WE4A

LOMBARD Alexandre
LITZLER Antoine
KAS Mohammed

WE4A

Cahier des Charges du Site

« Moodle Simplifié »

Le but de ce document est de préciser les demandes du sujet, de la perspective de WE4A. Même si le Moodle original reste un excellent référentiel pour ce que nous vous demandons, les différences sont aussi considérables, et pour répondre à votre demande pour une clarification, ce document a été produit.

Ce cahier des charges » a été rédigé de la perspective de WE4A. Il ne va donc vous décrire que les features demandées au niveau du site, sans se préoccuper de la qualité de la BDD qui le fournit en données.

Table des matières

I - Technologies à utiliser

Le site doit **absolument** **utiliser le framework Symfony 7.**
Pour que ce dernier fonctionne, il vous faut des serveurs Apache et MySQL, et la manière la plus simple de les avoir sur sa machine est d'installer XAMPP ou un équivalent.
Lors des corrections, le dossier de votre site sera placé dans une machine avec XAMPP et Symfony. Les « migrations » de BDD seront faites à partir de vos entités Symfony, des données exemples importées à partir d'un fichier SQL généré via PHPmyAdmin, et le site sera alors censé **fonctionner immédiatement sans avoir besoin d'installer autre chose.**

ATTENTION : AUCUNE AUTRE LIBRAIRIE, PACKAGE OU PLUGIN N'EST AUTORISE (à l'exception des 2 mentionnés si dessous). Si vous désobéissez à cette règle, votre note sera pénalisée.

En effet, si les étudiants ajoutent des librairies et technos de leurs choix, la correction de ces projets, déjà très longue et complexe, devient juste ingérable de la perspective des enseignants, car cela va **interférer avec nos barèmes** et critères de notation.
Je mentionnais des exceptions. Ce sont deux outils/librairies communs dans le web et qui ont l'avantage de pouvoir être importés par un lien global et/ou placées dans le dossier du site.

Nous n'aurons pas le temps de les étudier en TD/TP, mais vous avez le droit d'utiliser si vous les connaissez :

- **Bootstrap.** Pour aider à concevoir le HTML/visuel de la page
- **JQuery,** pour aider sur les Javascript côté client

J'insiste que ce sont bien ces deux là et seulement ces deux là. Il existe des dizaines d'alternatives moins connues à Bootstrap et JQuery, qui, elles aussi, peuvent être importées par un lien et jouent des rôles similaires.
Mais le critère pour refuser les autres librairies n'est pas d'éventuels problèmes d'installation, mais bien le BARÈME, qui ne peut pas tenir en compte tous les cas possibles de librairies et d'outils, et doit donc vous restreindre sur les librairies facultatives que vous avez le droit d'utiliser.

Mais avec Bootstrap et JQuery, on peut quasi littéralement tout faire, alors...

II – Rôles devant exister sur le Site :

II-A) Administrateur

- **Ce rôle permet, après login, d'accéder à une zone spécifique du back-end, réservée aux seuls administrateurs.**
  Dans ce back-end « admin », il est possible de créer les UE, et d'y assigner un ou plusieurs professeurs et étudiants.
- **L'administrateur a aussi le pouvoir de créer des comptes utilisateur** avec un mot de passe temporaire/par défaut, et de leur assigner les rôles de son choix (étudiant, professeur, ou même admin)
- **Permettre à n'importe qui de créer son compte via formulaire ne fait pas vraiment sens sur le site d'une université/école.**
  C'est l'administration qui s'en occupe. Il faudra juste créer « artificiellement » un compte admin dans la BDD afin de démarrer la chaîne de création de comptes.
- **Il a des professeurs qui sont aussi administrateurs**, mais il y a des administrateurs qui ont seulement ce rôle et ne sont pas professeurs (secrétariat, service de gestion de emplois du temps, etc.)
- **Si un prof qui est aussi admin se login,** il peut, au choix, arriver sur la zone d'administration, ou bien sur la page d'accueil normale des professeurs. Ce qui compte, c'est qu'en temps qu'admin, il aie accès à des liens pour passer facilement de la « zone admin » à la « zone profs »
- **Un étudiant ne peut pas être administrateur**
- Dans le cadre des « objectifs minimum », un administrateur **n'a pas la possibilité d'accéder à l'intérieur des UE** pour modifier leurs posts. Sauf si il s'avère qu'il est aussi professeur sur cette UE.
- **Un « Nice to Have » (voir fin de document)** serait que les admins puissent modifier le contenu des UE, mais que les professeurs d'une UE modifiée reçoivent une notification qu'une action admin a eu lieu.

