



DOSSIER PROFESSIONNEL (DP)

MINISTÈRE CHARGÉ
DE L'EMPLOI

Nom de naissance
Nom d'usage
Prénom
Adresse

- HOCINI
 - MOUHOUBI
 - SONIA
 - *16 rue bernard Palissy Immeuble
La Mallounière 2
13400 Aubagne*

Titre professionnel visé

Concepteur(trice) Développeur(se) d'Application

MODALITÉ D'ACCÈS :

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

Présentation du dossier

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

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

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

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

Pour prendre sa décision, le jury dispose :

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

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

Ce dossier comporte :

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

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

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

Sommaire

Exemples de pratique professionnelle

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité	p.	5
- Maquetter une application - Application Web HairSolutions	p.	5
- Maquetter une application - Wireframe Projet Application Web RH	p.	15
- Développer la partie front-end d'une interface utilisateur web - Site Admin HairSolutions	p.	17
- Développer la partie back-end d'une interface utilisateur web - Site Admin HairSolutions	p.	20
- Développer des composants d'accès aux données - API HairSolutions	p.	22
- Développer une interface de type desktop - Python		25
Concevoir et développer la persistance des données en intégrant les recommandations de sécurité	p.	28
- Concevoir une base de données	p.	28
- Mettre en place une base de données	p.	30
- Développer des composants dans le langage d'une base de données	p.	35
Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité	p.	37
- Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement	p.	37
- Développer des composants métiers	p.	41
- Construire une application organisée en couches	p.	42
- Développer une application mobile	p.	44
- Préparer et exécuter les plans de tests d'une application	p.	50
Titres, diplômes, CQP, attestations de formation (<i>facultatif</i>)	p.	52
Déclaration sur l'honneur	p.	53
Documents illustrant la pratique professionnelle (<i>facultatif</i>)	p.	54
Annexes (<i>Si le RC le prévoit</i>)	p.	55

EXEMPLES DE PRATIQUE

PROFESSIONNELLE

Activité-type 1

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

Exemple n°1 - Maquetter une application - Application Web HairSolutions

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

Dans le cadre de ma formation, j'ai réalisé pour mon projet final de fin d'année, le **maquettage** de l'application admin de HairSolutions (application fictive sur le thème de l'échange sur la beauté et les astuces capillaires).

En amont, une veille a été effectuée afin de cibler les besoins de notre application. Cela a inclus des recherches sur la psychologie du design pour comprendre les attentes des utilisateurs. Notre objectif était de créer une application qui toucherait un large public tout en conservant un aspect élégant et sophistiqué, en s'inspirant des tendances du domaine de la coiffure, tout en restant accessible à tous.

J'ai commencé par élaborer une **charte graphique** qui a servi de base pour la conception des maquettes du projet. Des **logos** ainsi que des éléments graphiques ont été conçus spécifiquement pour ce projet.

LOGO



Exemple CHARTE GRAPHIQUE

Couleurs

Couleurs utilisées pour l'application mobile, le site web et les éléments graphiques.

Logotype



Application en général



Site d'administration en général



Eléments graphiques



Favicon



Représentations de cheveux



Icons créés pour l'application



Ces éléments visuels ont été spécialement créé pour l'application HairSolutions

Typographies

Modernes, élégantes et lisibles, ces typographies peuvent être utilisées sur écran ou impression

Typographie des titres

Nunito

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

Typographie des sous-titres

Montserrat

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

Typographie des textes

Maitree

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

WIREFAME



Fiche détaillée membre HairSolutions

Utilisateurs

Chat général 24

Posts membres

Posts HairSolutions

Mentions légales

Compte admin

Déconnexion



Non connecté

Nom

Prenom

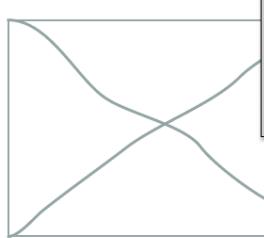
nom-prenom@gmail.com

Réinitialisation du mot de passe



Fiche personnelle cheveux

Type de cheveux :



Raide

En cliquant sur supprimer, l'utilisateur sera supprimé définitivement de la base de données.

Annuler

Supprimer

sit amet. Et repellendus nulla
ita aut maxime molestias qui
cum dolorem nulla. Et nihil
ternos est error rerum aut enim
iunquam et alias tenetur qui
pluptatibus quas id veritatis
eveniet aut commodi harum rem provident autem
sed blanditiis earum aut voluptatem magnam id
iste deserunt.
Ut aperiam commodi et culpa accusamus est optio
Quis nam exercitationem doloribus et quia ...

Publications

Titre 1

Titre 1

Titre 1

Titre 1

Titre 1

Ti

Supprimer le compte de l'utilisateur





Publications Admin HairSolutions

Utilisateurs

Chat général 24

Posts membres

Posts HairSolutions

Mentions légales

Compte admin

Déconnexion

Search...

Titre 1



Titre 3

Titre 4

Titre 5

Titre 6

Titre 7

Titre 8

Titre 9

Titre 10

Titre 11

Titre 12

Titre 13

Copyright © 2023

Logo

 John Doe
Administrateur

 Utilisateurs

 Chat général 24

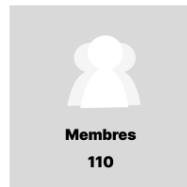
 Posts membres

 Posts HairSolutions

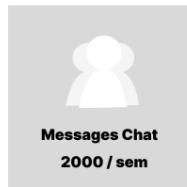
 Mentions légales

 Compte admin

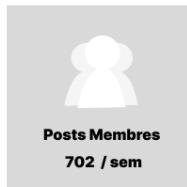
 Déconnexion



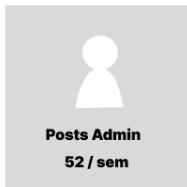
Membres
110



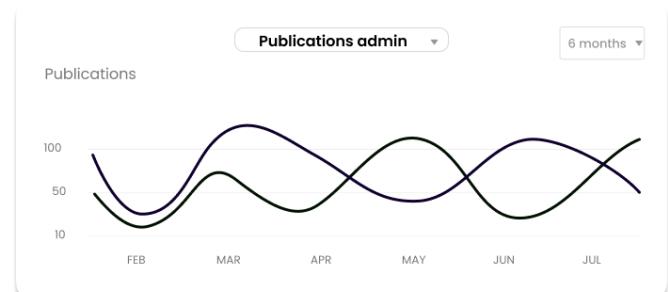
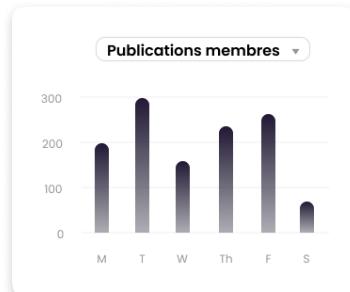
Messages Chat
2000 / sem



Posts Membres
702 / sem



Posts Admin
52 / sem



Copyright © 2023

MAQUETTE

Fiche détaillée membre HairSolutions

Utilisateurs

-  Chat général (24)
-  Posts membres
-  Posts HairSolutions
-  Mentions légales
-  Compte admin
-  Déconnexion



Non connecté

Tara

James

tara-james@gmail.com

Réinitialisation du mot de passe



Fiche : Présenter sa routine capillaire

Type de cheveux :

Raide



En cliquant sur supprimer, l'utilisateur sera supprimé définitivement de la base de données.

