

# Projet de développement WEB

Consignes du projet

## Organisation

Individuel / groupe (max 3) : info à donner avant le 31/01/2026

Sauf exception, le projet sera développé avec les technologies suivantes :

- Frontend : Angular 20+
- Backend : Java 21+ avec SpringBoot 3.5+

L'utilisation des outils / frameworks ci-dessous est recommandée mais non obligatoire :

- npm
- maven
- Angular Material
- TailwindCSS
- OpenAPI

## Consignes

Cahier des charges : voir cours de SGBD.

## Evaluation

### Objectifs

Mettre en application les concepts et bonnes pratiques étudiés pendant les cours.

L'évaluation est scindée en deux parties : la réalisation du projet (code) et l'examen oral.

### Projet

#### Requis :

- Le dossier d'architecture est présent ; l'architecture décrite dans le dossier est respectée dans le code.
- Le document d'exploitation est présent
- Les différentes parties du projet démarrent sans erreur
- Un mécanisme permet l'authentification d'utilisateurs ; différents rôles utilisateurs permettent de réaliser des opérations spécifiques à ces rôles
- Un frontend communique avec un backend via une API HTTP

#### Bonus :

- L'application est responsive et permet une utilisation correcte sur smartphones et tablettes
- L'interface utilisateur est agréable et simple d'utilisation
- L'application est performante
- Les erreurs sont gérées correctement (message informatif, pas de plantage brut...)

- Des tests automatisés et pertinents passent avec succès

## Examen

La date de l'examen oral est fixée au 15/06/2026.

Les modalités pratiques (horaire de passage) sont transmises à l'avance.

## Livraison

Un email est envoyé à l'adresse [r.hardenne@ephec.be](mailto:r.hardenne@ephec.be) avec l'url du dépôt git.

### Code

Le code est disponible dans un dépôt git public sur la plateforme du choix de l'étudiant.

Pour plus de facilité, **un seul dépôt contient tout le code du projet**, frontend et backend.

### Dossier d'architecture

Un dossier d'architecture est rédigé et est disponible à **la racine du dépôt git**.

Le dossier d'architecture décrit, de manière succincte :

- L'architecture frontend (couches...)
- L'architecture backend (couches...)
- La liste des librairies, outils et frameworks structurants du projet
- L'url vers le « swagger » permettant de visualiser l'API http (lorsque l'application est démarrée, pas besoin d'avoir une url permanente)

### Document d'exploitation

Un document décrivant les étapes permettant de compiler et exécuter le projet est disponible à **la racine du dépôt git**. Le but de ce document est de permettre un démarrage rapide du projet et de le tester.

### Frontend

- La commande pour initialiser / compiler le projet si nécessaire (npm install...)
- La commande / le script pour démarrer le projet
- Le port si différent de celui par défaut
- Toute autre information utile et nécessaire pour démarrer/utiliser le frontend
- La commande / le script permettant de lancer les tests
- ...

### Backend

- La commande pour compiler le projet (mvn clean install...)
- La commande / le script pour démarrer le projet
- Le port si différent de celui par défaut
- Toute autre information utile et nécessaire pour démarrer/utiliser le backend
- La commande / le script permettant de lancer les tests
- ...

## Base de données (optionnel pour le cours de PDW)

Si le projet utilise une base de données, il conviendra de fournir toutes les informations nécessaires pour la démarrer et l'initialiser (seeding – création du schéma + insertion des données de test). Il est important que ces étapes **soient simples et automatisées**.

Deux options :

- Utilisation de docker : la ligne de commande qui démarre le container est fournie ; le seeding est réalisé une fois au premier démarrage.
- Utilisation d'une base de données « in memory » (h2...), démarrée et seed au lancement du backend.

L'url de connexion ainsi que les credentials (login + password) sont également disponibles dans ce document.

A noter :

- Aucun logiciel ne devra être installé à part docker
- Aucun script SQL ne sera exécuté