce, gestion de crise sont au cœur des débats et méritent quelques éclairages pour mieux les comprendre.</p> <p><a href="#">Edit</a> <a href="#">Delete</a></p>

BlogIndex.js X

```
src > components > BlogIndex > BlogIndex.js > BlogIndex > reducer
1 import React, { useEffect, useReducer } from 'react'
2 import axios from 'axios'
3 import { Button } from 'semantic-ui-react'
4 function BlogIndex() {
5
6   const initialState = {
7     loading: true,
8     error: '',
9     blogs: []
10  }
11
12  const reducer = (state, action) => {
13    switch (action.type) {
14      case 'FETCH_SUCCESS':
15        return {
16          loading: false,
17          blogs: action.payload,
18          error: '',
19        };
20      case 'FETCH_ERROR':
21        return {
22          loading: false,
23          blogs: [],
24          error: 'Something went wrong!!!!!!',
25        };
26      default:
27        return state;
28    }
29  };
30  const [state, dispatch] = useReducer(reducer, initialState)
31
32  useEffect(() => {
33    axios.get('http://localhost:5000/myblog')
34      .then(response => {
35        console.log(response.data);
36        dispatch({ type: 'FETCH_SUCCESS', payload: response.data });
37      })
38      .catch(error => {
39        dispatch({ type: 'FETCH_ERROR' });
40      });
41  }, []);
42
43  return (
44    <React.Fragment>
45      
47      <h1 className="text-primary fs-1">My Blog : actualités et conseils.</h1>
48      <div className="row justify-content-evenly mt-4">
49        {state.loading ? 'loading....' : state.blogs.map((blog, index) => {
50          return (
51            <React.Fragment key={index}>
52              <div className="card col-lg-3 col-md-6 col-sm-12 col-xs-12 my-3">
53                <img className="img-fluid" src={'http://localhost:5000/${blog.imagename}' />
54                <h1 className='text-info'>{blog.titre}</h1>
55                <h3>{blog.username}</h3>
56                <p>{blog.content}</p>
57                <div className='d-flex justify-content-evenly'>
58                  <Button primary as='a' href={`/editblog/${blog._id}`}>Edit</Button>
59                  <form action={`/http://localhost:5000/deleteblog/${blog._id}`}
60                        method="DELETE" method="post">
61                    <input type="hidden" name="_method" value="DELETE" />
62                    <Button positive>Delete</Button>
63                  </form>
64                </div>
65              </div>
66            </React.Fragment>
67          )
68        })}
69      </div>
70    </React.Fragment>
71  );
72}
```

# DOSSIER PROFESSIONNEL (DP)

## Page new post :



### New Post

Title

Username

Parcourir... Aucun fichier sélectionné.

Envoyer

```
src > components > MyBlog > MyBlog.js > default
1  import React, { useState } from 'react'
2  import axios from 'axios'
3  import { useNavigate } from 'react-router-dom';
4  function MyBlog() {
5      let navigate = useNavigate();
6      const [file, setFile] = useState(null);
7      const [titre, setTitre] = useState('');
8      const [username, setUsername] = useState('');
9      const [content, setContent] = useState('');
10     const [imagename, setImagename] = useState('');
11     const handleFileChange = (event) => {
12         setFile(event.target.files[0]);
13         setImagename(event.target.files[0].name);
14     };
15     const handleUsernameChange = (event) => {
16         setUsername(event.target.value);
17     };
18     const handleTitreChange = (event) => {
19         setTitre(event.target.value);
20     };
21     const handleContentChange = (event) => {
22         setContent(event.target.value);
23     };
24     const handelSubmit = (event) => {
25         event.preventDefault();
26         if (file) {
27             const formData = new FormData();
28             formData.append('file', file);
29             formData.append('titre', titre);
30             formData.append('username', username);
31             formData.append('content', content);
32             formData.append('imagename', imagename);
33             axios.post('http://localhost:5000/submit-blog', formData)
34                 .then((response) => {
35                     console.log(response.data);
36                     return navigate('/blog')
37                 }).catch((error) => {
38                     console.log(error);
39                 });
40         }
41     };
42     return (
43         <div className="d-flex justify-content-center align-items-center vh-100">
44             <form className="col-lg-3 col-md-6 col-sm-12 col-xs-12 " onSubmit={handelSubmit}>
45                 <h2 className="text-success">New Post</h2>
46                 <div className="mb-3">
47                     <label className="form-label">Title</label>
48                     <input type="text" className="form-control" name="titre" onChange={handleTitreChange} />
49                 </div>
50                 <div className="mb-3">
51                     <label className="form-label">Username</label>
52                     <input type="text" className="form-control" name="username" onChange={handleUsernameChange} />
53                 </div>
54                 <div className="form-floating">
55                     <textarea className="form-control" name="content" onChange={handleContentChange}></textarea>
56                 </div>
57                 <div className="input-group my-3">
58                     <input className="form-control" type="file" onChange={handleFileChange} />
59                 </div>
60                 <button className="btn btn-primary" type="submit " >Envoyer</button>
61             </form>
62         </div>
63     );
64 }
65
66 export default MyBlog
```