Annuler

Supprimer

sit amet. Et repellendus nulla ita aut maxime molestias qui cum dolorem nulla. Et nihil tternos est error rerum aut enim ipsa. Quo adipisci numquam et alias tenetur qui autem quae. Sed voluptatibus quas id veritatis eveniet aut commodi harum rem provident autem...

Publications



Ma routine beauté



Coloration ratée



Coiffure coupe courte



Astuces pointes cassantes



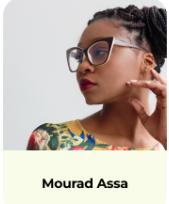
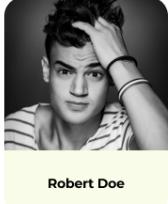
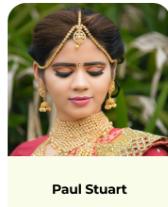
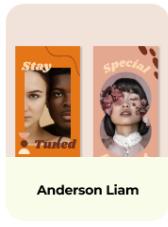
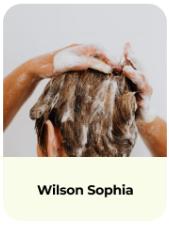
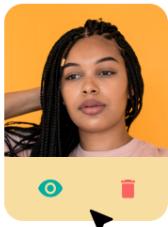
Mèches rebelles



L...

Supprimer le compte de l'utilisateur 

Publications Membres HairSolutions

[Utilisateurs](#)[Chat général 24](#)[Posts membres](#)[Posts HairSolutions](#)[Mentions légales](#)[Compte admin](#)[Déconnexion](#) Search...

Copyright © 2023

2. Précisez les moyens utilisés :

Figma : Pour le maquettage, J'ai utilisé Figma car cet outil est largement utilisé dans l'industrie du design, que ce soit pour la conception d'applications mobiles, de sites web ou d'autres types d'interfaces utilisateur interactives. Il est apprécié pour sa flexibilité, sa facilité de partage et sa capacité à faciliter la collaboration entre les membres d'une équipe de conception.

Google Docs : Le Cahier des charges a été réalisé sur **Google Docs** pour faciliter le travail en groupe.

Recherche internet :

<https://openclassrooms.com/fr/courses/5248811-appliquez-la-psychologie-au-design>

3. Avec qui avez-vous travaillé ?

Seule

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *A l'école La Plateforme*

Chantier, atelier, service ▶ *Developpement Web et Mobile*

Période d'exercice ▶ Du : 02/01/2023 au : 13/01/2023

5. Informations complémentaires (facultatif)

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

Activité-type 1

Exemple n°3 - Maquetter une application - Wireframe Projet Application Web RH

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

Le maquettage de ce projet a été effectué lors de mes travaux en entreprise. Afin de répondre aux besoins de la startup où j'ai effectué mon alternance, j'ai élaboré un wireframe résultant de mes recherches et de ma conception pour la création d'une application RH. Cette application web permettrait à l'entreprise de centraliser les demandes de congés, les jours de RTT et les demandes de télétravail de leurs employés.

Une veille technique a été effectuée dans le but d'analyser les propositions de la concurrence et de s'en rapprocher, afin de concevoir un projet réalisable et viable. Une documentation a été créée par mes soins en amont pour ce projet.

2. Précisez les moyens utilisés :

Maquettages : Mon choix s'est porté sur **Figma** pour la création de ces maquettes car c'est un outil disponible en ligne qui permet de collaborer en équipes. J'ai ainsi pu partager mon travail avec mon

tuteur et mon employeur qui ont pu voir l'avancement de mon travail et me faire des suggestions.

3. Avec qui avez-vous travaillé ?

Seule

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *Eco Impact*

Chantier, atelier, service ▶ *Developpement Web*

Période d'exercice ▶ Du : *06/02/2023* au : *09/02/2023*

5. Informations complémentaires (facultatif)

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

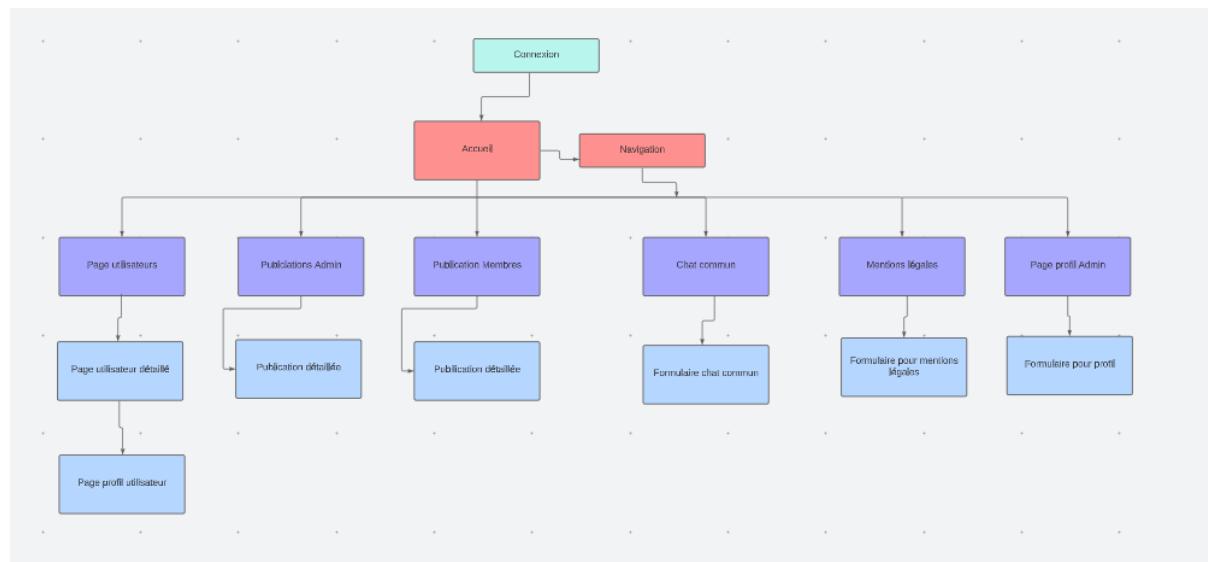
Activité-type 1

Exemple n°1 - Développer la partie front-end d'une interface utilisateur web - Site Admin HairSolutions

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

Une veille technique a été effectuée afin de comprendre les besoins pour concevoir un site d'administration pour l'application mobile HairSolutions. Les recherches portaient sur l'interface utilisateur (UI) et l'expérience utilisateur (UX) de l'administration du site. L'objectif était de trouver un équilibre entre cohérence et éviter la répétition excessive dans le design graphique précédemment élaboré pour l'application mobile HairSolutions. De plus, il était important de savoir ce qui était légalement autorisé et dans quel format un administrateur de site pouvait récupérer les informations de l'application créée précédemment. Nous devions également déterminer les possibilités de modération offertes à l'administrateur sur son site. Des recherches ont donc été menées dans ce sens.

