**Présentation d’un projet de Data Visualisation**

1. INTRODUCTION
   1. Pourquoi ce projet ?
   2. Comment le projet s’est-il construit ?
   3. D’où vient l’idée ?
   4. Pourquoi ne pas avoir utilisé les logiciels de data visualisation prévu à cet effet ?

* Présentation de données traitées.
* Formation en data visualisation.

1. LES OBJECTIFS DU PROJET
   1. *LES OBJECTIFS ‘utiles’ DU PROJET :*

* Mettre en évidence des données issues de médias belges (lesquelles ? et pourquoi ceux-là ? Afin de connaitre la neutralité politique des médias d’un point de vue quantitatifs)
* Le but était de mettre en évidence des données « politico-médiatiques » belges traitées (au travers de requêtes SQL, au travers de librairies JS de présentation de données, …) concernant :
  + Des mandataires politiques belges (députés, ministres, bourgmestres, …)
  + Des parti politiques belges.
  + Des médias belges.
  1. *LES OBJECTIFS PERSONNELS DU PROJET :*
* Apprendre un Framework Backend (Symfony car connu donc plus de ressource en cas de problèmes)
* Apprendre un Framework Frontend (Angular pour les mêmes raisons)
* Améliorer mes connaissances en PHP et en JavaScript (et m’initier à TypeScript par la même occasion)

1. LE PROCESSUS DE L’APPLICATION

* Comment est-ce que cela se déroule ?

(Diviser en :

* Partie Administration de l’application (côté serveur, les points A à C) => ensemble des services permettant le traitement et l’insertion en BDD des flux RSS (TRAITEMENTS), ainsi que de répondre aux requêtes côté client (API)
* Partie présentation de l’application (côté client) => ensemble des services qui vont permettre de présenter les données ainsi que de traiter certains types de données (exemples ?) afin qu’elles soient actualisées au moment où une requête AJAX est faite par le client)
  1. *Traitements des flux RSS pré importation en BDD:*
* Liens des flux RSS médias belges ;
* Récupérations des données de ces flux (nom média, titre article, description article, date article, catégorie article, lien vers l’article) ;
* Traitements de ces flux (nettoyage des données => récupération du nom de la catégorie dans l’url ou via l’url si pas présente dans le flux RSS ; formatage du nom du média ex : ‘la dh info’ devient ‘dh’) ;
* Importation des données en BDD ;
  1. *Autres traitements post importation en BDD :*
* Update du ‘Nom’ de la table ‘Media’;
* Update de la ‘Catégorie’ de la table ‘Média’ ;
* Insertion dans la table ‘Medpol’ des clés étrangères des tables ‘Politicians’ et ‘Media’ où il y a concordance entre le nom et le prénom de la table ‘Politicians’ ET la description de la table ‘Media’.

(En gros : remplissage de la table intermédiaire ‘Medpol’ en fonction de la table ‘Politicians’ et de la table ‘Media’) ;

* Sauvegarde de la BDD à chaque lancement de l’algorithme d’insertion en BDD ;
  1. *Requêtes SQL côté serveur (via Symfony) :*
     + - * Deux types de requêtes :
* Les requêtes longues (+ de quelques secondes) : le résultat de la requête est stockée dans un fichier de type JSON afin de permettre d’envoyer les informations à Angular plus rapidement. C’est ce fichier qui sera donc appelé par Angular dans une requête AJAX. Ce type de requêtes devrait donc être effectué au moment où de nouvelles données sont insérées en BDD (permettant de faire en sorte que la partie client de l’application soit aussi à jour que la partie serveur).
* Les requêtes courtes : vont renvoyer les données de la requête SQL directement au format JSON.
  1. *Requêtes AJAX côté client (via Angular) :* 
     + - * Deux type de requêtes :
* Les requêtes automatiques (de type GET) : Permettent de visualiser dès l’ouverture de l’application au niveau de la page ‘Homepage’ l’actualité hebdomadaire concernant les politiciens les plus cités dans l’ensemble des médias ainsi qu’au niveau de la page ‘Trends’ les politiciens les plus cités depuis les premières insertions en BDD (fin avril) ;
* Les requêtes sur base d’une recherche / en fonction de critères (de type POST) : Permettent de trouver des articles en fonction de critères de recherches, ainsi que de visualiser par le biais de graphiques généraux le nombre d’articles par médias mais surtout le ratio (en pour mille) du nombre d’articles par média permettant de comparer au travers de la moyenne et de l’écart-type les différents médias sur base d’une recherche.

1. LES PRINCIPES TECHNIQUES DE L’APPLICATION / LA STRUCTURE DE L’APPLICATION

(Côté serveur Symfony comme API et Symfony comme ‘*’’constructeur’’’ ;* Côté client Angular comme module de présentation mais aussi de traitement de requêtes AJAX)

* Symfony, lorsqu’une requête AJAX lui est envoyé par Angular, est utilisé comme API (application programming interface : interface de programmation applicative), il ne fait que délivrer du contenu JSON.
* Mais Symfony n’est pas seulement utilisé comme API mais aussi comme module de peuplement et de traitement de contenu inséré en BDD.
* Angular est principalement utilisé comme ‘module’ de présentation de données qui utilise AJAX afin de faire parvenir des données formatées en JSON.
* Utilisation de la librairie Chart.js

1. LES PERSPECTIVES DE L’APPLICATION

* Parler ici qu’au-delà de la visualisation des données sous formes de charts il y a aussi la présentation de données pertinentes (top articles des 7 derniers jours via l’utilisation de l’occurrence des 3 mots les plus employés ces 7 derniers jours).
* Idée que de pouvoir avoir une application qui centralise visuellement et au niveau de l’analyse, des données ‘politico-médiatiques’ ;

1. LES LIMITES DE L’APPLICATION À CE JOUR

* L’utilisation des flux RSS comme socle d’analyse plutôt que les articles en entiers peuvent limiter l’étendue de l’application et même fausser certains résultats.

(Par exemple le flux RSS de ‘RTL’ fournit l’entièreté de l’article en lieu et place du début de l’article comme le reste des médias et se trouve dès lors en mesure de fournir des résultats plus pertinent concernant à la fois les ‘top articles’ (qui utilisent l’occurrence des termes afin de sélectionner les articles les plus pertinent) mais aussi au niveau des graphiques où plus un article est complet plus les informations à extraire seront diversifiées ; On peut ainsi remarquer que comparativement aux autres médias ‘RTL’ est souvent au-dessus de la moyenne dans le ratio du nombre d’articles par média) ;

* Il faudrait implémenter une recherches SQL plus poussées ainsi qu’utiliser des REGEX afin d’obtenir des résultats de recherche plus proche de ce qui est recherché et ainsi repousser les faux-positifs de l’analyse.

1. AU-DELÀ DU PROJET

* Idée d’une application générique d’analyse de contenu qui fournirait des graphiques / résumés de contenus / les points essentiels d’un document en fonction de critères d’analyse et de critères de recherches.