Université de Lille - Cité Scientifique
42 Rue Paul Duez, 59000 Lille, France

Rapport de stage

Mise en place d'une application web de test de biais implicites

Lucas SAUVAGE

Niveau et formation : Licence 3 Informatique parcours Info

Dates du stage : Du 05 mai 2025 au 06 juin 2025

Tuteurs : Jean-Cristophe ROUTIER & Stéphanie HEMON

Table des matières

1 Contexte

1.1 Entreprise

Ce projet nous a été demandé par la Faculté des Sciences et Technologies (FST) qui fait partie de l'Université de Lille, secteur Cité Scientifique.

En 1559, l'Église autorise la création d'une faculté, initialement située à Douai. Jusqu'au XIX^{e} siècle la faculté persiste mais sera supprimée et réhabilitée plusieurs fois à la fin du XIX^{e} siècle, jusqu'à la dernière suppression en 1816.

Parallèlement, au début du XVIII^{e} siècle, une école de chirurgie est construite à Lille. Cette école deviendra une école préparatoire de médecine en 1805. Pour compenser la perte de la faculté de Douai, une école centrale sera créée en 1796. De plus, une chaire [1] municipale de sciences est établie en 1817, suivie par une chaire de chimie en 1823.

Finalement, en 1854, un décret fusionne toutes ces facultés, ainsi que l'École des arts industriels et des mines, et crée une faculté commune qui aura Louis Pasteur [2] comme doyen. De plus, l'école préparatoire de médecine deviendra la faculté mixte de médecine et de pharmacie de Lille.

En 1887, on compte 46 chaires réparties entre les facultés des sciences, de médecine, de droit et de lettres. La loi du 10 juillet 1896 rétablit les universités en France, dont désormais l'université de Lille, qui regroupe maintenant toutes les facultés publiques.

Face au nombre d'étudiants qui ne cesse d'augmenter, la création du campus de Cité Scientifique commence en 1964. Aujourd'hui l'Université compte 80 000 étudiants des 200 000 de l'Académie de Lille, pour 8 000 salariés.

1.2 Utilisation

La Qualité de Vie Travail et Vie Étudiante est une sous-section de la FST qui a pour but de développer des actions en direction des étudiants et des personnels pour les accompagner dans leurs études et dans leur travail dans la bienveillance, l'écoute et l'échange. L'un des enjeux majeurs de cette mission est d'améliorer le fonctionnement de la FST et simplifier la vie de chacun.

Dans le cadre du recrutement des futurs collaborateurs de la FST, les représentants de la Qualité de Vie au Travail et Vie Étudiante ont exprimé le souhait de mettre en place un test interne portant sur les biais, et plus spécialement sur le domaine "le genre et les sciences"

1. Poste de professeur titulaire dans l'enseignement supérieur, généralement associé à la création d'un laboratoire ou d'une unité de recherche
2. Scientifique français, chimiste et physicien de formation

2 Réalisation

2.1 Présentation du projet

Pour comprendre le fonctionnement de ce test, prenons comme exemple le test sur le domaine "le genre et les sciences". Lorsque l'utilisateur démarre le test, une interface [1] s'affiche avec un mot au milieu, une catégorie à gauche et une catégorie à droite. Chaque test repose sur 2 paires de catégories (par exemple : "Homme" / "Femme" et "Sciences" / "Lettres"), qui sont présentées 2 à 2 ou combinées selon l'avancée du test.

L'utilisateur doit alors associer le mot à l'une des 2 catégories à l'aide de touches ("E" pour la catégorie de gauche et "I" pour la catégorie de droite) le plus rapidement possible. En analysant les temps de réaction de l'utilisateur lorsqu'il classe des associations socialement cohérentes (par exemple, « Femme » et « Lettres ») par rapport à des associations jugées illogiques (comme « Femme » et « Sciences »), on peut alors déterminer si l'utilisateur est implicitement biaisé, et dans quel sens.

2.2 Développement du projet

2.2.1 Organisation

Dans le cadre du développement de cette application web, le travail a été réparti entre les membres de l'équipe, chacun prenant en charge un domaine spécifique. Une de mes collègues s'est consacrée à la partie front-end, en assurant la mise en forme visuelle à l'aide des feuilles de style CSS et en rédigeant les contenus affichés à l'utilisateur sur les différentes pages de l'application.

Un autre membre de l'équipe a pris en charge la gestion de la base de données. Cette tâche comprenait la création de la structure de la base, l'ajout des données au sein des collections, ainsi que la mise en place des routes permettant l'envoi de requêtes et la communication entre l'application et la base de données.