Afin d'avoir une organisation des informations de manière logique et structurée, notre application comportera plusieurs pages avec un niveau de hiérarchisation des informations. Voici un aperçu de l'interface :



Le fait d'avoir une organisation permet d'avoir un flux de navigation fluide qui permet à l'utilisateur de se retrouver facilement dans l'application. Une navigation simple et intuitive facilite l'expérience de l'utilisateur.

De plus, nous avons cherché à éviter la surcharge d'informations en utilisant des liens cliquables qui permettent d'afficher davantage d'informations si l'utilisateur le souhaite. Cela évite de perdre l'utilisateur en lui présentant trop d'informations d'un seul coup.

Notre objectif principal était d'offrir une expérience utilisateur efficace et conviviale.

De ces recherches, ainsi que de la charte graphique précédemment réalisée, est née la conception de l'admin de HairSolutions.

Nous avons opté pour une **Single Page Application** afin de nous concentrer sur l'aspect UX de notre application et éviter les temps de rechargement trop lent afin de garantir une expérience utilisateur agréable aux visiteurs du site (ici à l'admin). Bien que le SEO (SEARCH ENGINE OPTIMISATION) peut être souvent reproché au SPA car le code sourcé n'est pas toujours explicite et très optimisé pour un référencement naturel.

Nous nous sommes donc mises d'accord de rester sur les **MERN stack** pour le projet admin. Cela nous permettait de développer rapidement des applications web full stack avec un seul langage de programmation (javascript).

J'ai donc créé la structure du projet afin que nous puissions travailler ensemble. Des pages et des composants ont été créés en React.JS. Le projet a été typé avec le langage TypeScript afin de renforcer la sécurité de notre application et nous faciliter le débogage.

La partie modérée de la messagerie interne va être créée avec **Socket.io**. Des routes ont été ajoutées à l'**API** conçue pour l'application mobile afin que si un élément est mis à jour dans l'admin, il le soit sur l'application instantanément, comme par exemple lorsque l'admin publie un post d'astuce afin que les membres puissent le voir sur l'application.

Les formulaires ont été gérés par **React Hook Form**. Leur design a été conçu avec **Matérial UI** et une partie du site sera créée avec **MUI** ainsi que du **CSS**.

Nous avons utilisé **React Context API** afin de gérer l'état global de notre application. Cela nous a permis de faire passer des données aux autres composants de l'application tel que le **token** afin de vérifier si par exemple l'admin est bien connecté.

2. Précisez les moyens utilisés :

React, React Router, Socket.io, Material UI, CSS, TypeScript

Recherche internet :

https://linchnil.github.io/Guide-RGPD-du-developpeur/#Fiche_n%C2%B00%C2%A0:_D%C3%A9velopper_en_conformit%C3%A9_avec_le_RGPD

3. Avec qui avez-vous travaillé ?

Meriem BARKA

4. Contexte

Nom de l'entreprise, organisme ou association - *Ecole La Plateforme*

Chantier, atelier, service - *Développement Web et Mobile*

Période d'exercice

► Du : 2023 au 2023

5. Informations complémentaires (*facultatif*)

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 - Site Admin HairSolutions

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

Pour la partie backend du projet web, nous avons réutilisé le même, donc avec la même **API** de l'application mobile afin d'avoir un rendu synchronisé lors des mises à jour du flux d'actualité ou de la gestion des utilisateurs par l'administrateur du site. Des routes ont dû être ajoutées.

Le backend du projet HairSolutions est structuré de la manière suivante :

app.js => Fichier principal où est configuré Express.js, et où sont définis les routes et middlewares
dossier routes => où sont créées les routes du projet

dossier controllers => contient la logique métier de chaque route. Les controllers reçoivent les requêtes HTTP, interagissent avec les services appropriés et renvoient les réponses correspondantes.

dossier models => Il définit la structure de données de notre application. L'interaction avec notre base de donnée **Mongo DB** est gérée grâce à la bibliothèque **ORM** (Object Relational Mapping) **Mongoose**.

dossier middleware => contient les fichiers des middlewares, tels que l'authentification, l'autorisation, etc.

dossier asset => pour le stockage des images et documents téléchargeables.

dossier config => utilisé pour stocker la configuration de la base de données ainsi que les variables d'environnement.

server => fichier qui contient le lancement du serveur avec **Express.js** en utilisant le module HTTP de **Node.js**. et qui intègre les **Socket.io** pour la gestion de la communication en temps réel (messagerie instantanée de l'application)

2. Précisez les moyens utilisés :

Express.JS, Node.JS, Mongoose, Socket.io

3. Avec qui avez-vous travaillé ?

Meriem BARKA

4. Contexte

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

Chantier, atelier, service ▶ *Developpement Web*

Période d'exercice ▶ Du : 2023 au 2023

5. Informations complémentaires (facultatif)

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 - API HairSolutions

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

Ayant utilisé les stack MERN (MongoDB, Express.js, React, Node.js) pour notre projet, voici un exemple de développement de composants d'accès aux données pour interagir avec notre base de données.

Configuration de la connexion à la base de données (config/db.js):

```
// Importer mongoose
const mongoose = require("mongoose");
// Importer package pour variables d'environnement
require("dotenv").config();

mongoose
  .connect(process.env.DATABASE_URI, {
    autoIndex: false,
    maxPoolSize: 10,
    serverSelectionTimeoutMS: 5000,
    socketTimeoutMS: 45000,
    family: 4,
  })
  .then(() => {
    console.log("Connexion à MongoDB réussie");
  })
  .catch((err) => {
    console.log("Connexion à MongoDB échouée");
  });
});
```

Définition du modèle de données (models/User.js) :

```

        minlength: 2,
        maxlength: 100,
    },
    lastname: {
        type: String,
        require: true,
        trim: true,
        minlength: 2,
        maxlength: 100,
    },
    photo: { type: String, trim: true, default: false },
    admin: { type: Boolean, require: true, default: false },
    token: { type: String, trim: true },
},
{ timestamps: true }
);

module.exports = mongoose.model("users", Users);

```

Création de fonctions pour interagir avec les données (controllers/users.js):

```

// Fonction pour voir tous les utilisateurs
exports.getAllUsers = (req, res) => {
    Users.find()
        .select("firstname lastname")
        .then((resultat) => {
            res.status(200).json(resultat);
        })
        .catch((error) => {
            res.status(400).json({
                error: error,
            });
        });
};

```

```

// Envoyer un message privée
exports.sendMessagePrivate = (req, res) => {
    const messagesPrivates = new MessagesPrivates({
        idSender: req.auth.userId,
        idReceiver: req.body.idReceiver,
        message: req.body.message,
    });

    messagesPrivates
        .save()

```

```
.then(() => res.status(201).json({ message: "Message envoyé !" }))  
.catch((error) => {  
  res.status(400).json({  
    error: error,  
  })  
})  
};  
};
```

Utilisation des fonctions dans les routes (routes/user.js):

```
// Route pour voir tous les utilisateurs  
router.get("/users", auth, ctrlUsers.getAllUsers);
```

