

Dans le cadre de votre formation, vous allez concevoir et développer une application web complète en utilisant **React** pour le frontend et **Express** pour le backend.

Le but de ce projet est de mettre en œuvre les concepts appris en cours, notamment la création d'interfaces utilisateur dynamiques avec React, la conception d'une **API REST** avec Node.js - Express et l'intégration d'une base de données pour la gestion des données.

Vous avez la liberté de choisir le sujet de votre application.

Que ce soit une **plateforme de gestion**, une **application de réseau social**, une **marketplace**, ou tout autre concept, le choix du thème vous appartient. L'important est de respecter les fonctionnalités obligatoires et les technologies imposées.

OBJECTIFS DU PROJET:

En petits groupes, vous allez concevoir une application composée de deux parties distinctes :

Frontend : Développé avec **React**, le frontend sera organisé en composants réutilisables. Il offrira une interface utilisateur interactive et responsive, avec des mises à jour en temps réel sans rechargement de la page.

Backend : Construit avec **Express**, le backend fournira une **API REST** pour permettre la communication entre le frontend et le backend, facilitant l'échange de données.

FONCTIONNALITÉS OBLIGATOIRES :

1. Authentification complète avec JWT :

- Mise en place d'un système d'authentification avec gestion de tokens JWT pour sécuriser les routes de l'API.
- Les utilisateurs doivent pouvoir s'inscrire, se connecter, se déconnecter et réinitialiser leur mot de passe via un e-mail sécurisé.

2. Gestion des données via CRUD :

- Implémentez un système de gestion des données qui permet d'ajouter, de consulter, de modifier et de supprimer des éléments dans une base de données.

3. Affichage dynamique des données (React + API REST) :

- Les données récupérées via l'API REST doivent être affichées dynamiquement dans l'interface React.

4. Filtres et recherche multicritères :

- Intégrez des filtres et une recherche multicritères permettant à l'utilisateur de trier et de rechercher les données affichées en fonction de différents critères (ex : par date, nom, catégorie, etc.).

5. Validation des champs de formulaires et gestion des erreurs :

- Tous les formulaires doivent inclure une validation des champs obligatoire et afficher des messages d'erreur pertinents en cas de données invalides ou manquantes.

6. Téléchargement et gestion de fichiers :

- L'utilisateur doit pouvoir uploader des fichiers (cela peut être simplement une photo de profil par exemple). Assurez-vous que des contrôles de format et de taille de fichier sont en place, et que les fichiers sont gérés correctement côté serveur.

7. Système de rôles et permissions avec JWT :

- Implémentez un système de rôles (ex : utilisateur, administrateur) avec des **permissions différentes** selon le rôle attribué.

TECHNOLOGIE IMPOSEES :

- **Frontend** : ReactJS
- **Backend** : Node.js avec Express
- **API REST** : L'API doit être conforme aux standards REST pour la communication entre le frontend et le backend.
- **Base de données** : PostgreSQL ou MongoDB.
- **Gestion des sessions avec JWT** : Utilisez des **JSON Web Tokens** pour sécuriser les sessions utilisateur et l'authentification.
- **Git** : Utilisation obligatoire pour la gestion du code source et de la collaboration.
- **Hébergement** : L'application devra être hébergée sur un serveur accessible en ligne.

A venir, fourni par CCI Campus, pour l'instant travail en local.

Dans votre développement, tenir compte de la législation en vigueur internet EU (site web CNIL et RGPD)

ATTENTES MÉTHODOLOGIQUES

- Qualité du code, optimisation et commentaires en anglais
- Gestion de projet avec la méthodologie Agile / scrum
- Utilisation d'outils collaboratifs de projet (GIT, Trello ou autre)
- Faire une bonne analyse / conception du projet avant de se lancer dans le code !
- Liste des tests unitaires prévus pour le projet
- Rendus documentaires au format PDF présentés de manière professionnelle (efforts de mise en forme, mise en page, en-tête et pied-de-page, numérotation, plan, table des matières)

SOFT SKILLS :

- Ecoute des co-équipiers, cohésion d'équipe, prise en compte des avis de chacun, gestion conflits
- Gestion du temps et de l'avancée du projet (temps équipe, temps individuel), points réguliers
- Partage du travail, mettre chacun à contribution du mieux possible
- Communiquer, retour vers les autres membres de l'équipe et les formateurs
- Présenter le travail réalisé, faire une démonstration de l'application en groupe

FORMATEURS SUIVANT CE PROJET :

- **François SCHOSSIG** pour la conception / analyse
- **Sylvain GIBAUD** maquettage interface + UI/UX
- **Sofiane AL AMRI** codage React JS – node JS, GIT, optimisation et sécurisation
- **Duong Khang NGUYEN** pour la gestion de projet Agile/Scrum
- **Roger VARNIER** pour le déploiement DevOps + notions systèmes réseaux
- **Philippe SCHLEGEL** méthodologie de tests, tests unitaires

En fonction de l'avancée de leurs cours, vos formateurs peuvent réserver une heure en fin de journée de cours pour vous assister, répondre à vos questions.