2.2.2 Développement de la logique principale du test

Pour ma part, je me suis occupé de la mise en place du "jeu", c'est-à-dire la boucle principale. Au tout début, comme la base de données n'était pas encore pleinement implémentée, j'ai choisi de débuter en m'occupant de l'affichage des textes explicatifs introduisant le prochain round, présentés entre chaque séquence de mots [2].

Tout en analysant la forme et le contenu des textes, j'ai pu passer à plusieurs reprises le test

1. Exemple en Annexe (voir Figure 1)
2. Exemple en Annexe (voir Figure 2)

témoin [1], ce qui m'a permis d'observer certains détails, tels qu'un délai entre la disparition d'un mot (lorsqu'il est correctement classé par l'utilisateur) et l'apparition du mot suivant ou bien l'ordre de classification des catégories. Dans notre exemple on commencerait par afficher soit la paire de concepts ("Féminin" et "Masculin") soit la paire d'attributs ("Lettres" et "Sciences"). Ensuite on classe les mots appartenant à l'autre paire. Une fois ces 2 rounds terminés, l'utilisateur doit maintenant classifier pendant 2 rounds une congruence (que ce soit la congruence logique ou illogique). Après on demande à l'utilisateur de classer les attributs, en inversant le sens des catégories. Finalement, l'utilisateur se retrouve face à la congruence qu'il n'a pas rencontrée, et ce pendant 2 rounds.

Bien que la base de données ne soit pas complètement implémentée, nous avions discuté des schémas de la base de données, et j'ai pu m'appuyer dessus afin de commencer à développer le test. J'ai commencé par récupérer les mots dans la base de données, puis les trier selon la catégorie. Après avoir récupéré les mots selon les catégories, il fallait ensuite gérer les entrées de l'utilisateur. C'est-à-dire que, si l'utilisateur appuie sur son clavier ("E" ou "I" dans notre cas), il faut, en fonction de la véracité de sa classification :

1. Si elle est correcte :
   (a) Enregistrer le temps que l'utilisateur a mit pour classifier le mot, en l'arrondissant si nécessaire (Temps : < 300 ms ou > 3000 ms)
   (b) Attendre 400 ms de délai, pour respecter les conditions de l'étude de Greenwald.
   (c) Afficher le mot suivant.
2. Si elle est incorrecte :
   (a) Afficher une croix sur l'écran pour notifier à l'utilisateur qu'il s'est trompé.
   (b) Attendre que l'utilisateur se corrige en pressant l'autre touche.
   (c) Ajouter 600 ms au temps de l'utilisateur, comme recommandé dans l'étude de Greenwald
   (d) Puis retourner au point "S'il est correct" (On suppose que l'utilisateur est suffisamment attentif pour déduire que s'il se trompe sur une question à 2 choix, l'autre choix est forcément correct.)

Finalement, une fois que les temps de l'utilisateur ont été enregistrés, on peut calculer l'intensité du biais de l'utilisateur. On commence par calculer l'écart-type des deux jeux de données : celui des temps de réaction lors de la condition congruente, et celui lors de la condition incongruente.

L'écart-type correspond a "une mesure de la dispersion des valeurs d'un échantillon statistique ou d'une distribution de probabilité" [2] ce qui, dans notre cas, nous permet d'évaluer la variabilité des temps de réaction autour de la moyenne.

1. Site d'Harvard : `https://implicit.harvard.edu/implicit/`
2. `https://fr.wikipedia.org/wiki/Ecart_type`

On calcule ensuite l'écart-type standardisé afin de normaliser la différence moyenne entre les temps de réaction des deux conditions. La formule utilisée est la suivante (On suppose que les deux groupes comparés sont de taille équivalente, ce qui permet une simplification du calcul) :

$$SD_{\text{pooled}} = \sqrt{\frac{s_1^2 + s_2^2}{2}}$$

Le score final est obtenu avec cette formule :

$$dScore = \frac{\overline{X}_{\text{incompatible}} - \overline{X}_{\text{compatible}}}{SD_{\text{pooled}}}$$

Le test ne retournera pas directement le D Score a l'utilisateur car ce score n'a pas de représentation concrète pour ce dernier. À la place, on utilisera la valeur absolue du score, tout en gardant en mémoire vers quelle condition l'utilisateur était biaisé, et on utilisera le tableau ci-dessous pour retourner à l'utilisateur une intensité, ce qui est bien plus parlant pour lui.

Valeur du résultat	**Intensité**
$0 \leq dScore \leq 0{,}15$	Aucun biais
$0{,}15 < dScore \leq 0{,}35$	Faiblement biaisé
$0{,}35 < dScore \leq 0{,}65$	Modérément biaisé
$dScore > 0{,}65$	Fortement biaisé

TABLE 1 – Interprétation des valeurs du D Score

2.2.3 Difficultés rencontrées

La réalisation de ces méthodes n'a pas été la partie difficile. C'est au moment d'intégrer ces méthodes pour réagir aux inputs de l'utilisateur que la tâche s'est compliquée. Bien que les *event listeners* soient assez simples dans leur compréhension, parvenir à déclencher l'exécution des méthodes au moment approprié, tout en les empêchant d'être activées en dehors de ces instants, s'est avéré plus délicat que prévu.