## **DOSSIER PROFESSIONNEL (DP)**

## Page Edit post :

Myblog New post

## Edit Post

Title

Username

Le marché de l'emploi subit autant de bouleversements que notre société. Comment s'adapter aux nouvelles demandes des employeurs ? Comment réinventer sa carrière chaque jour, évoluer et gagner en compétences ? Des conseils de coaching d'entreprise sont disponibles sur notre site.

Parcourir... Aucun fichier sélectionné.



Envoyer



The screenshot shows two tabs in Visual Studio Code: 'EditBlog.js U' and 'EditBlog.js U'. The left tab displays the initial code for the 'EditBlog' component, which includes imports for React, useState, useEffect, and useNavigate from 'react-router-dom', and axios from 'axios'. It defines a function 'EditBlog' that uses these hooks to handle navigation, file selection, and form data. The right tab shows the same component with additional logic added for file submission. A new function 'handelSubmit' is defined, which prevents the default event, gets the file from the input, creates a FormData object, and appends 'file', 'titre', 'username', and 'content' to it. It then performs a PUT request to 'http://localhost:5000/blogedit/\${params.id}' with the formData. If successful, it logs the response data and navigates to '/blog'. If there's an error, it logs the error. Finally, the component returns a form with fields for title, username, content, and an image preview.

```
src > components > MyBlog > EditBlog.js > EditBlog

1 import React, {useEffect, useState} from 'react'
2 import { useParams } from 'react-router-dom'
3 import axios from 'axios'
4 import { useNavigate } from 'react-router-dom';
5
6 function EditBlog() {
7   const params = useParams();
8   let navigate = useNavigate();
9   const [file, setFile] = useState(null);
10  const [titre, setTitre] = useState('');
11  const [username, setUsername] = useState('');
12  const [content, setContent] = useState('');
13  const [imagename, setImageName] = useState('');
14  useEffect(()=>{
15    axios.get(`http://localhost:5000/blog/${params.id}`)
16    .then(response =>{
17      setTitre(response.data.titre)
18      setUsername(response.data.username)
19      setContent(response.data.content)
20      setImageName(response.data.imagename)
21    }).catch(error =>{
22      console.log(error);
23    })
24  },[])
25  const handelFileChange = (event) => {
26    setFile(event.target.files[0]);
27    setImageName(event.target.files[0].name);
28  };
29  const handelUsernameChange = (event) => {
30    setUsername(event.target.value);
31  };
32  const handelTitreChange = (event) => {
33    setTitre(event.target.value);
34  };
35  const handelContentChange = (event) => {
36    setContent(event.target.value);
37  };
38  const handelSubmit = (event) => {
39    event.preventDefault();
40
41    const handelSubmit = (event) => {
42      event.preventDefault();
43      if (file) {
44        const formData = new FormData();
45        formData.append('file', file);
46        formData.append('titre', titre);
47        formData.append('username', username);
48        formData.append('content', content);
49        formData.append('imagename', imagename);
50        axios.put(`http://localhost:5000/blogedit/${params.id}`, formData)
51          .then((response) => {
52            console.log(response.data);
53            return navigate('/blog')
54          }).catch((error) => {
55            console.log(error);
56          });
57      }
58      return (
59        <div className='d-flex justify-content-center align-items-center vh-100'>
60          <form className='col-lg-3 col-md-6 col-sm-12 col-xs-12' onSubmit={handelSubmit}>
61            <h2 className='text-success'>Edit Post</h2>
62            <div className='mb-3'>
63              <label className='form-label'>Title</label>
64              <input type='text' className='form-control' value={titre} onChange={handelTitreChange} />
65            </div>
66            <div className='mb-3'>
67              <label className='form-label'>Username</label>
68              <input type='text' className='form-control' value={username} name='username' onChange={handelUsernameChange} />
69            </div>
70            <div class='form-floating'>
71              <textarea className='form-control' name='content' rows='50' value={content} onChange={handelContentChange}></textarea>
72            </div>
73            <div className='input-group my-3'>
74              <input className='form-control' type='file' onChange={handelFileChange} />
75            </div>
76            <img src={`http://localhost:5000/${imagename}`} width="150px" alt="image" /><br/>
77            <button className='mt-5 btn btn-primary' type='submit'>Envoyer</button>
78          </form>
79        </div>
80      );
81    }
82  };
83
```

