

Etablissement : ISET-Charguia	Département : Technologies de l'Informatique
Unité : Framework Côté Client	Année Universitaire : 2023 - 2024
Classes : DSI 21 & DSI 22	Enseignante : Amel TRIKI

Mini Projet : Développement d'une application de promotion des activités/ événements d'un club

Vous faites partie d'un club à l'ISET de Charguia (Google, ArtVibe, Enactus, Ou même un club qui n'existe pas : théâtre, peinture, ...) et vous souhaitez développer une application web (CRUD¹) qui permet de promouvoir les activités/ événements/ actions de votre club.

Une activité peut être une formation, une sortie, une collecte, une action sociale, une fête, ... : c'est à vous de la définir.

L'application doit comporter 2 parties :

- Une partie back office destinée à l'administrateur du site
- Une partie front office destinée aux étudiants (ou tout autre internaute) qui pourraient consulter le site web.

L'activité de votre choix est décrite par:

- Un identifiant (de type numérique qui sera unique et que vous ne serez pas amené à gérer)
- Un intitulé (ou titre ou nom, ...) pour désigner l'intitulé de l'activité
- Une photo
- Un champ numérique à définir
- Un champ booléen à définir
- Un champ de type Date à définir
- Un champ de type tableau d'**objets** (liste des intervenants, endroits, prérequis, ...)
- D'autres champs si vous le souhaitez (catégorie, lieu...)

La nomination des champs est **libre**.

Exigences fonctionnelles :

1. Gestion de la partie back office

La partie back office est destinée à l'administrateur, elle doit permettre de :

- Consulter, ajouter, modifier, supprimer une activité (que vous avez définie)
- Rechercher une activité
- S'authentifier à l'application
- Changer son mot de passe

¹ CRUD : Create, Read, Update & Delete : Ajouter, afficher, modifier et supprimer

- Une autre fonctionnalité de votre choix

2. Gérer la partie front office

La partie front office est destinée à l'étudiant qui va visiter le site, elle doit permettre de :

- Consulter l'ensemble des activités. Chaque activité n'affichera qu'une partie de ses champs : (*par exemple : intitulé, photo et date*).
- Consulter le détail d'une activité qui décrit l'ensemble des informations relatives à une activité (Après avoir sélectionné au préalable une activité parmi la liste des activités)
- Rechercher une activité selon un ou 2 critères de votre choix (Exemple : activité à une date donnée et qui concerne une catégorie « formation » ou bien juste une recherche à une date donnée)
- Consulter une information publicitaire (photos, temps actuel,...) provenant d'une **API** gratuite de votre choix dont l'emplacement au niveau du site est libre.
- Consulter une interface d'informations qui comporte des informations sur votre site telle que l'auteur de l'application (vous) : c'est un contenu statique.
- Une autre fonctionnalité de votre choix

Exigences techniques :

L'application doit définir :

- Une classe typescript définissant votre activité et d'autres classes si nécessaire
- Un ou plusieurs services pour la gestion de la partie métier.
- La liste des activités doit être placée dans un composant parent et chaque activité doit être représentée dans un composant enfant
- Un ou plusieurs pipes (dont au moins un pipe personnalisé)
- La directive ngClass
- Un menu pour l'étudiant pour la navigation entre les interfaces et un menu différent pour l'administrateur.
- Des routes différentes pour l'affichage de l'activité ainsi que son détail (dans la partie front office)
- Des routes enfants (à vous de choisir si c'est dans la partie front office ou back office)
- Des guards pour la sécurisation des routes
- Des formulaires réactifs.
- Vous devez utiliser ou combiner un ou plusieurs frameworks de style de votre choix : Bootstrap, Angular Material, Materialize CSS, ...

De même l'application doit :

- Pouvoir accéder à un serveur JSON du moment que les données seront sauvegardées dans un ou plusieurs fichiers json.
- Etre déployée sur Surge.sh ou équivalent

Travail à rendre:

Vous devez remettre :

1. Au plus tard le **dimanche 5 Novembre 2023, 23h55** : sur la plateforme iset.uvt.tn :
Une première version de la fiche de projet à compléter
2. Au plus tard le **dimanche 26 Novembre 2023, 23h55** sur la plateforme iset.uvt.tn :
 - Un dossier compressé qui comporte :
 - Votre projet Angular (sans node_modules ni .angular) qui porte le nom : **Projet_Nom1_Nom2** où Nom1 et Nom2 désignent les noms du binôme (Exemple : **Projet_Rouissi_Jebali**).
 - Une fiche de projet à compléter par le binôme/ monôme
 - Un commentaire indiquant l'URL vers votre dépôt sur Github
(Exemple : https://github.com/Rouissi/Projet_Rouissi_Jebali.git)

Bonus:

Vous pouvez être bonifié dans le cas où :

- Vous utilisez Firebase et déployez votre application avec.
- Vous intégrez un backend de votre choix (Node JS, Spring Boot,...)
- Vous utilisez d'autres concepts avancés d'Angular : les modules, le lazy-loading...
- Vous utilisez un framework CSS autre que Bootstrap destiné à Angular tel qu'Angular Material.

Système d'évaluation:

L'évaluation portera sur un ensemble de critères tels que :

- La réponse aux exigences fonctionnelles et techniques
- La qualité du code, tests effectués, ...
- L'ergonomie de l'application
- L'utilisation de git et de Github
- La complexité et l'implémentation des 2 fonctionnalités ajoutées

Il est à noter que :

- **Le meilleur mini-projet par classe se verra octroyé la note de 20.00 /20.00**
- 2 codes identiques seront sanctionnés d'un **zéro**.
- Le projet peut être fait en binôme ou en monôme, c'est au choix, c'est la qualité du projet qui sera évaluée indépendamment du nombre d'étudiants qui y travaillent
- Le mini-projet fera l'objet d'une validation, tout étudiant qui ne saura pas expliquer son code ou qui n'assistera pas à la séance de validation aura une note de **zéro**

L'application doit être fonctionnelle/exécutable ne comportant pas d'erreurs de compilation, autrement, la note sera systématiquement au-dessous de 10