1. INTRODUCTION
2. Explication du projet :

* CADRE :
* Dans le cadre de ma formation j’ai dû créer un projet de de data visualisation (càd de représenter des données sous forme de graphique).
* De base j’aurais dû utiliser des logiciels de data visualisation prévu à cet effet mais personnellement je voulais utiliser les langages du web (JS, PHP, HTML, CSS, SQL, …) et les concepts de programmation web (AJAX) en utilisant des librairies JavaScript de représentation de données (chart.js).
* OBJET :
* L’objet de recherche qui m’a intéressé, suite à une discussion avec une amie, a été celui de l’analyse de données politico-médiatiques. Ces données ont été obtenues au travers de flux RSS (XML structuré / balisé selon une convention).
* Le but du projet a donc été de mettre en valeur des données à la fois politiques (concernant des mandataires politiques et des partis politiques belges) et aussi des données liées à différents médias belges (rtl info, rtbf, lecho, la dh, …).
* L’autre objectif du projet fut d’améliorer mes connaissances en PHP et SQL dans un premier temps mais aussi d’apprendre un Framework backend (qui a été Symfony) et un Framework frontend (qui a été Angular 2 et version d’après).

1. Comment le projet s’est-il construit :

* Phase 1 :
* Avant de commencer le projet, j’ai testé la faisabilité de l’idée. A savoir s’il était possible d’extraire des données issues de flux RSS et comment le faire.
* Phase 2 :
* Création de la base de données où il y a eu création de la table Médias (celle qui allait recevoir les données issues des flux RSS, et qui allait être une table dynamique) et la création de la table politicians (table plus ou moins statique qui n’allait pas évoluer beaucoup dans le temps et où les données ont été extraites de Wikipedia et du site cumuleo.be).
* Insertion des données issues des flux RSS en BDD
* Automatisation basique des tâches de récupération et d’insertion de contenu en BDD en utilisant l’attribut ‘refresh’ de la balise html.
* Phase 3 :
* Début de refactoring (réusinage de code) en utilisant progressivement des classes en PHP afin de structurer au mieux l’application.
* J’ai commencé à utiliser chart.js afin de mettre en évidence les données issues de la BDD.
* Phase 4 :
* Qui est la phase dans laquelle je suis toujours pour l’instant. J’ai commencé à implémenter le framawork backend Symfony et le framework frontend Angular.

1. Le processus de l’application :

* Cette partie-ci a pour but d’expliquer comment se déroule le processus d’insertion de données et le processus d’exploitation des données. Donc comment la base de données est gérée et comment l’application est gérée.

1. Administration de la BDD :

* Traitement avant import en BDD :
* Au niveau de l’administration de la BDD il y a tout d’abord le traitement des flux RSS avant l’importation en BDD.
* Pour ce point-ci, il y a tout d’abord une récupération des différents liens RSS des différents médias belges qui sont stockés dans un tableau en PHP.
* Ensuite, au moyen d’une boucle foreach, les données issues de ces liens RSS sont récupérées (nom média, titre article, description article, date et la catégorie de l’article).
* Ensuite, au moment où ces données passent dans la boucle foreach, elles sont nettoyées. C’est-à-dire que notamment le nombre de catégorie est réduite afin de n’en avoir que quelque unes et afin d’avoir des catégories pertinentes. Si la catégorie est absente alors il va y avoir une tentative de récupérations soit dans l’url (catégorie se trouve dans l’url pour certain média) soit via l’url (en allant rechercher la catégorie si elle est présente dans l’article). Dans cette partie de nettoyage de données il va aussi y avoir un formatage du nom des médias (rtlinfo va devenir rtl, ladhnet va devenir dh).
* Puis il va y avoir l’importation des données dans la BDD.
* Traitement après import en BDD :
* Dans cette partie-ci, il va s’agir d’avoir une table intermédiaire qui va permettre de lier la table Politiciens à la table média.
* Pour ce faire il y aura insertion dans cette table s’il y a concordance entre les champs nom + prénom de la table politiciens ET le champs description de la table médias.
* Puis il y a une sauvegarde de la BDD