2. Précisez les moyens utilisés :

Express.JS, Node.JS, Mongoose, Socket.io

3. Avec qui avez-vous travaillé ?

Meriem BARKA

4. Contexte

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

Chantier, atelier, service ▶ *Développement Web et Mobile*

Période d'exercice ▶ Du : 2023 au 2023

5. Informations complémentaires (facultatif)

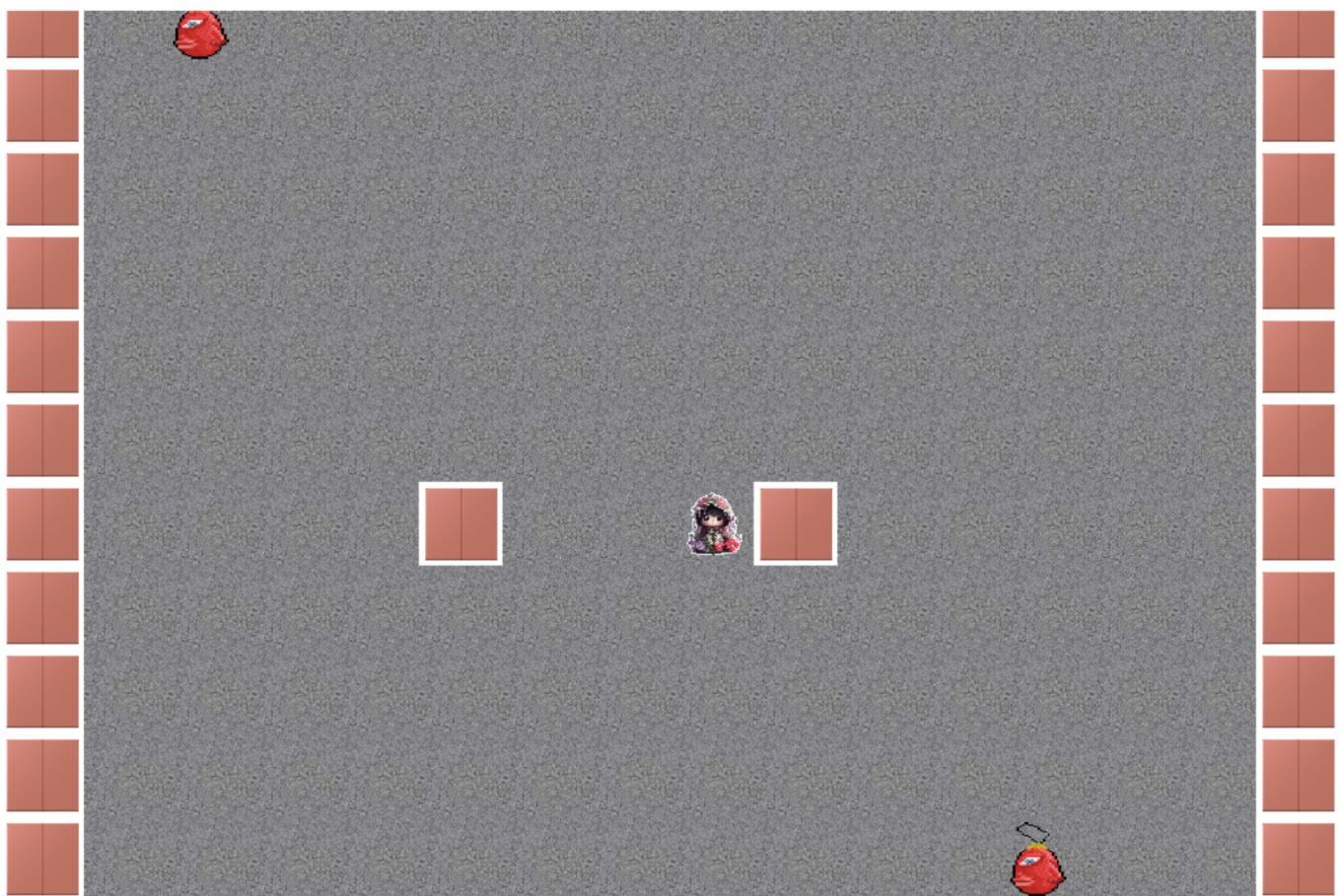
Activité-type 3

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

Exemple n°1 - Pygame

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

Dans le cadre du développement d'un programme qui se lance sur le bureau j'ai réalisé un jeu en python. A l'aide de pygames j'ai généré une boucle de jeu dans laquelle on va calculer l'espace disponible pour se déplacer et ce qu'il s'y passe à chaque instant. Le but du jeu est pour le moment de ramasser des objets. La partie est finie quand il n'y a plus d'objet.



Cette partie du code est la boucle principale du jeu, responsable de la mise à jour de l'affichage graphique et de la gestion des événements utilisateur en temps réel.

J'ai donc installé la bibliothèque Pygame pour créer le jeu en Python.

```
while continuer:
    if (shock_wave_animation_state.shockWave == True): ...

    for event in pygame.event.get():
        if event.type == pygame.KEYDOWN:

            if event.key == pygame.K_c: ...
            for item in goal: ...

            if event.key == pygame.K_ESCAPE: ...
            elif event.key == pygame.K_DOWN: ...
            elif event.key == pygame.K_UP:
                couscous = checkWalls('haut')
                if couscous: ...

            elif event.key == pygame.K_LEFT: ...
            elif event.key == pygame.K_RIGHT: ...

            fenetre.blit(perso, position_perso)
            pygame.display.update()
            pygame.display.flip()
```

2. Précisez les moyens utilisés :

Pygame, Python

3. Avec qui avez-vous travaillé ?

Seule

4. Contexte

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

Chantier, atelier, service ▶ *Développement Web et Mobile*

Période d'exercice ▶ Du : 2023 au 2023

5. Informations complémentaires (facultatif)

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 :

Ayant utilisé **MongoDB**, pour la création de notre base de données, le modèle conceptuel de données (MCD) traditionnel utilisé pour les bases de données traditionnelles n'est pas directement applicable à MongoDB. MongoDB étant une base de données non SQL orientée document, nous avons créé un MCD, MLD, MPD comme si nous avions utilisé une base de données SQL.

SCREEN MCD

SCREEN MLD

SCREEN MPD

2. Précisez les moyens utilisés :

MongoDB, LucidChart

3. Avec qui avez-vous travaillé ?

Meriem Barka

4. Contexte

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

Chantier, atelier, service ▶ *Développement Web et Mobile*

Période d'exercice ▶ Du : 2023 au : 2023

5. Informations complémentaires (facultatif)

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

Activité-type 2

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

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

Pour la mise en place de la base de données du projet HairSolutions, nous avons d'abord pris le choix de nos stacks (**MERN** => Mongo, Express, React et Node). Le choix de **MongoDB** comme système de gestion de base de données a été fait pour sa flexibilité et sa compatibilité avec JavaScript.

Après **installation** et **configuration** de **MongoDB**, nous avons créé notre **base de données**, puis nous avons conçu les **schémas** de la base de données.