II-B) Etudiant :

- **Après son login**, un étudiant arrive sur la page de « choix de l'UE ». Cette page contient aussi des informations sur l'activité récente sur les UE en question (voir plus loin)
- **En cliquant une UE, un élève va pouvoir voir les posts existants dans l'UE en question.** L'étudiant, ne peut, pour l'essentiel, que consulter : il peut voir les massages textes et télécharger les fichiers déposés par les professeurs dans les posts « fichier »
- En WE4B, vous pousserez le même site plus loin et offrirez à l'étudiant des interactions plus intéressantes. Mais dans le contexte de WE4A, il a un rôle très restreint et facile à implémenter.

II-C) Professeur :

- **Ce rôle permet, après login, d'accéder au mêmes pages que les étudiants.** ...Mais avec des liens/formulaires supplémentaires qui permettront **l'édition du contenu.**
- **Après son login**, un professeur (comme un étudiant) arrive sur la page de « choix de l'UE ». Cette page contient aussi des informations sur l'activité récente (voir plus loin)
- En cliquant une UE, un professeur va pouvoir voir les posts existants dans l'UE en question. Comme les étudiants. Mais lui pourra voir **des boutons supplémentaires afin d'ajouter/modifier/effacer les posts.**
- **Selon votre ambition/niveau/temps disponible,** les formulaires d'édition de contenu pourront être ou bien sur une page séparée, ou bien révélés/ajoutés dynamiquement à la page via Javascript et AJAX.

  Conseil : commencez par une version où les formulaires sont sur des pages différentes (objectif minimum) et essayez plus tard de les transformer en versions plus dynamiques et ergonomiques

III – Listes des Pages « nécessaires »

Vous aurez probablement besoin de pages supplémentaires pour accueillir, par exemple, le formulaire de création d'un nouveau post.
Mais dans la théorie, les pages ci-après sont les seules à être strictement nécessaires. ...Pour peu que vous utilisiez beaucoup, beaucoup de Javascript afin de mettre la page à jour « en direct ».

Ces Javascript destinés à éviter les pages supplémentaires sont des « Nice to Have ». Ils ne sont pas obligatoires.

Mais il y a une certaine quantité de Javascript obligatoire, qui vous sera demandé explicitement dans la description des pages ci-après.
Ils seront présentés ainsi, dans un encadré bleu clair facile à repérer.

III-A) Page de Login *(tout le monde)*

La page d'accueil consiste essentiellement en un formulaire de login.
Elle est la seule page qui peut être accédée sans login, et si un utilisateur tente d'accéder une autre page sans s'être identifié, il sera **redirigé** ici.

Javascript demandé : le formulaire de login doit proposer l'option de montrer/cacher le mot de passe (oui, les navigateurs le font souvent automatiquement maintenant, mais on vous demande quand même). Il doit aussi refuser d'envoyer les données si jamais un champ a été laissé vide.

Nice to Have : Vous pouvez, si vous le souhaitez, rendre cette page un peu plus intéressante, par exemple avec une esthétique plus poussée ou en affichant des statistiques, par exemple sur le nombre de professeurs, d'élèves, de posts, etc, etc...

III-B) Page(s) d'administration *(admin seulement)*

III-B1) Une page ou plusieurs ?