Prenons l'exemple de la fonction qui gère la réponse de l'utilisateur lorsqu'il classifie un mot. Afin de rendre l'expérience la plus agréable possible, il fallait faire en sorte que cette fonction ne s'exécute plus une fois que l'utilisateur a répondu, notamment entre le délai de 400 ms entre chaque mot. Sinon, si la fonction n'était pas bloquée, l'utilisateur aurait pu marteler les touches "E" et "I" et alors classer des mots qui n'avaient pas eu le temps d'apparaître.

De plus, l'utilisation de fonctions asynchrones fut un autre challenge. Comme l'application envoie des requêtes à la base de données, il fallait attendre la réponse avant de procéder au traitement de ces données. Au début du projet, il n'existait que les fonctions qui envoyaient des requêtes qui avaient besoin d'être asynchrones, mais au fur et a mesure que le projet avançait,

ces fonctions ont été intégrées dans d'autres plus globales, et j'ai eu de nombreux moments où des données étaient traitées avant d'avoir été correctement récupérée [1].

3 Conclusion

Ce projet fut une expérience enrichissante dans le domaine du développement web. Ayant déjà acquis les bases durant les 2 dernières année, je pense aujourd'hui que j'ai un niveau correct dans ce domaine. Comme ce projet n'était pas un projet de grande taille, nous avions décidé de ne pas utiliser des bibliothèques, telles que React par exemple. En effet, seul le composant lié à l'affichage des statistiques de l'utilisateur (âge, genre, résultats, etc.) aurait réellement tiré parti du framework. Nous avons essayé de rendre ce projet le plus modulable et extensible. Si le projet devait être amené à évoluer, l'organisation séparée du back-end et du front-end rendrait la tâche beaucoup moins complexe.

Je pense que nous sommes arrivés à la vision de l'application que nous avions au début. Les seules différences résident dans certains détails que nous n'avions pas anticipés, souvent liés à l'expérience utilisateur. Par exemple, le choix des couleurs, la taille de la police, la syntaxe et la formulation des textes etc. De plus, cela m'a fait prendre de bonnes habitudes de programmation tels qu'une documentation complète, un README clair et détaillé.

3.1 Remerciements

Je tiens à remercier tout d'abord mes camarades de projet, Omayma El Kadaoui et Anis Rekima, avec qui j'ai eu le plaisir de travailler durant ce stage. Je remercie également mes tuteurs, Jean-Cristophe Routier et Stéphanie Hemom, pour leur accompagnement, leur disponibilité et leurs conseils tout au long du projet. Enfin, j'adresse mes remerciements à Julie Jacques pour l'attribution du projet de substitution, ainsi qu'à l'Université de Lille et la Faculté des Sciences et Technologie pour m'avoir offert cette opportunité.

1. Comme preuve, voici un tweet que j'ai posté durant le mois de mais : "J'avais un problème de synchronisation dans mon code JS, alors j'ai décidé d'utiliser des fonctions asynchrone. problèmes :(maintenant ... deux J'ai".

