

ECF DWWM n°2/2

FÉVRIER 2025



L'Afpa a besoin d'une application en ligne pour gérer les absences des stagiaires sur La Teste.

Le formateur souhaite pouvoir :

- ajouter / modifier / supprimer un stagiaire avec ses infos usuelles et avec sa photo d'identité
- avoir un mur d'accueil "mosaïque trombinoscope" pour cliquer sur le stagiaire absent et saisir les détails de l'absence.
- ajouter / modifier / supprimer une absence sur un stagiaire.
- discrètement, par exemple dans une sidebar masquable (type offcanvas), obtenir quelques statistiques par exemple : classement stagiaire par ordre d'absence, ou perte de revenu estimé cumulé (dans le cas où un stagiaire reçoit 712€/mois sur 21 jours ouvrés)
- afficher partout en rouge dans l'interface d'admin le nom d'un stagiaire qui a eu plus de 5 absences "sans motif" dans son histoire.
- les parties CRUD doivent être réservées au seul formateur "admin" mais les statistiques peuvent être publiques pour la consultations des stagiaires
- quand on ajoute une absence en plus de la date, on doit aussi saisir la raison de l'absence au choix
 - **maladie**
 - **sans motif**, (*"sans motif" nous sert aussi de fourre tout, pas besoin d'une valeur "autres/divers"*)
 - **absence légale** (ça regroupe mariage / décès d'un proche / garde d'enfant),
 - **accident du travail**.
- on doit pouvoir avoir l'option de téléverser le PDF du justificatif (d'une ordonnance médical, d'une convocation tribunal, d'un avis de décès, mariage, etc)

Pour simplifier ce projet sur le temps court de l'ECF :

- Nous allons considérer uniquement ces [11 stagiaires](#) à entrer dans le jeu de données.

- Nous allons considérer aussi “par défaut” comme toujours présents sur les jours ouvrables **et donc uniquement “saisir” en BDD les absences à la journée.**
- pas de gestion des retards ou des demi-journées, **restons simple sur la situation d’une journée entière.**
- **pas de création de compte admin**, intégrer 1 seul admin en bdd :
(login : **Raphaël** / mdp : **DWWM@LATEST**)

Bien sûr on attend de vous les bonnes pratiques usuelles : documenter son code, structurer en MVC, bien classer le public et les assets, utiliser l’anglais dans les noms des fonctions, les BDD, les noms de fichiers, les commentaires.

Déroulement de l’ECF :

Le rendu est individuel, vous avez l’autorisation d’accès à toutes les ressources à votre disposition sur internet, vos notes, vos codes de précédents projets.

IMPORTANT : Dans un esprit d’agence et de réalisme du monde du travail, j’autorise aussi “**l’entraide et les discussions entre stagiaires**” durant toute la durée de l’ECF.

Par exemple : débattre en réunion après lecture individuelle du brief, pour valider si la compréhension est partagée par tous, éventuellement éclaircir les zones d’ombre.

Par exemple bis : comparer les MCD en réunion avant d’attaquer individuellement le PHP

En revanche, il faudra respecter ce cadre :

- **que ça reste très occasionnel**, pour ne pas briser la concentrations des autres
- **consenti**, savoir respecter le besoin d’isolement OU simplement le choix qu’une personne ne souhaite pas participer à ce concept d’entraide durant l’ECF.
- **subtilement** : ne solliciter vos pairs que lorsqu’on observe dans leur attitude qu’ils sont disponibles pour échanger. (encore une fois ne pas briser concentration d’un codeur en PLS)

PARTIE 1 : Mise en place et manipulation des bases de données

pour valider votre compétence : Mettre en place une base de données relationnelle (la compétence 5 si vous voulez relire le REAC !)

Activité 1 : Analyse des besoins et création du schéma physique

- Fournir un dossier de conception incluant un diagramme des utilisations du site schématisant vos réflexions + modèle entité-association. MCD
- créer le schéma physique des données en respectant les règles de nommage et de relationnalité. MLD

Activité 2 : Création et configuration de la base de données

- Utilisation d'un SGBD
- Implémentation des tables selon le schéma physique et création des utilisateurs avec des droits spécifiques.

Activité 3 : Tests et sauvegarde

- Remplissage de la base avec un jeu d'essai ([les 11 stagiaires](#))
- sauvegarder export sql de votre table et du jeu d'essai.