Les schémas ont été créés avec la bibliothèque **Mongoose**. Nous avons donc identifié les **entités principales** de notre application :

- **Users** => contient les informations sur les utilisateurs
- **MessagesGenerals** => contient les messages général des utilisateurs de l'application
- **MessageGroups** => contient les messages de groupe des utilisateurs de l'application
- **MessagePrivates** => contient les messages privée des utilisateurs de l'application
- **Publications** => contient les publications des utilisateurs de l'application
- **Categories** => contient les catégories des publications

Puis nous avons défini les **attributs** de chaque entité.

```
const Users = mongoose.Schema ({  
  email: { type: String, require: true, trim: true, unique: true },  
  password: { type: String, require: true, trim: true },  
  firstname: {  
    type: String,  
    require: true,  
    trim: true,  
    minlength: 2,  
    maxlength: 100,  
  },  
  lastname: {  
    type: String,  
    require: true,  
    trim: true,  
    minlength: 2,  
    maxlength: 100,  
  },  
  photo: { type: String, trim: true, default: false },  
  admin: { type: Boolean, require: true, default: false },
```

```
  token: { type: String, trim: true },
},
{ timestamps: true }
);
```

Pour permettre l'interaction avec la base de données, nous avons mis en place une **API** (Application Programming Interface) avec Node.js/Express.js.

Configuration de la connexion à la base de données :

```
// Importer mongoose
const mongoose = require("mongoose");
// Importer package pour variables d'environnement
require("dotenv").config();

mongoose
  .connect(process.env.DATABASE_URI, {
    autoIndex: false,
    maxPoolSize: 10,
    serverSelectionTimeoutMS: 5000,
    socketTimeoutMS: 45000,
    family: 4,
  })
  .then(() => {
    console.log("Connexion à MongoDB réussie");
  })
  .catch((err) => {
    console.log("Connexion à MongoDB échouée");
  });
});
```

Exemple d'intégration avec le frontend :

J'ai intégré les appels aux API dans l'application React.js, en utilisant Axios pour effectuer des requêtes et afficher les données de la base de données dans l'interface utilisateur :

```

const authLogin = (email, password) => {

  axios.post(API_URL + '/login', {
    email,
    password,
  })
  .then(res => {
    if (res.status === 201) {
      setUserExist(true);
      setUserDontExist(false);
      setUserToken(res.data.token)
      // setUserID(res.data.userId)
      const decoded_token = jwt_decode(res.data.token);
      const ID_USER = decoded_token.userId;
      setIdDecoded(ID_USER);
      setTokenDecoded(decoded_token);
      AsyncStorage.setItem('authUserToken', JSON.stringify(res.data.token));
      setIsLogging(true);
    }
  })
  .catch(error => {
    if (error.response.status === 401) {
      setUserExist(false);
      setUserDontExist(true);
      setIsLogging(false);
      setToken(null);
    }
    console.log(`login error ${error}`);
  });
};

```

Configuration de l'application Express.js pour utiliser les routes des utilisateurs :

```

// Importer la package HTTP de Node JS pour la création du server
const http = require('http');
// Importer l'application app.js
const app = require('./app');
// Importer package pour variables d'environnement
require('dotenv').config();
// importer socket.io pour le chat
const { Server } = require("socket.io");

// Faire tourner l'app sur le port défini
app.set("port", process.env.PORT);
// Création du server
const server = http.createServer(app);

// Ecouter requête server sur port 3000
server.listen(process.env.PORT);

```

Sécurité pour protéger l'accès à la base de donnée :

Nous avons utilisé la bibliothèque `jsonwebtoken` pour générer et vérifier les JWT.

```

// On récupère le token grâce à la méthode split de js pour enlever l'espace
const token = req.headers.authorization.split(' ')[1];
// Décoder le token
const payload = jwt.verify(token, `${process.env.JWT_KEY_TOKEN}`);

```

Authentification pour le login :

```

exports.login = (req, res) => {
  const emailcryptoJS = cryptoJS.HmacSHA256(req.body.email, ` ${process.env.SECRET_CRYPTOJS_EMAIL}`).toString()
()

  Users.findOne({ email: emailcryptoJS })
  .then(user => {
    if (!user) {
      return res.status(401).json({ error: "Paire login/mot de passe incorrecte" });
    }
    bcrypt.compare(req.body.password, user.password).then(valid => {
      if (!valid) {
        return res.status(401).json({ error: "Paire login/mot de passe incorrecte" });
      }
      const userToken = jwt.sign({ userId: user._id }, `${process.env.JWT_KEY_TOKEN}`, { expiresIn: '12h' });

      // Mettre à jour le token de l'utilisateur dans le backend
      updateUserToken(user._id, userToken);

      // // Stockage du token dans la session
      // req.session.userToken = userToken;

      return res.status(201).json({ token: userToken });
    });
  })
  .catch(error => res.status(500).json({ error }));
};


```

Middleware d'authentification :

```

// Importation du token pour l'authentification
const jwt = require('jsonwebtoken');
// Importer package pour variables d'environnement
require('dotenv').config();

module.exports = (req, res, next) => {
  try {
    // Récupérer le token dans le headers authorization : bearer token
    // On récupère le token grâce à la méthode split de js pour enlever le
    const token = req.headers.authorization.split(' ')[1];
    // Décoder le token
    const payload = jwt.verify(token, `${process.env.JWT_KEY_TOKEN}`);

    // Objet request qui va être transmit au route appelé par la suite
    req.auth = {
      userId: payload.userId,
      // admin: payload.admin,
    };
    next();
  } catch (error) {
    res.status(401).json({ error: "JWT Error" });
  }
};


```

Utilisation du middleware :

```
// Chat
// Route pour voir tous les utilisateurs
router.get('/users', auth, ctrlUsers.getAllUsers);
// Route pour voir un utilisateur
router.get('/users/:id', auth, ctrlUsers.getUser);
// Route pour créer un utilisateur
router.post('/users', auth, ctrlUsers.createUser);
// Route pour mettre à jour un utilisateur
router.put('/users/:id', auth, ctrlUsers.updateUser);
// Route pour supprimer un utilisateur
router.delete('/users/:id', auth, ctrlUsers.deleteUser);
```

2. Précisez les moyens utilisés :

MongoDB, Mongoose, React, React Native, Node, Express, Visual Studio Code

3. Avec qui avez-vous travaillé ?

Meriem BARKA

4. Contexte

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

Chantier, atelier, service ▶ *Développement Web et Mobile*

Période d'exercice ▶ Du : 2023 au 2023

5. Informations complémentaires (facultatif)

Activité-type 2

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

Exemple n°1 - Développer des composants dans le langage d'une base de données

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

Développer les composants dans le langage d'une base de données de mon projet backend implique le fait d'avoir développé des composants qui permettent de lire, écrire, mettre à jour et supprimer des données dans notre base de données MongoDB.

Voici quelques exemples pour notre application mobile, des fonctionnalités nécessaires qui ont été créées pour stocker et récupérer les données du chat de la base de données.