Accessible uniquement aux administrateurs, cette ou ces pages sont celles où ce type d'utilisateur va se retrouver immédiatement après son login.
Un professeur qui est aussi un admin n'arrive pas ici, mais sur la page de sélection de l'UE après son login (voir plus loin).
Un « prof admin » a en revanche en permanence accès à un lien qui l'amène sur cette page d'administration *(par exemple dans une barre de menu en haut)*

Dans la pure théorie, il est possible de répondre aux besoins de cette section admin avec une seule page bardée de Javascript. Mais vous avez l'option d'utiliser plusieurs pages si vous le désirez *(la création/modification d'un user ou d'une UE, par exemple, peuvent être des pages séparées)*

III-B2) Le catalogue

La page qui est scrupuleusement indispensable est le « catalogue », qui permet de passer en revue tous les utilisateurs et UE enregistrés dans le système/site.
Les deux « collections », users et UEs sont affichés sur la même page, mais pas simultanément, via un système « d'onglets »

En haut de chaque collection se trouve un bouton « Créer » (donc créer user ou créer UE, selon l'onglet actuellement affiché).
Pour chaque « entrée » d'une collection *(un user individuel ou une UE individuelle, donc)*, il y a deux liens/icônes à proximité : modifier, et « effacer ».

> **Javascripts demandés**
>
> UE et users sont sur deux « onglets » de la même page « catalogue », avec la possibilité d'afficher un onglet ou l'autre via Javascript.
>
> L'effacement des UE et étudiants doit se produire via AJAX, sans changer de page. Une pop-up demandant confirmation doit également être affichée avant que l'effacement et la mise à jour de la page aient vraiment lieu.

A moins que vous n'implémentiez le « Nice to have » suggéré juste après, **la création ou la modification d'un user ou UE va avoir lieu sur une autre page, dédiée.**

Nice To Have
Lorsque l'on veut ajouter ou modifier une UE ou un User, un usage intensif et intelligent de Javascript et AJAX permet de « faire apparaître les formulaires » dans la page, et d'envoyer les modifications et mettre à jour la page, le tout sans « aller sur une page de formulaire séparé ».
Si vous implémentez ça, le site sera plus dynamique à utiliser, et c'est un +

III-B2) Création/modification d'un User

Un user contient les données suivantes : nom, prénom, email, rôle (admin, prof, admin ET prof, étudiant), mot de passe, UE assignées.
Lors des login, c'est le couple e-mail/mot de passe qui est demandé.

La page de création peut être la même que celle de modification : dans le cas d'une modification, les champs seront pré-remplis par les données pré-existantes.
A noter que, lors d'une création, le mot de passe peut être pré-rempli avec une valeur par défaut.
Un utilisateur pourra ensuite changer son mot de passe par lui même, même si il n'est pas admin. Via sa petite page de gestion de compte. L'admin conserve néanmoins le pouvoir de changer les mots de passe, au cas où *(pour les oublis, la nécessité d'accéder au compte de quelqu'un qui est parti, etc.)*

Pour « assigner des UE à l'utilisateur », **je vous demande d'utiliser un peu votre créativité et de mettre au point une interface par vous-même.**

- L'admin doit pouvoir sélectionner une UE parmi celles déjà existantes dans le système, et « l'ajouter » à la liste des UE associées à cet user.
- Une UE déjà présente dans cette liste doit pouvoir être « retirée »
- Un user administrateur n'a pas besoin de liste d'UE assignées, sauf si il est aussi un professeur

Essayez de concevoir cette petite interface de « gestion de liste » comme aussi pratique que possible. N »hésitez pas à utiliser Javascript si besoin afin d'améliorer l'ergonomie et le « confort ».

Nice To Have
Ajouter un système qui permet d'aller créer un UE dans une popup ou un autre onglet, mais sans perdre un utilisateur qu'on est en train de saisir.
Utile si jamais on se rend compte qu'une UE est manquante alors qu'on est en train de saisir un user !

