

# "PolyStages"

## Projet du 3<sup>ème</sup> semestre (S9)

Une version PDF de ce sujet est disponible ici :  
<http://nicolas.durand.perso.luminy.univ-amu.fr/pub/>

### I) Informations générales

L'objectif du projet est de réaliser une **application de gestion de stages**.

#### *Description*

Le projet consiste à concevoir et à développer une application de gestion de stages d'élèves-ingénieurs Polytech. Grâce à cette application, les élèves ingénieurs pourront uploader leurs **rapports** de stage et leurs **présentations** pour les soutenances, et remplir leurs **grilles** de compétences. Les tuteurs en entreprise pourront remplir en ligne les **fiches d'évaluation** des stagiaires encadrés (c.f. annexes) ainsi que les **grilles** de compétences (c.f. annexes). Les enseignants devront pouvoir **accéder facilement** aux divers documents (rapports, présentations, évaluations, grilles).

#### *Mots-clés*

Node.js, JavaScript, HTML, CSS, MySQL, Sécurité, etc.

#### *Travail demandé*

Vous devrez écrire un **cahier de conception** décrivant les solutions conceptuelles et techniques choisies pour la mise en œuvre, un **cahier de tests** et un **guide de déploiement**. Une **application fonctionnelle**, testée en condition réelle, devra être livrée à la fin du projet.

Ce projet mettra à l'épreuve vos capacités d'**auto-apprentissage** et de **travail en équipe** qui sont des aspects importants du métier d'ingénieur. Concrètement, cela signifie que vous allez devoir apprendre un environnement, un langage, une notion, etc. que vous ne connaissez pas en vous aidant de livres, de tutoriels, etc. (certains vous sont donnés comme points de départ à la fin de ce document). De plus, vous devrez **chercher les solutions** à vos problèmes de conception et de développement. Enfin, vous devrez vous organiser pour travailler efficacement en équipe (répartition et planification des tâches, outils collaboratifs, etc.).

#### *Déroulement*

Le projet se déroulera de fin novembre 2019 à mi janvier 2020.

Le projet sera réalisé par **groupe de 4**. Les groupes seront fixés avec les encadrants.

Les **livrables** seront le cahier de conception, le cahier de tests, le guide déploiement, un rapport final (organisation du travail, planification des tâches, accomplissement des objectifs, bilan, perspectives, etc.), une présentation et le code de l'application (code source et autres fichiers comme, par exemple, scripts sql, dump de votre base de données, readme, etc.).

Une **soutenance incluant une démo** aura lieu mi janvier 2020.

## II) Fonctionnalités de l'application

L'application devra permettre l'upload et le download de fichiers correspondant aux rapports de stage et aux présentations (avec éventuellement une date limite pour l'upload des rapports). Pour les rapports, seuls les fichiers PDF seront autorisés.

Les fichiers seront renommés et classés automatiquement dans des dossiers. Par exemple, pour un rapport de stage de 3A en 2019-2020, dossier "2020/3A/", fichier "2020\_3A\_NomEleve\_PrenomEleve\_rapport.pdf".

L'idée est d'avoir ensuite un dossier AMUBOX partagé entre les enseignants.

Il faudra bien entendu utiliser des formulaires pour l'upload des rapports et des présentations, compléter les informations des stages, et remplir les fiches d'évaluation des stagiaires.

Pensez à "externaliser" le plus de choses possibles (par exemple, ne pas mettre les questions de la fiche d'évaluation "en dur" dans le code).

L'application devra permettre à un administrateur de lancer la procédure d'évaluation des stagiaires d'une certaine année par les tuteurs en entreprise. L'application enverra alors un email à chaque tuteur pour l'inviter à compléter la fiche d'évaluation (via une URL générée automatiquement). De même pour les grilles de compétences (stages 5A).

Une fois remplie, la fiche d'évaluation sera transformée en un fichier PDF et stockée selon le même procédé que les rapports de stages (avec l'exemple précédent, on aura "2020/3A/2020\_3A\_NomEleve\_PrenomEleve\_evaluation.pdf").

Les textes des messages envoyés automatiquement devront être externalisés dans des fichiers afin de faciliter la maintenance et l'évolution.

Dans cette version de l'application, les stages ne seront pas chargés automatiquement à l'avance. Chaque élève entrera les informations liées à son stage lors de sa première connexion.

L'application sera capable d'envoyer automatiquement une alerte (par email) à l'administrateur si des élèves sont en retard (rapports non rendus) ou si des élèves n'ont pas eu d'évaluations (sauf stages 3A). Pour chaque élève, il y aura son nom, son prénom, son adresse email, l'adresse email du tuteur, etc.

Une partie réservée à l'administrateur permettra d'initier les évaluations et les compétences (comme vu précédemment) pour un couple niveau/année choisi, de consulter les informations sur les stages d'un couple niveau/année, d'accéder à certaines statistiques, ...

Remarques : actuellement, seuls les stages 4A et 5A sont évalués par les tuteurs en entreprise. Les grilles de compétences sont remplies seulement par les élèves 5A et leurs tuteurs. Il n'y a pas forcément de soutenance pour les stages 3A.

## III) Sécurité de l'application

Les communications devront être confidentielles (utilisation de SSL/TLS (HTTPS)).

L'application devra éviter l'injection de commandes et l'injection SQL.

Les utilisateurs (élèves et enseignants) s'authentifieront via le CAS de l'université. On pourra mettre en place une authentification locale comme solution alternative (en plus du CAS AMU). L'identifiant pourra être l'adresse email amu. Les tuteurs en entreprise n'auront pas à s'authentifier (ils recevront directement l'URL du formulaire d'évaluation à remplir).