Création d'un message dans la base de données, Models :

```
const mongoose = require('mongoose');

const Messages = mongoose.Schema({
  idUser: {
    type: mongoose.Schema.Types.ObjectId,
    ref: "users"
  },
  idReceiver: {
    type: mongoose.Schema.Types.ObjectId,
    ref: "users"
  },
  message: {type: String, require: true, trim: true },
}, { timestamps: true });

module.exports = mongoose.model('messages', Messages);
```

Création d'un message dans la base de données, Controllers :

```
// Envoyer un message
exports.sendMessage = (req, res, next) => {
  const message = new Messages({
    idUser: req.auth.userId,
    message: req.body.message
  });
  message.save()
    .then(() => res.status(201).json({ message: 'Message envoyé !' }))
    .catch(
      (error) => {
        res.status(400).json({
          error: error
        });
      }
    );
};

};
```

Création d'un message dans la base de données, Routes:

```
// Route pour envoyer les messages dans le chat de groupe
router.post('/messages', auth, ctrlMessages.sendMessage);
// Route pour afficher les messages
```

2. Précisez les moyens utilisés :

MongoDB, Mongoose, React, React Native, Node, Express, Visual Studio Code

3. Avec qui avez-vous travaillé ?

Meriem BARKA

4. Contexte

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

Chantier, atelier, service ▶ *Développement Web et Mobile*

Période d'exercice ▶ Du : 2023 au 2023

5. Informations complémentaires (facultatif)

Activité-type 3

Exemple n° 1

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

La création de **HairSolutions** s'est faite avec Meriem Barka. Nous devions au début être 3 développeurs puis au final, le projet s'est fait à deux. Nous avons dû nous organiser afin de nous répartir les tâches du mieux que possible.

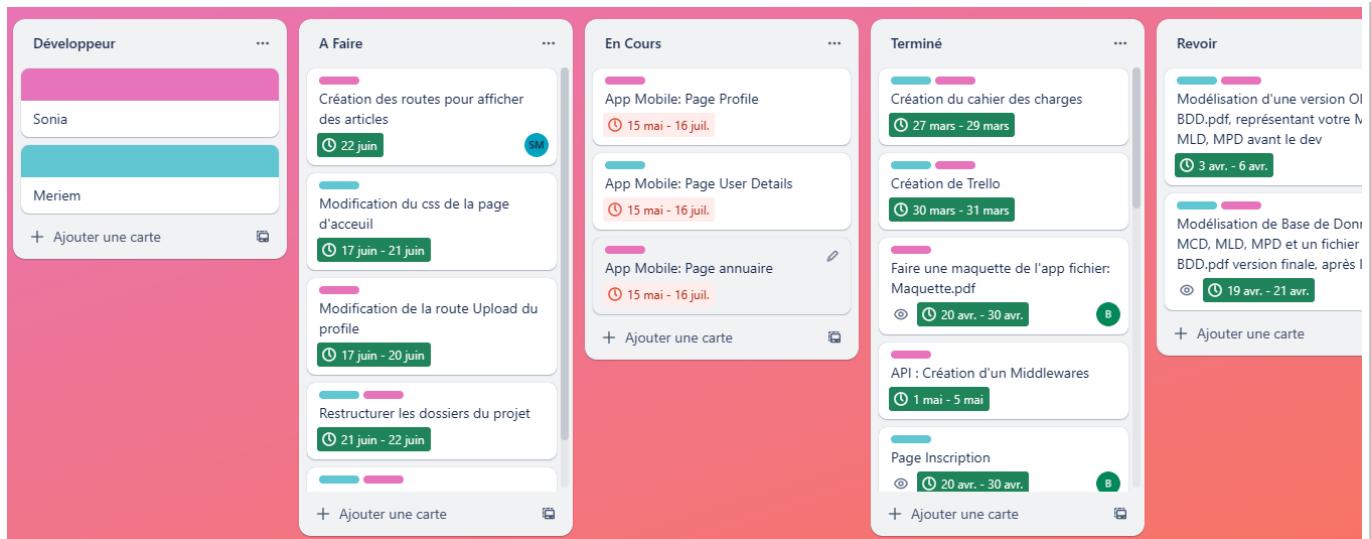
Avant la création du projet, nous nous sommes mis d'accord sur un choix de thème. Pour ce faire, nous avons utilisé la technique du brainstorming afin de partager nos envies et idées et se mettre d'accord sur quel genre d'application nous allions travailler.

Après avoir choisi le thème de notre application, nous avons énuméré les fonctionnalités principales de notre application, ainsi que les fonctionnalités secondaires si nous avions du temps, en tenant compte de nos emplois du temps respectifs.

Nous avons d'abord dû maîtriser individuellement les nouvelles technologies que nous découvrions, puis répartir les tâches, étant donné que nous n'étions que deux personnes. Afin de progresser chacune de notre côté dans les aspects frontend et backend de la création d'une application, nous avons d'abord pris le temps d'acquérir une expertise individuelle sur ces technologies à travers des projets personnels. Ensuite, nous avons travaillé ensemble sur l'application d'échanges d'astuces capillaires.

Nous avons donc pu échanger sur nos apprentissages et sur les difficultés rencontrées, ce qui nous a permis de collaborer efficacement et d'avancer dans la même direction.

Pour nous répartir les tâches de travail, nous avons utilisé **Trello** qui nous a permis de nous répartir le travail. Nous l'avons utilisé comme plateforme de gestion de projet. Cela nous a permis d'accéder rapidement aux tâches terminées, à faire, ou encore voir, ce qui a été assigné à l'autre. L'outil étant en ligne, on peut y accéder à tout moment et simultanément. Nous avons donc utilisé la version gratuite pour notre projet école.



A l'aide de **Google Docs**, nous avons créé un cahier des charges afin d'encadrer notre projet. Cela nous a permis de lister toutes les fonctionnalités que recouvriraient notre application, ainsi que l'environnement technique de notre projet, ainsi que les outils à utiliser. Cela nous a permis de jouer le jeu de créer un livrable pour un client fictif.

Nous avons aussi créé une documentation liée à l'API de notre backend avec **Google Docs**, afin de faciliter la compréhension et l'utilisation de notre **API**.

SCREEN

Nous avons utilisé **Figma** pour créer la charte graphique, les maquettes ainsi que les éléments graphiques pour la création de notre projet HairSolutions (=> application + site admin). Cela nous permettait de travailler sur le même document.

2. Précisez les moyens utilisés :

Trello, Google Docs, Figma, Github, GIT, Visual Studio Code,

3. Avec qui avez-vous travaillé ?

Meriem Barka

4. Contexte

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

Chantier, atelier, service ➤ **Développement Web et Mobile**

Période d'exercice ➤ Du : **Décembre 2022** au : **Juillet 2023**

5. Informations complémentaires (*facultatif*)

Activité-type 3

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

Exemple n°1 ▶ Développer des composants métiers

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

Pour le développement des composants métiers de notre application mobile HairSolutions, nous avons créés des composants permettant :

- **La gestion des utilisateurs et authentification** : Ces composants permettent aux utilisateurs de créer un compte, de se connecter. L'authentification est mise en œuvre en utilisant JSON Web Tokens (JWT) pour garantir la sécurité des échanges.
- **Le chat en temps réel** : Ce composant permet aux utilisateurs de communiquer en tant réel.
- **La gestion des astuces capillaires** : Ce composant permet aux utilisateurs de publier, consulter les astuces capillaires. Les astuces sont stockées dans notre base de données MongoDB et récupérées via notre API.
- **L'architecture technique** : L'architecture technique de l'application suit le modèle client-serveur. Le serveur est développé avec Node.js et Express.js, utilisant Socket.io pour la communication en temps réel. La base de données MongoDB est utilisée pour stocker les données, tandis que le frontend est développé avec React Node.js pour offrir une interface utilisateur réactive et conviviale.