# DOSSIER PROFESSIONNEL (DP)

## Annexe 20 Controller :

### Ajouter et afficher les postes : (POST, GET)

```
//Blog*****
const Blog = require('../models/Blog');
app.post('/submit-blog', upload.single('file'), function (req, res) {

    if (!req.file) {
        res.status(400).send('No file uploaded');
    } else {
        res.send('file uploaded successfully')
        const Data = new Blog({
            titre: req.body.titre,
            username: req.body.username,
            imagename: req.body.imagename,
            content: req.body.content
        })
        Data.save().then(() => {
            console.log("Data saved successfully !");
            res.redirect('/');
        })

        }).catch(err => { console.log(err) });
    }
})
app.get('/myblog', function (req, res) {
    Blog.find()
        .then(data => {
            console.log(data);
            res.json(data);
        })
        .catch(err => console.log(err))
});
```

### Supprimer les postes : (DELETE)

```
app.delete('/blogdelete/:id', function (req, res) {
    Post.findOneAndDelete({ _id: req.params.id })
        .then(() => {
            res.redirect('http://localhost:3000/blog')
        })
        .catch(err => console.log(err));
});
```

# DOSSIER PROFESSIONNEL (DP)

## Editer les postes : (PUT)

```
app.get('/blog/:id', function (req, res) {
  Blog.findOne({ _id: req.params.id })
    .then((data) => {
      res.json(data);
    })
    .catch(err => console.log(err));
});

app.put('/blogedit/:id', upload.single('file'), function (req, res) {
  if (!req.file) {
    res.status(400).send('No file uploaded');
  } else {
    res.send('file uploaded successfully')
  const Data = ({
    titre: req.body.titre,
    username: req.body.username,
    imagename: req.body.imagename,
    content: req.body.content
  })
  Blog.updateOne({ _id: req.params.id }, { $set: Data })
    .then(() => {
      res.json("data updated")
    })
    .catch(err => console.log(err));
  };
});
```

## Développer une application mobile. Le responsive design

### Annexe 21 :

# DOSSIER PROFESSIONNEL (DP)

## Code Html :

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <link rel="stylesheet" href="media.css">
8      <title>Media querie</title>
9  </head>
10 <body>
11     <h1>Développement Web</h1>
12     <p>
13         Le développement web correspond à l'ensemble des tâches liées à la création et la gestion d'un site web. Cela
14         peut inclure la conception et le codage du site, la création et la gestion du contenu, la maintenance, etc.
15
16         Les tests constituent également une partie importante du développement d'un site car ils permettent de s'assurer
17         que ce dernier fonctionne correctement et que tout le contenu est exact et à jour.
18
19         La mission principale d'un développeur web consiste à créer des sites web visuellement attrayants, accessibles
20         et optimisés pour les moteurs de recherche.
21
22         Pour atteindre ces objectifs, le développeur utilise des langages informatiques tels que HTML, CSS ou encore
23         JavaScript. Il utilise également des outils tels que les systèmes de gestion de contenu (CMS) pour l'aider à
24         créer et gérer des sites web. Régulièrement, il procède à des séries de test pour s'assurer que ses sites web
25         fonctionnent correctement.
26     </p>
27     <div class="img">
28         
29     </div>
30
31 </body>
32 </html>
```

## Code CSS :

# DOSSIER PROFESSIONNEL (DP)

media.css X mediaquerie.html

jour4 > media.css > ...

