# PROJET WEB CLIENT-SERVER : 2020

## SCOPE DU PROJET

Mettre en place un website publique avec authentification et rôles utilisant un backend composé d’api en .net core 3.1.  
  
Le but du website est de permettre à des gestionnaires de bar, boîte de nuit, salle de concert, cercle étudiant de se créer un compte sur le site et d’inscrire un ou plusieurs établissements.

Un administrateur doit valider les établissements avant qu’ils puissent apparaitre de manière publique pour un utilisateur non loggé.

Les rôles seront respectivement :

* Administrateur
* Gestionnaire
* Utilisateur
* Anonyme

Un administrateur :

* Peut lister et valider les inscriptions des établissements
* Peut lister/ajouter/modifier/supprimer des news
* Peut lister/ajouter/modifier/supprimer des établissements
* Peut lister et supprimer des comptes utilisateurs
* Permettre d’utiliser les réseaux types : facebook, instagram, twitter directement du website (avec les hashtags lié à l’application)

Un Gestionnaire :

* Peut lister/ajouter/modifier/supprimer des établissements
* Permettre de lister/ajouter/modifier/supprimer des horaires.
* Générer un « shorty url » par établissements
* Permettre d’utiliser les réseaux types : facebook, instagram, twitter directement du website.
* (avec les hashtags lié à l’application)
* Lister l’ensembles des établissements et validés par l’administrateur
* Visualiser le détail réduit des établissements

Un utilisateur :

* Peut se ajouter/modifier/supprimer un compte utilisateur
* Permettre d’utiliser les réseaux types : facebook, instagram, twitter directement du website (avec les hashtags lié à l’application)
* Lister l’ensembles des établissements et validés par l’administrateur
* Visualiser le détail des établissements (horaire, signalétique, photo…)

Un utilisateur anonyme :

* Lister l’ensembles des établissements inscrits et validés par l’administrateur
* Visualiser le détail réduit des établissements (signalétique, photo)

## CONTRAINTES STRUCTURELLES :

* Création d’un compte :
  + Nom
    - max 50 char
    - Pas de caractères spéciaux
    - Obligatoire
  + Prénom
    - Max 50 char
    - Pas de caractères spéciaux
    - Obligatoire
  + Adresse email
    - Max 255 char
    - Pas de caractères spéciaux
    - Adresse email valide
    - Adresse email unique (identifiant utilisateur)
    - Obligatoire
  + Numéro de téléphone
    - Max 25 char
    - Téléphone mobile
  + Sexe
    - Liste déroulante (Male, Female, Non binary)
    - Obligatoire
  + Date de naissance
    - Une date de naissance valide au format (DD/MM/YYYY)
    - Obligatoire
  + Professionnel
    - Boolean
    - Obligatoire
* Création d’un établissement :
  + Identification
    - Type d’établissement :
      * Une énum (Bar, boite de nuit, salle de concert, cercle étudiant)
      * Obligatoire
    - Nom de l’établissement
      * Max 50 char
      * Obligatoire
    - Numéro de TVA
      * Numéro de tva valide
      * Format du numéro de tva valide
      * Obligatoire
    - Adresse email pro
      * Max 255 char
      * Pas de caractères spéciaux
      * Adresse email valide
      * Obligatoire
    - Zone de texte libre (max 2000 char)
    - Logo de l’établissement
      * Upload d’un fichier de type image (pgn, ico, jepg, …)
      * Max size 1Mo
  + Infos établissement
    - Adresse
      * Code postal
        + Max 20 char
        + Obligatoire
      * Ville
        + Max 100 char
        + Obligatoire
      * Pays
        + Liste déroulante
        + Obligatoire
      * Rue
        + Max 100 char
        + Obligatoire
      * Numéro-Boite
        + Max 20 char
        + Obligatoire
    - Numéro de téléphone
      * Max 25
      * Fixe ou Mobile
    - Adresse email établissement
      * Max 255 char
      * Pas de caractères spéciaux
      * Adresse email valide
    - Site web
      * Format URL
    - Instagram :
      * Format URL
    - Facebook
      * Format URL
    - LinkedIn
      * Format URL
  + Horaire de l’établissement :
    - Visualiser
  + Photos :
    - max 5 photos
    - max 3Mo par photos

## CONTRAINTES TECHNIQUES :

* Utilisation de .net core 3.1 pour la partie backend (api)
* FrontEnd : angular, knockOut, React, Razor, Blazor.
* Authentication et autorisation via IdentityServer4
* La validation du numéro de TVA doit être validé par une api externe
* La validation du numéro de téléphone doit être validé par le google libphonenumber
* La maps doit être créer via une api externe
* Integration d’un date picker pour les dates
* Utilisation de notification type toastr pour le opération de sauvegarde afin d’afficher un message à l’utilisateur finale
* Validation des modeles
* Pagination
* Usage de script
* Usage de call AJAX

## DEMANDE BUSINESS :

Les établissement seront affichés par défaut sur base d’une carte.  
  
Les utilisateurs authentifiés visualiseront sur la carte uniquement les établissements ouverts le jour-même, un indicateur affichera clairement le statut d’ouverture en fonction des horaires. Une fois l’horaire dépassé l’établissement disparait de la carte. Rafraichir automatiquement la carte, toutes les 15 minutes.  
  
Les utilisateurs non authentifiés visualiseront sur la carte TOUS les établissements inscrits.

Les news seront accessibles pour tous le monde, avec un animation au choix et publieront automatiquement sur instagram, facebook, twitter, linkedIn.

## REFERENCE TECHNIQUE :

N’hésitez pas à trouver d’autres alternatives sinon voici des références utiles.

Integration identityServer4 : <https://feras.blog/how-to-use-asp-net-identity-and-identityserver4-in-your-solution/>

Documentation about IdentityServer4 : <https://www.scottbrady91.com/Identity-Server>  
  
Validation of Belgian VAT (use free version with limited number of request per day) : <https://vatlayer.com/>

Libphonenumber : <https://github.com/twcclegg/libphonenumber-csharp>

Utilisation d’une maps afin d’afficher les établissements : <https://developers.google.com/chart/image/docs/gallery/new_map_charts>

Integration de toastr : <https://codeseven.github.io/toastr/>