2. Précisez les moyens utilisés :

MongoDB, Mongoose, React Native, Node, Express, Visual Studio Code

3. Avec qui avez-vous travaillé ?

Meriem BARKA

4. Contexte

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

Chantier, atelier, service ▶ *Développement Web et Mobile*

Période d'exercice ▶ Du : 2023 au 2023

5. Informations complémentaires (facultatif)

Activité-type 3

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

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

HairSolutions utilise la pile technologique MERN (MongoDB, Express.js, React Native.js, Node.js). Nous avons une partie backend qui est réutilisé pour le site d'administration du projet et une partie frontend.

Le backend du projet comporte :

- **Dossier models** : contient les fichiers de modèles de données de l'application
- **Dossier controllers** : contient la logique métier de chaque route. Les controllers reçoivent les requêtes HTTP, interagissent avec les services appropriés et renvoient les réponses correspondantes.
- **Dossier middleware** : contient les fichiers des middlewares, tels que l'authentification, l'autorisation, etc.
- **Dossier asset** : contient le stockage des images et documents téléchargeables.
- **Dossier config** : utilisé pour stocker la configuration de la base de données ainsi que les variables d'environnement.
- **Dossier routes** : contient les routes créées pour le projet
- **app.js** : Fichier principal où est configuré Express.js, et où sont définis les routes et middlewares
- **server.js** : fichier qui contient le lancement du serveur avec **Express.js** en utilisant le module HTTP de **Node.js**. et qui intègre les Socket.io pour la gestion de la communication en temps réel (messagerie instantanée de l'application)

Le frontend du projet comporte :

- **Dossier asset** : contient le stockage des images et documents téléchargeables.
- **Dossier config** : utilisé pour stocker la constante d'appel d'URL de l'API afin de pouvoir la réutiliser avec Axios pour la récupération ou l'envoi de données.
- **Dossier src** : où se trouvent =>
 - **Dossier components** : contient les composants réutilisables de l'application.
 - **Dossier contextes** : contient les fichiers qui gèrent l'état de connexion, déconnexion, la réinitialisation des états avant chaque tentative de connexion...
 - **Dossier routes** : contient le router de l'application.
 - **Dossier screens** : contient les pages de l'application.
 - **App.js** : le point d'entrée principal de l'application. Il contient le composant racine de l'application.

2. Précisez les moyens utilisés :

Recherche internet : <https://www.netsolutions.com/insights/mobile-app-architecture-guide/>
React, react native, node, Express, Mongo

3. Avec qui avez-vous travaillé ?

Meriem BARKA

4. Contexte

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

Chantier, atelier, service ▶ *Développement Web et Mobile*

Période d'exercice ▶ Du : 2023 au 2023

5. Informations complémentaires (facultatif)

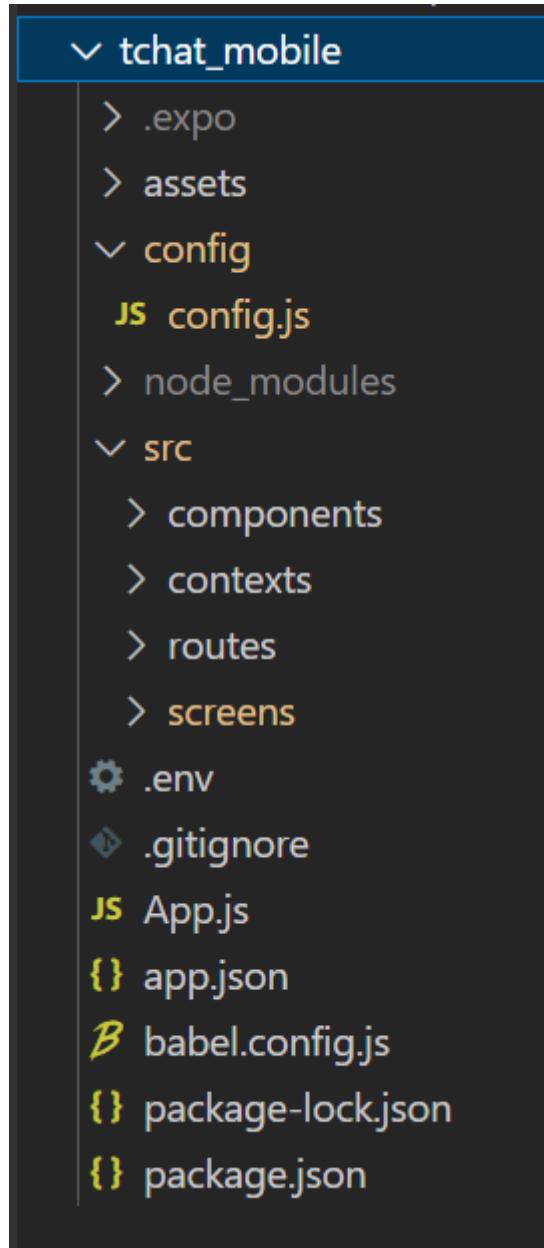
Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

Activité-type 3

Exemple n°1 - Développer une application mobile

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

Notre application mobile côté front a donc été conçue en react native. Vous trouverez ci-dessous quelques parties significatives de l'application.



Composant TabNavigator qui gère la navigation de base de l'application.

```
<SafeAreaView style={styles.container}>
    <Tab.Navigator>
```

```

initialRouteName="Home"
activeColor="#cb8a4a"
inactiveColor="#95a5a6"
barStyle={{ backgroundColor: '#fffff' }}
screenOptions={{

  tabBarActiveTintColor: '#00aea2',
  // // pour cacher le header
  headerShown: false
} }

>

<Tab.Screen
  name="Home"
  component={HomeScreen}
  options={{

    tabBarLabel: 'Accueil',
    tabBarIcon: ({ color, size }) => (
      <MaterialCommunityIcons name="home" color="#BC9E83" size={24} />
    ),
  } }
/>

<Tab.Screen
  name="ChatRoom"
  children={() => <ChatRoomScreen email={email} />}
  options={{

    tabBarLabel: "Chat",
    tabBarIcon: ({ color, size }) => (
      <MaterialCommunityIcons
name="chat-processing-outline" size={24} color="#BC9E83" />
    ),
    tabBarBadge: null,
  } }
/>

```

```

<Tab.Screen
  name="Users"
  component={UsersScreen}
  options={ {
    tabBarLabel: 'Annuaire',
    tabBarIcon: ({ color, size }) => (
      <MaterialCommunityIcons name="home" color="#BC9E83" size={24} />
    ),
  } }
/>

</Tab.Navigator>

</SafeAreaView>

