

# PROJET ARGILES

## RESUME DU PROJET

### Vue d'ensemble

Le projet ARGILES qui nous a été confié vise à fournir un système structuré de visualisation et d'analyse des données, stockées dans une base de données de l'UGA, contenant les réponses d'élèves en primaire et collège à des œuvres littéraires, codifiées en XML. Ce projet repose sur PHP et HTML pour l'interface web; Python pour le traitement des données; phpMyAdmin et SQL pour le stockage des informations. L'objectif était de développer et proposer une expérience ergonomique et esthétiquement cohérente aux futurs usagers de ces données, en particulier pour les chercheurs de l'UGA et d'autres établissements publics. Étant donné les contraintes liées à FileZilla et GitHub, nous avons opté pour une organisation des fichiers qui limite les mises à jour inutiles, tout en conservant un affichage personnalisé sur chaque page. Ce projet a été marqué par plusieurs défis techniques et ergonomiques que nous avons progressivement résolus.

Le projet est accessible via GitHub au lien suivant :

<https://github.com/zancap/ARGILES-Database-Management-and-NLP>

Ce projet a été développé dans le cadre d'un master en traitement automatique des langues. Il porte sur la création et la gestion d'une base de données, l'analyse en TAL/NLP, la conception d'un site web pour l'exploitation des données. L'objectif principal est d'offrir un accès libre aux données collectées et annotées afin de favoriser les recherches futures en linguistique, en éducation et en intelligence artificielle.

## Informations complémentaires

L'accès à la base de données se fait via phpMyadmin et avec les informations suivantes, à personnaliser avec vos propres identifiants :

```
$servername = "i3l.univ-grenoble-alpes.fr";  
$username = "...";  
$password = "...";  
$dbname = "argiles";
```

Dossier [gestion\\_bdd](#) sur GitHub → Contient deux scripts utiles à la gestion de la base de données, en particulier un script [phpinfo](#) pour extraire les informations techniques sur la base de données et un fichier [add\\_admin](#) pour la gestion des utilisateurs autorisés (ajoute d'utilisateurs et mots de passe dans la base de données).

Limites d'importation → La taille maximale des fichiers pouvant être téléversés dans la base de données est fixée à 2 Mo.

Emplacement sur FileZilla → Le projet est hébergé sur le serveur à l'emplacement suivant : [/data/www/html/argiles](#)

Pour plus d'informations, et si vous avez besoin d'aide avec le code, n'hésitez pas à nous contacter via nos adresses mail :

[Paola.Zancanaro@etu.univ-grenoble-alpes.fr](mailto:Paola.Zancanaro@etu.univ-grenoble-alpes.fr)

[Romain.Girard@etu.univ-grenoble-alpes.fr](mailto:Romain.Girard@etu.univ-grenoble-alpes.fr)

## Intégration des scripts Python avec PHP

Dès le départ, un des enjeux majeurs a été d'assurer la bonne exécution des scripts Python ([get\\_audite.py](#) et [stats.py](#)) depuis les pages PHP. Plusieurs problèmes sont apparus, notamment les dépendances Python nécessaires à l'analyse des données et la récupération des données à partir d'un tableau dynamique. L'une des erreurs les plus bloquantes concernait l'absence du modèle de traitement `fr_core_news_md` de SpaCy,

qui empêchait l'exécution des analyses textuelles. Après plusieurs tests, nous avons confirmé que l'environnement d'exécution du serveur ne disposait pas de cette dépendance, ce qui aurait nécessité l'installation manuelle de la part d'un acteur externe et qu'on a donc dû éliminer du projet final. Ce problème était d'autant plus critique que la bibliothèque SpaCy était centrale pour le lemmatisation et l'extraction de statistiques sur les réponses des élèves.