```
1 body {  
2     background-color: aquamarine;  
3 }  
4 h1 {  
5     text-align: center;  
6     color: red;  
7 }  
8 p {  
9     color: blue;  
10    text-align: justify;  
11 }  
12 .img {  
13     text-align: center;  
14 }  
15 /* Media querie */  
16 @media screen and (min-width:400px) and (max-width:960px) {  
17     body {  
18         background-color: rgb(212, 179, 199);  
19     }  
20  
21     h1 {  
22         color: rgb(76, 0, 255);  
23     }  
24  
25     img {  
26         width: 400px;  
27         height: 300px;  
28     }  
29 }
```

Ordinateur :

# DOSSIER PROFESSIONNEL (DP)

## Développement Web

Le développement web correspond à l'ensemble des tâches liées à la création et la gestion d'un site web. Cela peut inclure la conception et le codage du site, la création et la gestion du contenu, la maintenance, etc. Les tests constituent également une partie importante du développement d'un site car ils permettent de s'assurer que ce dernier fonctionne correctement et que tout le contenu est exact et à jour. La mission principale d'un développeur web consiste à créer des sites web visuellement attrayants, accessibles et optimisés pour les moteurs de recherche. Pour atteindre ces objectifs, le développeur utilise des langages informatiques tels que HTML, CSS ou encore JavaScript. Il utilise également des outils tels que les systèmes de gestion de contenu (CMS) pour l'aider à créer et gérer des sites web. Régulièrement, il procède à des séries de test pour s'assurer que ses sites web fonctionnent correctement.



## Mobile

### Développement Web

Le développement web correspond à l'ensemble des tâches liées à la création et la gestion d'un site web. Cela peut inclure la conception et le codage du site, la création et la gestion du contenu, la maintenance, etc. Les tests constituent également une partie importante du développement d'un site car ils permettent de s'assurer que ce dernier fonctionne correctement et que tout le contenu est exact et à jour. La mission principale d'un développeur web consiste à créer des sites web visuellement attrayants, accessibles et optimisés pour les moteurs de recherche. Pour atteindre ces objectifs, le développeur utilise des langages informatiques tels que HTML, CSS ou encore JavaScript. Il utilise également des outils tels que les systèmes de gestion de contenu (CMS) pour l'aider à créer et gérer des sites web. Régulièrement, il procède à des séries de test pour s'assurer que ses sites web fonctionnent correctement.



## Tablette

### Développement Web

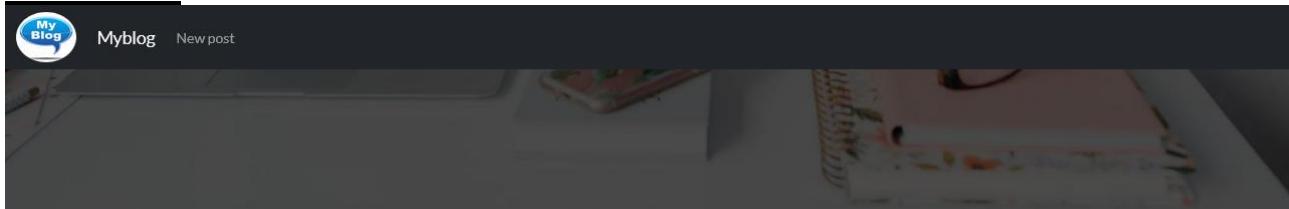
Le développement web correspond à l'ensemble des tâches liées à la création et la gestion d'un site web. Cela peut inclure la conception et le codage du site, la création et la gestion du contenu, la maintenance, etc. Les tests constituent également une partie importante du développement d'un site car ils permettent de s'assurer que ce dernier fonctionne correctement et que tout le contenu est exact et à jour. La mission principale d'un développeur web consiste à créer des sites web visuellement attrayants, accessibles et optimisés pour les moteurs de recherche. Pour atteindre ces objectifs, le développeur utilise des langages informatiques tels que HTML, CSS ou encore JavaScript. Il utilise également des outils tels que les systèmes de gestion de contenu (CMS) pour l'aider à créer et gérer des sites web. Régulièrement, il procède à des séries de test pour s'assurer que ses sites web fonctionnent correctement.



# DOSSIER PROFESSIONNEL (DP)

## Annexe 22 (Bootstrap)

Ordinateur :



### My Blog : actualités et conseils.



#### Emploi

Sarah Ben

Le marché de l'emploi subit autant de bouleversements que notre société. Comment s'adapter aux nouvelles demandes des employeurs ? Comment réinventer sa carrière chaque jour, évoluer et gagner en compétences ? Des conseils de coaching d'entreprise sont disponibles sur notre site.

Edit

Delete



#### Education

Will Smith

L'éducation des plus jeunes et la formation de plus grands est un enjeu d'avenir : école, études, formations, universités, apprentissage, soutien scolaire, orientation, stages. Découvrez nos retours d'expériences et les tendances dans ce domaine

Edit

Delete



#### Entreprise

Maxime Lerither

Le monde de l'entreprise est en perpétuelle mutation. Carrière, salaire, adaptabilité, relation entre collègues, bien-être au travail, droit des salariés, création d'entreprise, e-commerce, gestion de crise sont au cœur des débats et méritent quelques éclairages pour mieux les comprendre.

Edit

Delete

## Mobile



### My Blog : actualités et conseils.



#### Emploi

Sarah Ben

Le marché de l'emploi subit autant de bouleversements que notre société. Comment s'adapter aux nouvelles demandes des employeurs ? Comment réinventer sa carrière chaque jour, évoluer et gagner en compétences ? Des conseils de coaching d'entreprise sont disponibles sur notre site.

Edit

Delete



### My Blog : actualités et conseils.



#### Emploi

Sarah Ben

Le marché de l'emploi subit autant de bouleversements que notre société. Comment s'adapter aux nouvelles demandes des employeurs ? Comment réinventer sa carrière chaque jour, évoluer et gagner en compétences ? Des conseils de coaching d'entreprise sont disponibles sur notre site.

Edit

Delete



#### Education

Will Smith

L'éducation des plus jeunes et la formation de plus grands est un enjeu d'avenir : école, études, formations, universités, apprentissage, soutien scolaire, orientation, stages. Découvrez nos retours d'expériences et les tendances dans ce domaine

Edit

Delete



#### Entreprise

# DOSSIER PROFESSIONNEL (DP)

## Code :