Critères de performance issus du REAC :

- Les données du schéma conceptuel et leurs relations sont identifiées et prises en compte
- Le schéma physique est conforme aux besoins exprimés dans le dossier de conception et respecte les
- règles des bases de données relationnelles
- Les règles de nommage sont respectées
- La sécurité, l'intégrité et la confidentialité des données est assurée
- La base de données de tests mise en place est conforme au schéma physique
- Les utilisateurs sont créés avec leurs droits respectifs conformément au dossier de conception
- Un jeu d'essai complet est créé
- La base de données créée est sauvegardée et elle peut être restaurée en cas d'incident

- La documentation technique des bases de données est comprise, en langue française ou anglaise
-

PARTIE 2 : Développement back-end

Compétences du REAC :

- Développer des composants d'accès aux données SQL et NoSQL (votre compétence #6)
- Développer des composants métier côté serveur (#7)

Activité 1 : Développement des accès aux données

- Implémentation de requêtes SQL (CRUD) sécurisées.
- **OPTIONNEL** : Implémentation de requêtes NoSQL (par exemple en écrivant les logs des actions de l'admin dans un fichier json, et en appelant ces Logs dans une page de statistique exploitant ce json)

Activité 2 : Développement de composants métier côté serveur

- Développement d'une fonctionnalité métier basée sur un besoin donné (*le générateur de rapport statistiques à partir des données ou le gestionnaire de fichier (upload des pdf) valident cette compétence*).
 - Validations du code via des outils de qualité de code. (avec PHP Intelephense par ex.)
 - **OPTIONNEL** : Validation via des tests unitaires (avec PHPUnit via composer par ex.)
-

PARTIE 3 : Documentation, déploiement automatisé

Compétences du REAC :

- Documenter le déploiement d'une application dynamique web ou web mobile (#8)

Activité 3 : Documentation et déploiement

- Rédaction d'un guide PDF de déploiement expliquant les étapes pour mettre en ligne le projet + comment utiliser cette application (mode d'emploi) + un screen des validateurs de qualité de code
- faire un script PHP d'automatisation de l'installation (dézipper + injecter le sql dans la bdd puis détruire le fichier install.php + le zip + le sql)

3 LIVRABLES ATTENDUS sur DRIVE dans un ZIP nominatif :

- **1 PDF** : Schémas des interactions du site + les modèles MERISE MCD + MLD
- **1 PDF** : guide de déploiement appli / mode d'emploi appli
- **1 ZIP** : contenant l'auto-installeur "install.php" + zip du site + sql de la bdd (je vais installer vos projets sur mon wamp pour vous noter (important : préconfigurer BDD avec login "root" et pas de mot de passe (vide) et la dbname sera "ecf2_votreprenomici")

Critères de performance made in REAC partie 2 & 3 :

- Les traitements relatifs aux manipulations des données répondent aux fonctionnalités décrites dans le dossier de conception
 - L'intégrité et la confidentialité des données sont maintenues
 - Les cas d'exception sont pris en compte
 - Les conflits d'accès aux données sont gérés
 - Toutes les entrées sont contrôlées et validées dans les composants serveurs sécurisés
 - Les tests unitaires et de sécurité sont associés à chaque composant (*OPTIONNEL sur cet ECF*)
 - La démarche structurée de résolution de problème est adaptée en cas de dysfonctionnement
 - Le système de veille permet de suivre les évolutions technologiques et les problématiques de sécurité liées aux bases de données SQL et NoSQL
-
- Les traitements répondent aux fonctionnalités décrites dans le dossier de conception
 - Les composants métier sont sécurisés
 - Les bonnes pratiques de la programmation orientée objet (POO) sont respectées

- Les règles de nommage sont conformes aux normes de qualité de l'entreprise
- Le code source est documenté, y compris en anglais
- La qualité du code est vérifiée, éventuellement à l'aide d'un utilitaire de contrôle de qualité de code
- Un jeu d'essai fonctionnel et les tests unitaires ont été réalisés pour les composants concernés
- Les tests de sécurité sont réalisés
- La démarche structurée de résolution de problème est adaptée en cas de dysfonctionnement

- La procédure de déploiement est rédigée ou mise à jour
- Les scripts de déploiement sont écrits et documentés
- Le système de veille permet de suivre les évolutions technologiques et les problématiques de sécurité
- liées au déploiement d'une application dynamique web ou web mobile, y compris dans le cadre d'une
- démarche DevOps