III-B3) Création/modification d'une UE

Une UE, au niveau de la gestion admin, n'est pas très complexe.
Elle contient : un code (comme WE4A), un intitulé (Comme « Développement Web »), une « image pour la page de choix d'UE ».

Nous ne demandons pas la possibilité de gérer la liste des users associés depuis cette page, la rendant de fait bien plus simple que la précédente.
Néanmoins, si vous voulez un challenge…

> **Nice To Have**
> Permettre la gestion des users associés à l'UE depuis cette page. Cela a beaucoup de similitudes avec la gestion de la liste décrite pour la page précédente.
> ...Mais comme les users seront bien plus nombreux que les UE, un système adapté pour les sélectionner est nécessaire. L'idéal étant une recherche par nom, qui opérera via AJAX.

III-C) Page de choix de l'UE *(prof, étudiant)*

Cette page est **celle qui accueille un professeur où un étudiant après son login.** Un utilisateur peut y voir des liens très proéminents vers les UE auxquelles il est assigné. **Ces liens vont utiliser les images qui ont été uploadées à la création de l'UE afin d'apparaître comme une boîte bien visible** *(inspirez vous du vrai Moodle)*

En plus de ces liens « imagés » vers les UE, cette page propose également un fil « actualités », qui sera rempli de notifications sur l'activité récente ayant eu lieu sur les UE liées à l'utilisateur.
Par exemple :

- M. LITZLER a posté un nouveau fichier, « Sujet Projet », dans WE4A
- M. LOMBARD a posté un message « Changement salle » dans WE4A
- M. LITZLER a posté un message « Sources TP1 » dans LP2A
- etc, etc...

Ce fil d'actualité est dans l'ordre chronologique décroissant (du plus récent au plus ancien) et montre un nombre arbitraire de messages (10 ? 20?). Ceux qui sont trop anciens pour faire partie de cet échantillon sont juste ignorés.

Nice To Have
Introduisez un système de « pages » ou un bouton AJAX « chargez des actualités plus anciennes » afin d'offrir un moyen à l'utilisateur d'accéder aux actualités plus anciennes.

III-D) Page de contenu UE *(prof, étudiant)*

En quelque sorte la « page principale » de notre pseudo-Moodle, cette page affiche les posts associés à une UE, du plus récent au plus ancien.

Les posts possibles sont de 2 types : « **message » et « partage de fichier »**

Chaque type de post doit avoir un affichage distinct, non seulement niveau données pour tenir compte des informations spécifiques à chacun, mais aussi niveau CSS, afin de les reconnaître facilement en un coup d'œil *(icônes différentes, et peut être même couleurs différentes)*.
Les 2 types de posts sont décrits en détail dans la section IV. Merci de vous y reporter pour les informations supplémentaires sur les informations qu'ils contiennent et comment vous pourriez les afficher.

Les posts dans une UE sont rangés dans un ordre chronologique décroissant (du plus récent au plus ancien) et ils sont TOUS affichés, peu importe leur nombre.

Si je suis login en temps que professeur, un bouton « Créer un nouveau post » est disponible en haut à droite de la page, me permettant d'ajouter un post à l'UE.
L'ajout a lieu via une page séparée décrite ci-après.
De même, si je suis login en temps que professeur, deux boutons sont ajoutés près du titre de chaque post : « modifier » et « effacer ».

La modification a lieu via une autre page (qui est la même que celle de création).
L'effacement, lui, devra avoir lieu directement en AJAX, sans avoir besoin de changer de page.

Javascripts demandés

L'effacement des posts doit se produire via AJAX, sans changer de page. Une pop-up demandant confirmation doit également être affichée avant que l'effacement et la mise à jour de la page aient vraiment lieu.

III-E) Page de création/modif de Post *(prof)*

La création des 2 types de posts doit être gérée par la même page.
En haut de la page, deux gros boutons permettent de choisir le type de post parmi ceux disponibles (message texte, partage fichiers). Selon le type de post choisi, un formulaire différent est affiché.

