

Sujets des projets Web-Avancé :

Sujet A : Prise en charge des réclamations des clients « Retail » particuliers d'une banque coté client.

Effectif max : 3 étudiants

L'application doit permettre à un client de la banque d'exprimer sa réclamation :

- Le client ouvre l'application (sans connexion : « without login page ») et remplit le formulaire de la réclamation (nom, prénom, date naissance, civilité, CIN, RIB, sujet de la réclamation, texte de la réclamation etc. ...)
 - Les sujets des réclamations peuvent être prédéfinis (exple : compte, carte, accueil dans l'agence etc. ...).
 - La SPA (single page application) doit compter minimum deux composants métiers [Components] (étape 1 et étape 2) :
1. **La première étape 1** : le client remplit ses données persos, puis en cliquant sur (exemple : bouton ou lien ou autre) il passe à un autre formulaire.
 2. **La deuxième étape 2** : le deuxième formulaire où il remplit les données de la réclamation (sujet, texte etc....) et envoie (submit sa réclamation)

Notions :

Front End :

- Components & modules (composants et modules angular)
- Routing (routage entre composant) : étape 1 et étape 2 sont le minimum de composant à construire, il vaut mieux utiliser le routage pour naviguer entre les deux
- Communication : Passage d'information entre composants (soit avec les @Input, @Output soit avec les services @Injectable, soit avec le routage [paramètres passés pendant le routage])
- Angular Forms (les formulaires dans angular) : vous allez utiliser les formulaires de angular (validation, databinding etc ...) (visitez la documentation officielle de angular pour les formulaires)

Back End :

- ExpressJs : développer un serveur express qui expose des APIs pour :
 - La liste des sujets prédéfinis des réclamations (prévoir une collection « subjects » dans la base de données)
 - Sauvegarder les réclamations (prévoir une collection « réclamations » dans la base de données)
- Mongo-DB :
 - Créer une base de données
 - Créer les collections nécessaires (« subjects » , « réclamations » etc....)

Sujet B : Prise en charge des réclamations des clients « Retail » particuliers d'une banque coté banque

Effectif max : 5 étudiants

L'application doit permettre à un agent de la banque de prendre en charge les réclamations des clients :

- L'agent ouvre l'application (avec connexion : « with login page ») il introduit un login et un mot de passe
- L'agent consulte la liste des réclamations (il peut filtrer la liste avec : sujet de la réclamation, CIN du client, date, état de la réclamation etc...), la réclamation peut avoir les états (en attente, traitée, rejetée)
- La SPA (single Page Application) doit compter minimum trois composant :
 1. **Login Page** : contenant un formulaire simple (login et mot de passe)
 2. **Liste des réclamations** : (tableau avec filtre ou autre composant au choix)
 3. **Un écran de modification de la réclamation** : seule l'état de la réclamation est modifiable :
 - De « En Attente » => « Traitée »
 - De « En attente » => « Rejetée »

Notions :

Front End :

- Components & modules (composants et modules angular)
- Routing (routage entre composant) : login page, liste des réclamations, edit reclamation sont le minimum des composants a construire, il vaut mieux utiliser le routage pour naviguer entre eux (***l'utilisation du « Guard » pour le passage de Login => liste des réclamations est un plus***. Visiter la documentation officielle de angular pour connaître davantage sur les « Guards »).
- Communication : Passage d'information entre composants (soit avec les @Input, @Output soit avec les services @Injectable, soit avec le routage [paramètres passés pendant le routage])
- Angular Forms (les formulaires dans angular) : vous allez utiliser les formulaires de angular (validation, databinding etc ...) (visitez la documentation officielle de angular pour les formulaires)

Back End :

- ExpressJs : développer un serveur express qui expose des APIs pour :
 - L'authentification (login & mot de passe) (prévoir une collection « credentials » dans la base de données)
 - La liste des réclamations (prévoir une collection « reclamation » dans la base de données)
 - Modification d'une réclamation (changement de son « état »)
- Mongo-DB :

Créer une base de données

Créer les collections nécessaires (« agent et/ou credentials », « reclamation » etc ...)

NB : Sécuriser les API (coté serveur [utilisation de JWT : Json Web Token par exemple] est un plus mais il est impératif dans le cas des groupes de 5 étudiants)