```

<h1 className='text-primary fs-1 '>My Blog : actualités et conseils.</h1>
<div className='row justify-content-evenly mt-4'>
  {state.loading ? 'loading....' : state.blogs.map((blog, index) => {
    return (
      <React.Fragment key={index}>
        <div className="card col-lg-3 col-md-6 col-sm-12 col-xs-12 my-3" >
```

## Annexe 23 : Préparer et exécuter les plans de tests d'une application. Les tests de sécurité :

```
"use strict";
var express = require('express');
var app = express();

// Middleware pour éviter la pollution des paramètres HTTP et empêcher l'ajout d'informations inutiles
const hpp = require('hpp');
app.use(hpp());

// Middleware pour sécuriser les requêtes HTTPS et désactiver la politique Cross-Origin Resource Sharing (CORS) pour les ressources statiques
const helmet = require('helmet');
app.use(helmet({ crossOriginResourcePolicy: false }));

// Middleware pour contrôler le cache et vider le cache des réponses
const nocache = require('nocache');
app.use(nocache());

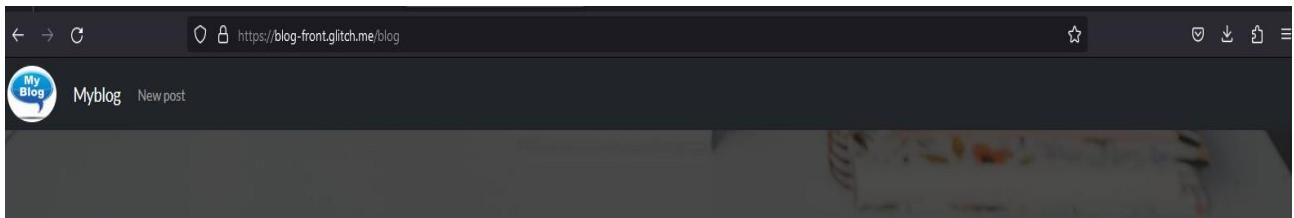
// Middleware toobusy pour protéger le serveur contre les attaques DoS (Denial of Service)
const toobusy = require('toobusy-js');
app.use(function (req, res, next) {
  if (toobusy()) {
    res.status(503).send("Server too busy");
  } else {
    next(); // Exécution de l'action suivante
  }
});