En général, cette intégration a été pensée pour offrir une navigation fluide et intuitive, tout en garantissant un accès rapide aux résultats d'analyse. Un autre défi majeur a été de permettre aux utilisateurs d'explorer les données issues des réponses des élèves de manière dynamique et interactive. L'un des éléments clés de l'intégration repose sur le lien direct entre chaque ID d'élève et la page de résultats détaillés. En cliquant sur un identifiant, l'utilisateur est redirigé vers une page qui affiche l'ensemble des réponses de l'élève, classées par catégorie. Cette organisation permet de visualiser rapidement ce que chaque élève a évoqué, aimé, rejeté ou attendu du texte analysé. La transition se fait de manière fluide, et chaque réponse est mise en contexte avec les autres informations disponibles. Dans la même logique, la page de visualisation de la base de données est connectée à une page de statistiques avancées. Cette dernière exploite un script Python pour générer une analyse de fréquence des mots employés par les élèves dans leurs réponses. Grâce à cette fonctionnalité, l'utilisateur peut immédiatement voir quels sont les termes les plus récurrents et ainsi dégager des tendances globales sur les réactions aux œuvres littéraires.

Un autre aspect particulièrement bien intégré est l'usage d'un code couleur dans la présentation des réponses. Chaque type de réponse (évocation, appréciation, rejet, attente) est représenté par une couleur spécifique, ce qui facilite la lecture et la compréhension des résultats. Cette convention visuelle est réutilisée sur les pages PHP et dans les résultats générés par Python, garantissant ainsi une cohérence esthétique et fonctionnelle à travers tout le site. Un bouton de sauvegarde a été ajouté afin de permettre aux utilisateurs de récupérer directement la requête SQL générée. Si cette approche peut sembler rudimentaire, elle offre néanmoins un avantage certain :

l'utilisateur peut copier et coller la requête pour la réutiliser ailleurs, facilitant ainsi l'exploitation des données sans nécessiter une interface supplémentaire de requêtage.

Enfin, ce qui rend cette plateforme réellement performante, c'est son caractère dynamique. L'ensemble des filtres est interactif, permettant d'explorer les données selon divers critères sans jamais recharger la page. Grâce à l'utilisation d'AJAX, les requêtes sont traitées en arrière-plan, rendant la navigation fluide et réactive. L'utilisateur peut filtrer par tous les types d'informations présentes dans la base de données et voir instantanément les résultats correspondants. Cette approche assure une expérience utilisateur optimale, tout en exploitant la puissance combinée de PHP et Python pour gérer efficacement les données et produire des analyses en temps réel. Cette architecture garantit une grande efficacité entre l'analyse de données et leur visualisation, offrant un outil performant et intuitif pour explorer les réactions des élèves aux textes étudiés.

## Gestion et analyses des données et accès sécurisé à la base de données

Un aspect clé du projet était l'intégration d'un système interactif de visualisation des données issues des réponses des élèves. Nous avons mis en place une page de visualisation du contenu la base des données qui permet aux utilisateurs de consulter les enregistrements, de filtrer les réponses et d'accéder directement aux analyses via des liens dynamiques. Une difficulté importante a été d'assurer que chaque sélection effectuée par l'utilisateur (ID audité, titre du roman, etc.) soit transmise correctement aux pages d'analyse, tout en conservant la possibilité de retrouver une analyse passée via des URLs accessibles (GET). Cela permet maintenant à un utilisateur d'accéder à une analyse préexistante directement via une URL, ce qui est particulièrement utile pour la recherche et le partage des résultats. Par ailleurs, nous avons entrepris une sécurisation du système de connexion. Initialement, les identifiants administratifs étaient statiques et stockés en dur dans les fichiers PHP, ce qui représentait un risque. Nous avons donc ajouté une authentification via base de données, avec stockage des mots