> **Javascripts demandés**
>
> Les deux formulaires apparaissent et se cachent selon le bouton cliqué en dernier. Au final, c'est une logique similaire à celle des onglets de la page admin.

Les formulaires ont beau avoir des points communs (présence d'un titre, d'un « texte descriptif/principal), ils peuvent être totalement distincts au niveau du HTML.

Pour voir ce que chaque type de post doit contenir, et pouvoir en déduire les champs nécessaires pour chaque formulaire, consultez la section IV, consacrée à la description des types de posts.

III-F) Liste des inscrits à une UE *(prof, étudiant)*

Une UE est, pour l'essentiel, sa « page de posts ». C'est là que l'essentiel des interactions du « Moodle simplifié » auront lieu. ...Mais elles contiennent aussi une seconde page : **une liste des participants/inscrits ».**

Cette page, accessible par un lien sur la page « de posts » d'une UE, **affiche la liste de tous les professeurs et tous les étudiants assignés à cette même UE.**

- Les noms, prénoms et e-mails de chaque personne de l'UE sont affichés.
  Les mails sont des « liens mailto » qui permettent d'ouvrir automatiquement le logiciel de mail de la machine pour envoyer un mail à l'adresse en question.
- Rien n'est éditable. Cette page n'est que consultative, et est générée à partir du travail fait par les admins dans leur espace réservé.
- Séparez les profs et étudiants en deux listes distinctes, pour la clarté

III-G) Page de gestion de Compte *(tout le monde)*

Une page très minimaliste, accessible en permanence par un lien « dans la barre de titre », pour peu que l'on soit loggé.
Elle permet à l'utilisateur de modifier son nom, prénom, et/ou son mot de passe, via un petit formulaire. (le mail étant une clé unique dans la BDD, et les UE étant attribuées par les admins, ils ne peuvent pas être modifiés).

Nice To Have
La gestion de compte étant si restreinte, il est possible de la placer dans un simple « menu déroulant », qui pourrait être ouvert depuis n'importe quelle page et permettre une édition via AJAX.
Vous avez aussi l'option de rajouter un upload d'avatar/photo, ou d'autres informations que vous pensez pertinentes de pouvoir enregistrer dans un tel outil (numéro de téléphone ? Adresse postale ? ...)

IV – Types de Posts pouvant être crées

Un utilisateur « professeur » peut ajouter du contenu sur les pages d'UE.
Votre « Moodle simplifié » doit proposer, au minimum, ces trois types de posts.

IV-A) Les messages Texte

Le type de poste le moins complexe. I**l est simplement constitué d'un titre, une date+heure de post, d'un « type de message » et d'un texte** (potentiellement long, plusieurs paragraphes).
« Type de message » désigne des catégories de message pré-établies dans le système. Je vous en demande au moins deux : « Information » et « Important » Vous avez l'option d'être créatif et d'en ajouter davantage.
A l'affichage, ces informations ressortent simplement pour afficher un texte avec un titre, le titre étant précédé d'une petite icône adaptée.
Le « type de message » va amener l'affichage d'une icône/image différente, comme par exemple un panneau « ! » pour les messages importants. Si vous le voulez, la nature du message peut aussi affecter le CSS/l'apparence du message de façon plus globale.

Un étudiant ne peut pas interagir avec le message texte, autrement qu'en le lisant. De sa perspective, il n'est pas interactif.
Un professeur, en revanche, a accès à des boutons « éditer » et « supprimer » sur chaque « post message » d'une UE dont il est professeur *(même si un autre professeur, assigné à la même UE, est l'auteur du message).*
Cliquer le bouton éditer ouvre un formulaire d'édition. Comme déjà mentionné, ce formulaire peut se trouver sur une autre page *(et c'est conseillé de commencer ainsi)*

Le processus de création de posts est décrit plus haut, dans la « liste de pages indispensables ». **N'oubliez pas que chaque post, peu importe son type, va générer aussi un affichage sur la liste « activité récente » visible sur la page de choix d'une UE.**

Javascripts demandés

Pour un professeur, la suppression d'un message doit pouvoir avoir lieu sans changer de page : Une pop-up apparaît pour demander confirmation, et si on valide, l'effacement et la mise à jour de la page ont lieu par AJAX.