Les formateurs noteront les différents livrables. Certains seront présents pour les démonstrations de fin de projet. Remontez régulièrement des informations à vos formateurs sur l'avancée de vos applications / sites (invitez les formateurs sur des plateformes collaboratives en ligne type GIT, Trello..)

LIVRABLES ET ECHEANCES :

11/12/2024 : Rendu Note de cadrage (2 pages maximum)

- Description du projet (idée générale)
- Fonctionnalités principales à développer

A envoyer en version numérique (PDF) à s.alamri@ccicampus.fr et dnguyen@ccicampus.fr et fschossig@ccicampus.fr

20/12/2024 : Audit de projet intermédiaire

Groupe par groupe, vous présentez à Sofiane AL AMRI et François SCHOSSIG votre avancée et travail réalisé

12/01/2025 : rendu V1 Dossier d'analyse contenant :

- schémas UML : diagramme de classes, uses-cases, diagramme des flux, séquences
- schéma relationnel BDD
- dictionnaire des données
- export données et structure BDD (fichiers .SQL)
- spécifications fonctionnelles et techniques

A envoyer en version numérique (PDF) à François SCHOSSIG fschossig@ccicampus.fr + s.alamri@ccicampus.fr

+ Rendu maquettage interface 3 écrans principaux à Sylvain GIBAUD s.gibaud@ccicampus.fr

14/01/2025 : Audit de projet intermédiaire

Groupe par groupe, vous présentez à Sofiane AL AMRI et François SCHOSSIG l'ensemble de votre analyse et documents de conception

26/01/2025 : rendu V2 Dossier d'analyse corrigé (si nécessaire) selon remarques formateurs du 14/01

A envoyer en version numérique (PDF) à François SCHOSSIG fschossig@ccicampus.fr + s.alamri@ccicampus.fr

29/01/2025 : Front React statique

- Le frontend de l'application avec des données simulées
- Le lien du répertoire public GIT
- Le fichier **README.md** doit fournir tous les détails afin de lancer l'application
- Spécification de l'API REST : définition des routes, méthodes HTTP

04/02/2025 : Audit de projet intermédiaire

Groupe par groupe, vous présentez à Sofiane AL AMRI et François SCHOSSIG votre avancée et travail réalisé

03/03/2025 : Audit de projet intermédiaire

Groupe par groupe, vous présentez à Sofiane AL AMRI et François SCHOSSIG votre avancée et travail réalisé

21/02/2025 au soir : Rendu du code en fin de journée + mise en ligne sur serveur d'hébergement

On ne touche plus au code après la date du 21/02/2025, même si ça ne fonctionne pas.

Concentrez-vous sur vos présentations et préparez celles-ci. L'oral doit être bien préparé.

Dépôt Git > Sofiane AL AMRI

Scénarios de tests / tests unitaires + Liste des débogages principaux effectués

- Optimisation du code : liste des principales corrections / améliorations effectuées

26/03/2025 : Démonstration finale

- Application complète fonctionnelle avec toutes les fonctionnalités obligatoires implémentées et déployé.
La plateforme reste à définir
- **Démonstration** de l'application devant le groupe et formateur (30 min + questions/réponses)
- **Tenue correcte attendue, comme pour examens**

Le projet doit couvrir toutes les compétences ci-dessous (attendus du référentiel du Titre professionnel de Concepteur Développeur d'Applications).

Le sujet / thème de l'application est libre du moment qu'il répond aux attentes énoncées dans les pages suivantes.

N° Fiche AT	Activités types	N° Fiche CP	Compétences professionnelles
1	Développer une application sécurisée	1	Installer et configurer son environnement de travail en fonction du projet
		2	Développer des interfaces utilisateur
		3	Développer des composants métier
		4	Contribuer à la gestion d'un projet informatique
2	Concevoir et développer une application sécurisée organisée en couches	5	Analyser les besoins et maquetter une application
		6	Définir l'architecture logicielle d'une application
		7	Concevoir et mettre en place une base de données relationnelle
		8	Développer des composants d'accès aux données SQL et NoSQL
3	Préparer le déploiement d'une application sécurisée	9	Préparer et exécuter les plans de tests d'une application
		10	Préparer et documenter le déploiement d'une application
		11	Contribuer à la mise en production dans une démarche DevOps