de passe via `password_hash()`, garantissant ainsi une connexion sécurisée. Une fonctionnalité essentielle a été mise en place pour permettre l'importation et le stockage des données XML dans la base de données. L'objectif principal était de structurer les réponses des élèves en les associant aux œuvres analysées (Sirius et Roméo et Juliette). Une interface a été développée pour permettre aux utilisateurs autorisés de téléverser des fichiers XML via une page sécurisée du site. Les données extraites sont ensuite insérées dans la base via des requêtes SQL structurées. Chaque œuvre dispose de sa propre organisation en base de données, avec des tables adaptées à la gestion des réponses et des métadonnées associées. Cette structuration permet plusieurs choses : une recherche rapide et ciblée des réponses; une analyse statistique automatisée grâce aux scripts Pythonne ainsi qu'une visualisation dynamique des données directement sur le site. Pour garantir une consultation efficace des données, des pages PHP dédiées ont été développées afin d'afficher un tableau dynamique contenant l'ensemble des informations extraites et un accès aux analyses statistiques et aux résultats des réponses en fonction des paramètres sélectionnés par l'utilisateur. Cette approche modulaire et automatisée assure une meilleure traçabilité des données et facilite leur exploitation ultérieure, notamment pour les analyses qualitatives et quantitatives des retours des élèves.

## Design et ergonomie du site

Au fil des tests, il est apparu que les différentes pages du site, bien que fonctionnelles, n'étaient pas totalement cohérentes d'un point de vue esthétique. L'une des principales difficultés était de s'assurer que chaque nouvelle page intégrée au projet respecte les codes visuels de l'ensemble du site tout en permettant d'y afficher des données complexes. Un autre problème récurrent était l'affichage du menu latéral (sidebar). En ajoutant une gestion explicite via JavaScript, nous avons pu assurer que le menu s'ouvrait immédiatement au premier clic. L'intégration d'une effet de vague en bas des pages a aussi nécessité des ajustements pour éviter que celle-ci ne recouvre des éléments interactifs comme les liens du pied de page.

## Contraintes et difficultés techniques

Un autre défi rencontré était lié au déploiement des fichiers sur le serveur via FileZilla et GitHub. Afin de réduire la charge des mises à jour, nous avons choisi de garder les styles CSS directement intégrés dans les pages PHP plutôt que d'externaliser une feuille de style unique. Cette approche, bien que nécessitant un peu plus de maintenance, simplifie le processus de mise à jour, car seules les pages modifiées doivent être envoyées sur le serveur et permet de personnaliser chaque page selon son contenu sans multiplier les fichiers inutilement (le code reste très claire à modifier grâce aux commentaires et à la guide utilisateur). Enfin, les fichiers XML utilisés pour stocker les réponses des élèves n'avaient pas de validation DTD, ce qui pourrait poser problème en cas d'incohérence dans la structure des données. Une amélioration future consisterait à ajouter une validation systématique des fichiers XML avant leur exploitation, afin d'éviter des erreurs d'analyse dues à des entrées mal formées.

## En futur

Le projet ARGILES a considérablement évolué au fil des semaines, grâce à notre intégration fluide entre PHP, Python et SQL, permettant une analyse approfondie et facilement accessible aux perceptions littéraires des élèves. Le site est maintenant fonctionnel et ergonomique, offrant une interface intuitive pour la navigation et la consultation des données. Toutefois, plusieurs améliorations restent envisageables :

- ❖ Affiner la mise en page de chaque page.
- ❖ Développer une gestion avancée des utilisateurs, avec la possibilité d'ajouter plusieurs niveaux d'accès (administrateurs, enseignants, chercheurs).
- ❖ Ajouter un contrôle du format des xml plus spécifique.
- ❖ Ajouter des analyses NLP plus précises selon différents projets de recherche.
- ❖ Ajouter un design plus puissant pour la visualisation des analyses et des données, par exemple avec un dashboard interactif ou avec des outils comme Chart.js (bibliothèque JavaScript open-source).

En tous cas, grâce à notre travail minutieux, la plateforme ARGILES permet aujourd'hui de visualiser, stocker et analyser efficacement les retours des élèves sur leurs lectures. Les améliorations à venir permettront d'affiner l'expérience utilisateur et de garantir un environnement d'analyse toujours plus performant et avec un design encore plus soigné.