Cette fonctionnalité est aussi demandée pour les autres types de posts.

Nice To Have

Si vous avez le temps/les moyens, vous pouvez essayer de concevoir un système Javascript/AJAX permettant d'éditer les posts sans quitter la page.

IV-B) Les dépôts de Fichier(s)

Un post « dépôt de fichier » est assez similaire à un message texte : il contient un titre, un texte descriptif, une date et heure de post, et, bien sûr, un fichier uploadé par le professeur.
Le seul type de fichier « valide » absolument exigé/nécessaire est ZIP.

A l'affichage, les dépôts de fichiers s'affiche de façon similaire à un message texte, mais inclut en plus **un lien (très proéminent) qui, quand il est cliqué, lance le téléchargement du fichier associé.**
Les fichiers ont aussi leur propre petite icône (voire CSS si vous voulez), qui les distingue encore davantage des différents types de « messages »

Un étudiant a une interaction assez limité avec un « post fichier », mais cette « petite interaction » est critique : il peut en voir le titre, lire le texte descriptif, et **cliquer le lien afin de télécharger le fichier**.

Un professeur, en revanche, a accès à des boutons « éditer » et « supprimer » sur chaque « post de fichier » d'une UE dont il est professeur *(même si un autre professeur, assigné à la même UE, est l'auteur du message).*
Cliquer le bouton éditer ouvre un formulaire d'édition. Comme pour les posts de messages textes, le formulaire peut être sur une autre page. Ce formulaire propose les mêmes options que pour le message texte, avec en plus l'option de remplacer le fichier associé, si on le désire.

Le processus de création de posts est décrit plus haut, dans la « liste de pages indispensables ».

Javascripts demandés

Pour un professeur, la suppression d'un message doit pouvoir avoir lieu sans changer de page : Une pop-up apparaît pour demander confirmation, et si on valide, l'effacement et la mise à jour de la page ont lieu par AJAX.

(Même demande que pour les « posts message »)

Nice To Have

Idem que les « post messages » : une édition « directe sur la page » comme dans le « vrai » Moodle, serait un plus.
Vous pouvez aussi introduire une gestion de différents types de fichiers en plus du ZIP, et éventuellement faire que cela résulte en des différences visuelles sur le post (icône de type sur le lien, etc...)

V – Autres « Nice To Have »

Vous avez pu constater que ce document est rempli d'encadrés « Nice to Have », suggérant l'implémentation de fonctionnalités facultatives.
Les « Nice to have » font, en réalité, partie intégrale de la note, avec 4 points qui leur sont consacrés. Pour être vraiment « noté sur 20 », vous êtes donc censés en implémenter quelques uns !
Le but de ce système est de permettre aux étudiants de « personnaliser » un peu ce sur quoi ils travaillent, et de capitaliser sur leurs points forts.

Javascript vous parle ? Alors choisissez des Nice to have qui utilisent cet outil. Vous, c'est plutôt le PHP et la BDD ? Alors ajoutez des possibilités supplémentaires dans les posts, etc...
Pour ceux qui sont particulièrement à l'aise avec le web et veulent un challenge, voilà 3 « nice to have » particulièrement ambitieux. Un seul, bien réalisé, vous rapportera déjà une bonne fraction des 4 points. Le dernier, en particulier, vous les rapportera tous.

Ayez bien conscience que ces nice to have sont bien plus complexes que les autres suggestions du document. Et que vous serez moins guidés sur la manière de les accomplir aussi.
Il est plus « rentable » de se focaliser d'abord sur ce qui est exigé, et de voir ENSUITE si vous avez le temps de tenter un de ces objectifs.