```

Router :

```

const Routes = () => {

  const Stack = createNativeStackNavigator();

  const { isLogging } = useContext(AuthContext);

  return (
    <NavigationContainer>
      {isLogging ? (
        <Stack.Navigator
          initialRouteName="TabNavigator"
          screenOptions={ { headerShown: false } }
        >
          <Stack.Screen name="TabNavigator" component={TabNavigator} />
          <Stack.Screen name="ChatPrivate" component={ChatPrivateScreen} />
          <Stack.Screen name="ChatGroup" component={ChatGroupScreen} />
          <Stack.Screen name="Profile" component={ProfileScreen} />
          <Stack.Screen name="PersonalInfo" component={PersonalInfoScreen} />
          <Stack.Screen name="Password" component={PasswordScreen} />
          <Stack.Screen name="AllTips" component={AllTipsScreen} />
        
```

```

<Stack.Screen name="TipsDetails" component={TipsDetailsScreen} />
<Stack.Screen name="Category" component={CategoryScreen} />
<Stack.Screen
  name="UsersChatGroup"
  component={UsersChatGroupScreen}
/>
<Stack.Screen name="ListChatGroup" component={ListChatGroupScreen} />
<Stack.Screen name="LegalNotice" component={LegalNoticeScreen} />
</Stack.Navigator>
) : (
<Stack.Navigator
  initialRouteName="Login"
  screenOptions={{ headerShown: false }}
>
<Stack.Screen name="Login" component={LoginScreen} />
<Stack.Screen name="Register" component={RegisterScreen} />
</Stack.Navigator>
) {}
</NavigationContainer>
);
}

```

Exemple partie de Page :

```

const { userToken, userIdDecoded } = useContext(AuthContext);

const getUsers = async () => {
  try {
    const response = await axios.get(`${API_URL}/users`, {
      headers: {
        Authorization: `Bearer ${userToken}`,
      },
    });
  }
}

```

```

    console.log("response", response.data);

    const newAllUsers = response.data
      .filter((user) => user._id !== userIdDecoded)
      .map((user) => ({
        ...user,
        isConnected: !!user.token, // Vérifier si l'utilisateur a un token valide
        unreadMessages: 0, // Stocker le nombre de messages non lus
      }));
    setAllUsers(newAllUsers);
    allUsersRef.current = newAllUsers;
  } catch (error) {
    console.log(error);
  }
};

}

```

AuthContext :

```

<AuthContext.Provider
  value={{

    authLogin,
    resetAuthState,
    isLoging,
    authLogout,
    userExist,
    userDontExist,
    userToken,
    userIdDecoded,
    userTokenDecoded,
    userInfosConnected,
    getInfosUser

  }}>

```

```
{children}  
</AuthContext.Provider>
```

2. Précisez les moyens utilisés :

React native,

3. Avec qui avez-vous travaillé ?

Meriem BARKA

4. Contexte

Nom de l'entreprise, organisme ou association ➔ *Ecole La Plateforme*

Chantier, atelier, service ➔ *Développement Web et Mobile*

Période d'exercice ➔ Du : 2023 au 2023

5. Informations complémentaires (facultatif)

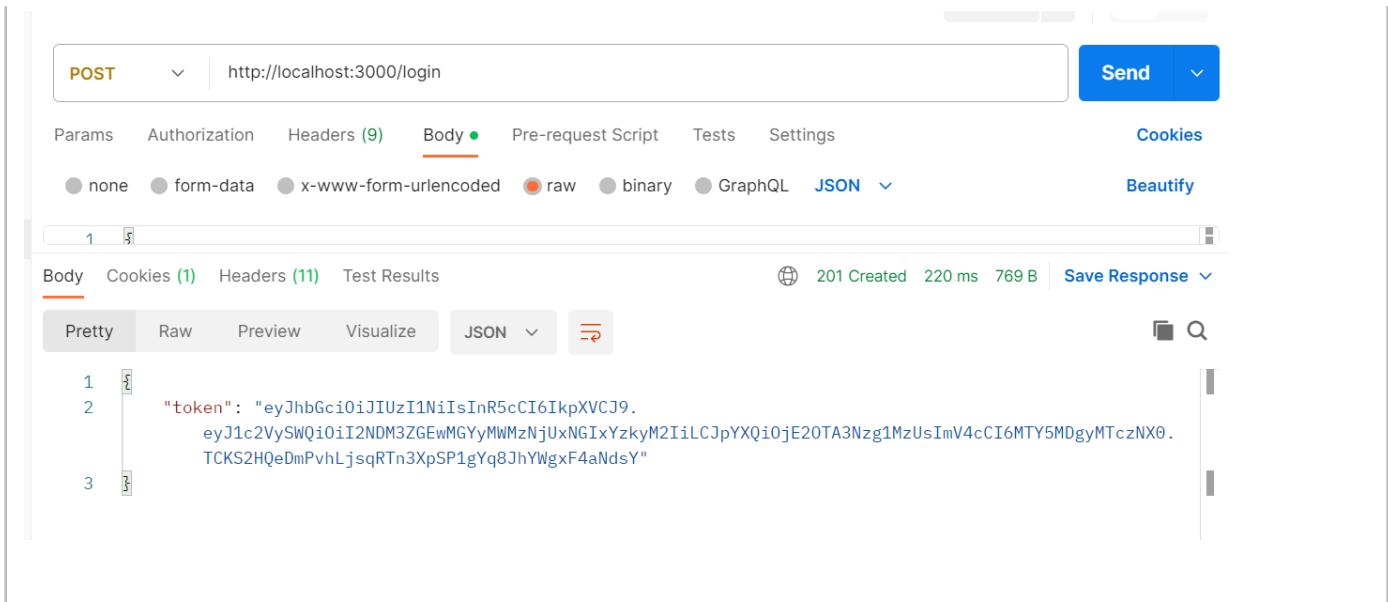
Activité-type 1 *Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité*

Exemple n°1 ➔ Développer la partie back-end d'une interface utilisateur web - Site Admin HairSolutions

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

Afin de tester le bon fonctionnement des routes et de façon simple et efficace, nous avons utilisé **Postman**.

Postman est un outil populaire et puissant pour tester, développer et documenter des API (Interfaces de programmation d'application). Il s'agit d'une plateforme de collaboration pour les développeurs qui facilite l'interaction avec les API en permettant d'envoyer des requêtes HTTP personnalisées, d'afficher les réponses et de gérer les collections d'API.



POST http://localhost:3000/login

Params Authorization Headers (9) **Body** • Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL **JSON** Beautify

1 2 3

Body Cookies (1) Headers (11) Test Results 201 Created 220 ms 769 B

Pretty Raw Preview Visualize **JSON**

1

2

3

2. Précisez les moyens utilisés :

Postman

3. Avec qui avez-vous travaillé ?

Meriem BARKA

4. Contexte

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

Chantier, atelier, service ▶ *Développement Web*

Période d'exercice ▶ Du : 2023 au 2023

5. Informations complémentaires (facultatif)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Déclaration sur l'honneur

Je soussigné(e) SONIA MOUHOUBI

,

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 AUBAGNE

le 11/07/2023

pour faire valoir ce que de droit.

Signature :

SONIA Mouhoubi

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé

Cliquez ici pour taper du texte.

ANNEXES

(Si le RC le prévoit)