4 Annexe

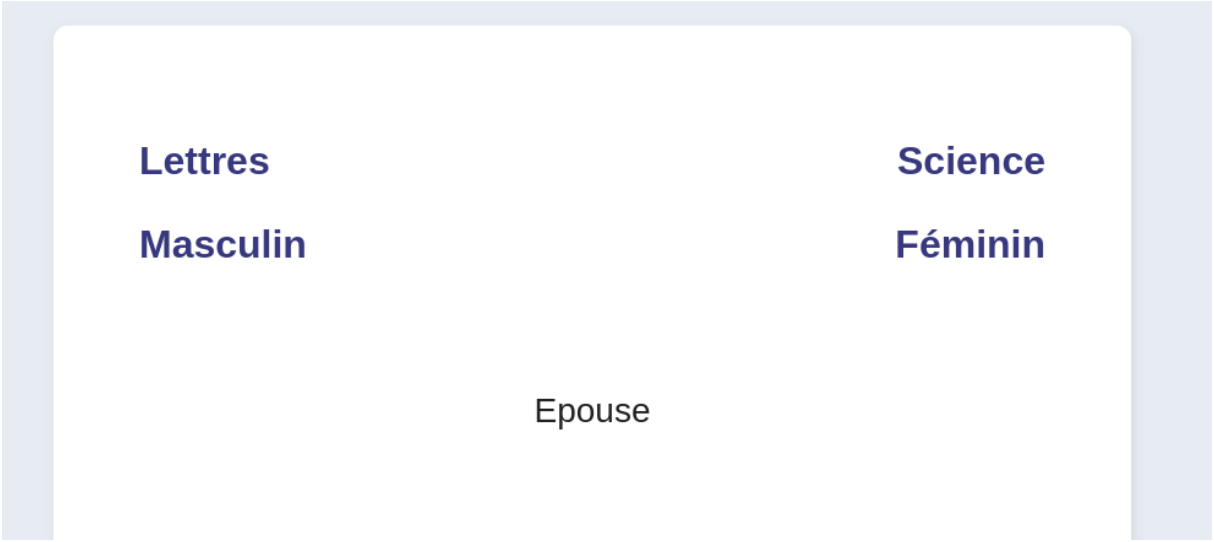

FIGURE 1 – Capture d'écran de l'interface du test

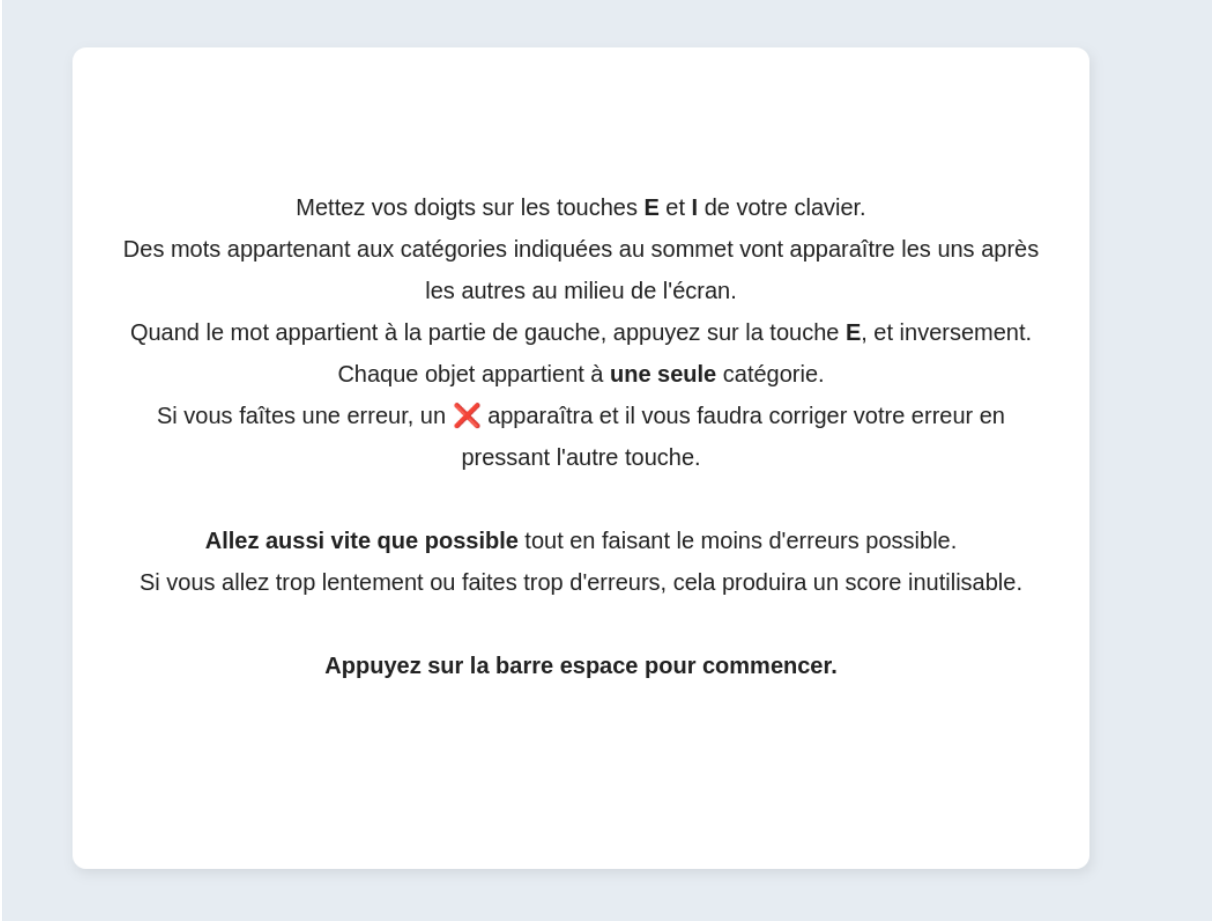

FIGURE 2 – Capture d'écran du texte d'introduction au test