**Déposer sur LEA (Projet de session) une seule version finale du projet par équipe.**

**Chaque membre de l’équipe doit remplir ce document et fournir de manière détaillée l’ensemble des fonctionnalités qu’il a mis en place dans le projet dans le document (fonctionnalités\_par\_membre\_équipe) afin qu’il/elle puisse bien être évalué(e) selon ses efforts fournis dans le projet.**

**Nom : Nguyen**

**Prénom : Patrick**

**Numéro de l’étudiant : 0855068**

**Sujet du projet : Portail du département informatique**

**Les membres de l’équipe qui ont participé au projet avec vous :**

**Anayees Sarkes**

**Othmane Sedjari**

**Grille de notation :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Compétences** | **Points** | | | **Commentaires** |
| **Évaluation personnelle et des collègues sur Moodle (10 points)** | | | | | |
| Les évaluations sont remplies correctement | | | 1 |  | |
| Contribution individuelle dans le projet (en cas d’équipe) | | | 5 |  | |
| Collaboration et soutien entre membres de l’équipe | | | 4 |  | |
| **Fonctionnement général (7 points)** | | | | | |
| Respect des consignes, | | | 1 |  | |
| Application fonctionnelle et livraison sur LEA/GitLab | | | 4 |  | |
| Documentation de l’application (documents techniques : fichier readme, diagramme de la base des données, entité-association, explication de l’architecture de l’application, les rôles et fonctions des principaux classes et méthodes) | | | 2 |  | |
| **Base de données (10 points)** | | | | | |
| Base relationnel | | | 1 |  | |
| Cohérence des données fournies | | | 4 |  | |
| Organisation et relations des tables de la base de données | | | 5 |  | |
| **Structure de l'application (20 points)** | | | | | |
| Modèle MVC | | | 2 |  | |
| Fichier config | | | 1 |  | |
| Package Controller **Couche présentation : -Classes contrôleurs -**Logique de l'intégration des classes contrôleurs dans le patron architectural MVC   * Pertinence des contrôleurs développés. * Chacune des classes contrôleurs effectue uniquement l'action qui lui est associée. Chacune des classes contrôleurs ne fait que les opérations nécessaires pour la transmission des données entre le modèle et la vue. * **Classes et gabarits de la vue** * Logique de l'intégration des classes de la vue dans le patron architectural MVC * Pertinence des vues développés * Logique et pertinence du découpage des composants des vues | | | 2 |  | |
| Package action / service **Couche métier ou couche service -Classes action/service**  --Réalisation des fonctionnalités de l’application. -Les classes jouent bien le lien logique entre les classes d’accès aux données et les classes contrôleurs | | | 2 |  | |
| Package dao / repository  **Couche de données -Classes d'accès aux données (DAO)** Logique de l'intégration des classes d'accès aux données dans le patron architectural MCV -Pertinence des méthodes d'accès aux données développées | | | 2 |  | |
| Package entités **Couche de données** : **Classes du modèle** -Pertinence des classes développées en lien avec le modèle de base de données | | | 2 |  | |
| Qualité du code et l’implémentation des différents écrans et les éléments composant l’interface (Utilisation de HTML5, CSS3 et Framework CSS, par exemple Bootstrap de préférence) | | | 2 |  | |
| Utilisation adéquate des pages JSP / Thymealef (inclusion et factorisation des vues communes de l’application) | | | 2 |  | |
| Utilisation des JSTL dans les JSP/ | | | 2 |  | |
| Utilisation adéquate de JavaScript/JQuery et Ajax et traitement asynchrone | | | 3 |  | |
| **Fonctionnalités demandées (40 points)** | | | | | |
| Identification et mis en place des fonctionnalités offertes par l’application et le public cible sont bien respectés | | | 20 |  | |
| Identification et mis en place des différents acteurs et de quelle(s) façon(s) ils utilisent le système (c'est-à-dire, les fonctionnalités que le système offre à chacun des acteurs) sont bien mis en évidence | | | 20 |  | |
| **Exemple détaillé du projet portail du département d’informatique** | | |  |  | |
| Login – connexion  Se souvenir de l'utilisateur (cookie) | | | 3 |  | |
| Inscription | | | 3 |  | |
| Déposer son projet /matériel | | | 5 |  | |
| Consulter la liste des projets des étudiants/ notes de cours d’un professeur | | | 3 |  | |
| Visualiser la vidéo de présentation d’un projet | | | 4 |  | |
| Effectuer une recherche pour trouver un projet ou des notes de cours | | | 3 |  | |
| Cloner / télécharger projet ou notes de cours | | | 3 |  | |
| Envoyer un message aux responsables du projet | | | 3 |  | |
| S’inscrire au service d’aide et de tutorat (actions d’un étudiant en difficulté, actions d’un tuteur) | | | 3 |  | |
| S’inscrire dans la liste des anciens étudiants | | | 1 |  | |
| Donner une évaluation générale à un projet | | | 1 |  | |
| Fournir des liens utiles internes | | | 1 |  | |
| Validations côté client/ serveur | | | 2 |  | |
| Administration du site | | | 5 |  | |
| **Fonctionnalités supplémentaires (3 points)** | | | | | |
| Ajout d’une ou plusieurs fonctionnalités supplémentaires, originalité de l’application et appréciation globale :   * **Communication avec une application Angular (5 bonus)** * Instaurer un système d'évaluation des membres (rating) * Ajouter un chat * Internationalisation du site (Anglais/Français) | | | 3 |  | |
| **Présentation de l'application (10 points)** | | | | | |
| Esthétique et ergonomie du site | | | 4 |  | |
| Présentation oral | | | 3 |  | |
| Présentation power-Point | | | 3 |  | |
| **Total** | | | **100** |  | |