Les données sensibles de la base de données devront être protégées. Par exemple, les mots de passe devront respecter les conventions actuelles de sécurité. Ils ne seront pas stockés en clair dans la base de données mais stockés hachés. Comment traiter l'éventuelle confidentialité des rapports et des présentations ?  
Etc.

#### IV) Fiabilité de l'application

Le serveur devra être disponible quelque soit le problème rencontré. Par exemple, on pourra utiliser "forever" pour que le serveur tourne en permanence.  
Etc.

#### V) Ressources

(regroupées ici : <http://nicolas.durand.perso.luminy.univ-amu.fr/pub/projetalts9/ressources/>)

[1] Tuto "Des applications ultra-rapides avec Node.js" (Mathieu Nebra) (livre chez Eyrolles)  
<http://sdz.tdct.org/sdz/des-applications-ultra-rapides-avec-node-js.html>

<https://storage.taz8du29.fr/Tutos%20SDZ/Tuto%20OCR%20-%20Node.js%20.pdf>

-> Application "ToDo List" (utilisant express, ejs, express-session, body-parser)

[2] Vidéos/tutoriels Youtube Grafikart (en 6 parties)

<https://www.youtube.com/user/grafikarttv/search?query=nodejs>

-> Application "Livre d'or" (utilisant MySQL, express, ejs, express-session, body-parser, moment)

[3] Exemples de base : <https://www.w3schools.com/nodejs/>

[4] Livre gratuit "Apprenez Node.js"

<https://riptutorial.com/fr/ebook/node-js>

[5] Site/Tuto/Livre "Node.js : Apprendre par la pratique" (Thomas Parisot) (livre chez Eyrolles)

<https://oncletom.io/node.js/>

#### VI) Informations techniques

##### Modules

Vous utiliserez le module "**express**".

Proposition d'autres modules : ejs, mysql, fs, formidable, express-session, body-parser, moment, nodemailer, https, cas-authentication, passport, crypto, helmet, forever, ...

## **Architecture globale**

L'architecture globale d'un projet node (utilisant express) est en général :

fichiers : server.js ou app.js, package.json,

dossiers : public/, views/, [routes/,] models/, middlewares/, config/

## **Base de données MySQL à l'école**

Vous trouverez toutes les informations sur la demande de création et l'utilisation de votre base de données MySQL à l'école ici :

[http://nicolas.durand.perso.luminy.univ-amu.fr/pub/bd3/intranet/TPaide\\_mysql.pdf](http://nicolas.durand.perso.luminy.univ-amu.fr/pub/bd3/intranet/TPaide_mysql.pdf)

Nom du serveur MySQL : dbs-perso.luminy.univmed.fr

Voici une version de départ de la base de données (à améliorer) :

Les clés primaires sont signalées avec un \* et les clés étrangères avec #.

**eleves** (ideleve\*, nom, prenom, email, numetudiant, mdp)

email est unique, numetudiant est unique.

mdp est le mot de passe.

**inscription** (#ideleve\*, annee\*, niveau)

annee correspond à l'année N de l'année universitaire N-1/N. Par exemple 2020 représente l'année universitaire 2019/2020.

niveau correspond à 3A, 4A ou 5A.

**enseignants** (idens\*, nom, prenom, email, mdp, admin)

email est unique.

admin indique si l'enseignant est un administrateur de l'application.

**tuteurs** (idtuteur\*, nom, prenom, email, #identreprise)

email est unique.

**entreprises** (identreprise\*, sigle, nomcomplet)

sigle est une chaîne de caractères en majuscules.

**stages** (idstage\*, #ideleve, niveau, annee, #idtuteur, #identreprise, #idens, datedebut, datefin, titre, description, adressestage, adremailstage, fichrapport, daterapp, fichpresentation, datepres, ficheevaluation, dateeval, confidentiel)

idens correspond l'enseignant référent du stage. Peut être NULL.

description correspond quelques phrases expliquant le contenu du stage

adressestage est l'adresse où le stage a lieu.

adremailstage est l'adresse email du stagiaire pendant le stage (email entreprise).

Peut être NULL. adremailstage est unique.

fichrapport correspond au chemin vers le fichier du rapport de stage.

daterap est la date (TIMESTAMP) d'upload du rapport

fichpresentation correspond au chemin vers le fichier de la présentation en vue de la soutenance.

datepres est la date (TIMESTAMP) d'upload de la présentation.

ficheevaluation correspond au chemin vers le fichier correspondant à l'évaluation remplie par le tuteur entreprise.

dateeval est la date (TIMESTAMP) de l'évaluation.  
confidentiel indique si le rapport et la présentation sont confidentiels.

Contraintes d'intégrité :

Un élève ne peut effectuer qu'un seul stage par année universitaire (et dans le niveau correspondant à son inscription).

Pour une année donnée, un élève ne peut faire un stage que pour le niveau où il est inscrit.  
Etc.

Réfléchissez bien aux types de données afin de choisir les plus adaptés. Pensez à toutes les contraintes d'intégrité (des plus simples au plus complexes).

Le tables "enseignants", "entreprises", "eleves" et "inscription" devront être préremplies (automatiquement ou manuellement).

### ***Central Authentication Service (CAS) d'AMU***

<https://ident.univ-amu.fr/cas>  
<https://ident.univ-amu.fr/cas/login>  
<https://ident.univ-amu.fr/cas/logout>

## **VII) Annexes**

- Fiche d'évaluation de stage 4A.

- Fiche d'évaluation de stage 5A.

- Grille de compétences (pour stages 5A).

Remarque : pour chaque compétence (i.e., ligne), il faut indiquer un niveau (i.e., colonne).

Vous pouvez télécharger les annexes ici :

<http://nicolas.durand.perso.luminy.univ-amu.fr/pub/projetalts9/ressources/annexes/>