// Middleware pour utiliser des captchas SVG
const session = require('express-session');
const svgCaptcha = require('svg-captcha');
app.use(
  session({
    secret: "secret-key",
    resave: false,
    saveUninitialized: true
  })
);
```

# DOSSIER PROFESSIONNEL (DP)

## Annexe 24 : Préparer et exécuter le déploiement d'une application.(Glitch)

### La page Myblog :



### My Blog : actualités et conseils.



#### Emploi

##### Sarah Ben

Le marché de l'emploi subit autant de bouleversements que notre société. Comment s'adapter aux nouvelles demandes des employeurs ? Comment réinventer sa carrière chaque jour, évoluer et gagner en compétences ? Des conseils de coaching d'entreprise sont disponibles sur notre site.

[Edit](#)[Delete](#)

#### Entreprise

##### Maxime Lerither

Le monde de l'entreprise est en perpétuelle mutation. Carrière, salaire, adaptabilité, relation entre collègues, bien-être au travail, droit des salariés, création d'entreprise, e-commerce, gestion de crise sont au cœur des débats et méritent quelques éclairages pour mieux les comprendre.

[Edit](#)[Delete](#)

#### Education

##### Will Smith

L'éducation des plus jeunes et la formation de plus grands est un enjeu d'avenir : école, études, formations, universités, apprentissage, soutien scolaire, orientation, stages. Découvrez nos retours d'expériences et les tendances dans ce domaine.

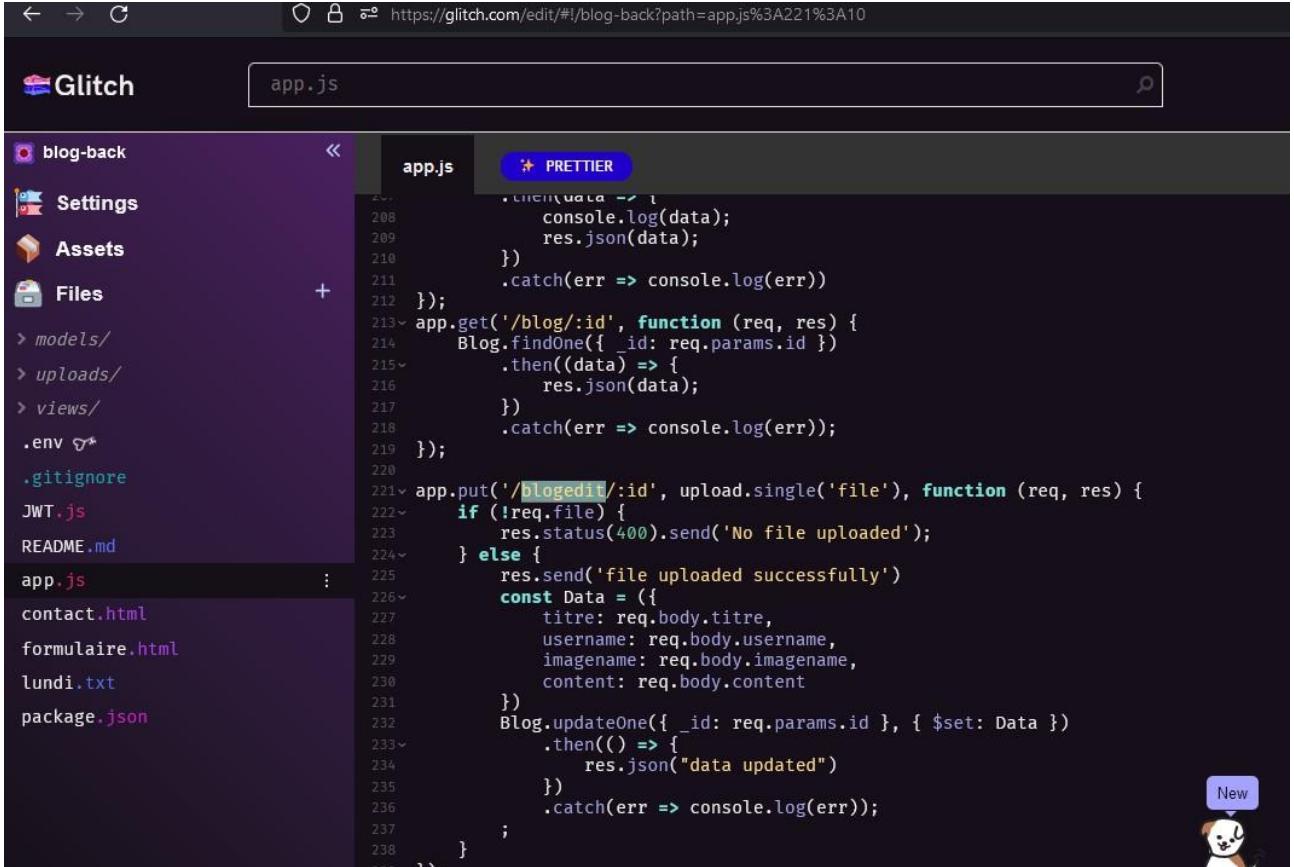
[Edit](#)[Delete](#)