1. Administration de l’application :

* Côté serveur via Symfony :
* DEUX TYPE DE REQUÊTES sont effectuées :
* La première est une requête que j’ai nommé requête de longue durée
  + Ce sont des requêtes qui durent plusieurs secondes et qui ne sont pas directement appelée par Angular lors de requêtes AJAX mais qui sont stockées dans un fichier de type JSON afin de permettre d’envoyer à les données à Angular plus rapidement. C’est donc ce fichier qui sera donc appelé par Angular dans une requête AJAX.
* Le second type de requête est une requête de courte durée càd que lors d’une requête AJAX coté Angular les données renvoyées seront des données JSON issues directement d’une requête SQL et donc non stockées dans un fichier JSON.
* Ainsi Symfony pour cette partie administration de l’application fonctionne comme une API, c’est une interface de programmation applicative càd une interface de programmation qui permet de se brancher / lier à une application afin d’échanger des données.
* Côté client au moyen d’Angular :
* Il y a tout d’abord le premier type de requête AJAX, qui sont des requêtes automatiques. Càd qu’à l’ouverture de l’application une requête AJAX est envoyée afin de visualiser directement (tout en étant à jour par rapport à la BDD) l’actualité hebdomadaire concernant les politiciens les plus cités ces 7 derniers jours pour la Homepage et sur l’ensemble de la base de données pour la partie tendance.
* Ensuite il y a un second type de requête, qui sont les requêtes en fonction de critères de recherches qui permettent de trouver des articles 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 perspectives de l’application
2. Au-delà de la visualisation de données

* Au-delà de la visualisation de données sous forme de charts il y a aussi l’idée de pouvoir présenter des données textuelles pertinentes. Par exemple j’ai créé un algorithme qui permet de récupérer les 3 termes les plus utilisées dans les différents médias des 7 derniers jours et ce afin pouvoir présenter les 10 articles les plus pertinents de ces 7 derniers jours. Pour l’instant, Je ne me suis basé que sur l’occurrence des termes mais il suffirait de se baser aussi sur d’autres critères afin de pouvoir améliorer l’algorithme de recherche.
* Il y aussi l’idée de savoir ce qu’a pu dire un politicien ; J’ai commencé à créer un algorithme qui recherche les citations des politiciens en fonction des articles liés à ces politiciens et les guillemets utilisé dans les articles pour les cités. Cependant cet algorithme n’est pas encore au point vu que dans un article plusieurs politiciens peuvent être cités et dès lors la citation attribuée à un politicien peut être fausse. Il faudrait peut-être utiliser des regex dans l’analyse afin d’être plus sûr concernant l’attribution de citation.

1. Centralisation de données politique et médiatique :
   1. Au travers de cette application il y aussi l’idée de pouvoir centraliser les données politiques et les données relatives aux médias afin d’avoir un résumé de ce qu’a pu dire un politicien mais aussi de permettre d’avoir une analyse sur le contenu des médias. L’idée serait d’avoir à la fois des graphiques résumant certaines idées mais aussi avoir une analyse détaillée de ces graphiques.
2. Les limites de l’application :

* Il y a certains problèmes dans l’application qui sont liés à l‘utilisation des flux RSS. Le fait d’analyser ces flux plutôt que des articles dans leurs entierté peuvent limiter l’étendue de l’application et de ce qu’elle veut mettre en évidence Mais aussi l’utilisation des flux RSS peut fausser l’analyse. 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 le taux de faux-positifs de l’analyse.
* Ce qu’il reste à faire dans un court-terme : migrer partie administration de la BDD vers Symfony et mettre en valeurs plus de résultats concernant les médias.

1. Au-delà de l’application

* Au-delà de ce projet j’ai eu l’idée d’une application générique d’analyse de contenu qui fournirait des graphiques, des résumés de contenus, qui fournirait les point essentiel d’un document en fonction de critères d’analyse et de critères de recherche.