V-1) Possibilité d'épingler/désépingler un post

- Lorsqu'un professeur crée/modifie un post qui n'est pas un devoir, il aurait l'option de « l'épingler ».
- Un post épinglé apparaît en haut de la page. Comme un « devoir » le fait dans ce qui est exigé. Dans ce contexte, on peut considérer qu'un devoir est « épinglé automatiquement ».
- Si il y a plusieurs posts épinglés, ils sont triés entre eux selon l'ordre « chronologique décroissant » habituel
- La BDD doit pouvoir stocker si un post est « pinned », et par quel utilisateur
- Le plus simple de ces trois « gros nice to have », et de loin. Il vaut en conséquence, moins de points.

2) Les Admins peuvent jouer un rôle de « modérateur »

- Dans la fonctionnalité de base, un administrateur ne peut pas rentrer dans les UE, sauf si il est aussi un professeur sur cette UE.
- Ce « Nice to Have » transformerait les admins en modérateurs : ils peuvent rentrer dans n'importe quelle UE, et ajouter/modifier/effacer des posts comme le ferait un professeur.
- Il convient, en revanche, que les professeurs soient bien notifiés qu'une action admin a eu lieu sur une de leurs UE, de la prt d'un non-professeur : des alertes doivenet donc être générées et affichées dans « l'activité récente » sur la page de choix d'UE.
- Ces « alertes » doivent être priorisées en haut de la liste d'activité (comme si elles étaient épinglées, donc), et mises en valeur (couleur, image de panneau « ! »…) pour attirer l'attention
- L'alerte doit inclure un lien qui amène sur la page de l'UE concernée, et, si possible, place le focus immédiatement sur le post modifié.
- Un professeur peut retirer une alerte en cliquant sur un bouton « J'ai compris ». Elle ne sera alors plus épinglée en haut de la liste »
- Les « alertes action admin » ont besoin d'avoir une existence dans la BDD
- Ce Nice-to-Have est complexe, mais inclut des similitudes avec des choses exigées, et réutilise certains systèmes, y compris du Nice-to-Have précédent. Ce dernier est donc un marche-pied vers celui-ci.

3) Donner aux profs le contrôle de l'ordre des posts

- L'objectif le plus ambitieux que je vous propose : reproduire une partie des fonctionnalités du « mode édition » que propose Moodle.
- **Pour cela, la logique d'affichage du contenu dans les UE doit être modifiée :** au lieu que le rangement soit basé sur l'ordre chronologique, ce sont les professeurs qui vont totalement contrôler l'ordre d'affichage des posts à l'intérieur de ces groupes.
- Auprès de chaque post, un professeur de l'UE pourra voir une liste déroulante « Déplacer ce post vers... ». Les choix qui se trouvent dans cette liste sont de type « Avant le post (titre du post) », et une dernière option, « à la fin »
- Le post lui-même doit, idéalement, être exclu de sa liste déroulante associée.
- Selon votre préférence, il peut y avoir un bouton pour valider l'action de déplacement, ou alors elle se déclenche automatiquement une fois un choix pris dans la liste déroulante.
- Le déplacement peut avoir lieu via AJAX, pour que l'utilisateur le voie se réaliser sans recharger la page.

- Les déplacements devant être permanents, l’ordre des posts doit être stocké dans la BDD.
- Les posts épinglés (si vous les avez) peuvent ne pas proposer l’option de « changement de position », et garder la logique d’ordre chronologique…
  ...Mais vous pouvez aussi proposer l’option de les ranger « entre eux ».
- Beaucoup de changements dans les comportements PHP, Javascript, et dans la BDD sont nécessaires pour faire ce Nice-to-Have.
  A moins que vous ne soyez déjà expérimentés dans le développement de sites web complets (et peut être même « déjà expirimentés avec un Framework »), il est vraiment déconseillé de le tenter !