### Le front-end

A screenshot of the Glitch web interface. On the left, the file structure shows 'EditBlog.js' as the active file. The code in 'EditBlog.js' handles form submission and axios requests to update a blog post. On the right, the preview window shows the 'MyBlog' application running at 'blog-front.glitch.me/'. The preview shows the three sections: 'Emploi', 'Entreprise', and 'Education', each with a thumbnail, title, and a brief description. A sidebar on the right shows a smiling man at a desk.

# DOSSIER PROFESSIONNEL (DP)

## La back-end



The screenshot shows a Glitch editor interface with the URL <https://glitch.com/edit/#/blog-back?path=app.js%3A221%3A10>. The left sidebar lists files: blog-back, Settings, Assets, Files, models/, uploads/, views/, .env, .gitignore, JWT.js, README.md, app.js, contact.html, formulaire.html, lundi.txt, and package.json. The main area shows the app.js code:

```
207     .then((data) => {
208       console.log(data);
209       res.json(data);
210     })
211   .catch(err => console.log(err))
212 });
213 app.get('/blog/:id', function (req, res) {
214   Blog.findOne({ _id: req.params.id })
215     .then((data) => {
216       res.json(data);
217     })
218     .catch(err => console.log(err));
219 });
220
221 app.put('/blogedit/:id', upload.single('file'), function (req, res) {
222   if (!req.file) {
223     res.status(400).send('No file uploaded');
224   } else {
225     res.send('file uploaded successfully')
226     const Data = ({
227       titre: req.body.titre,
228       username: req.body.username,
229       imagename: req.body.imagename,
230       content: req.body.content
231     })
232     Blog.updateOne({ _id: req.params.id }, { $set: Data })
233       .then(() => {
234         res.json("data updated")
235       })
236       .catch(err => console.log(err));
237     }
238   });
239 }
```