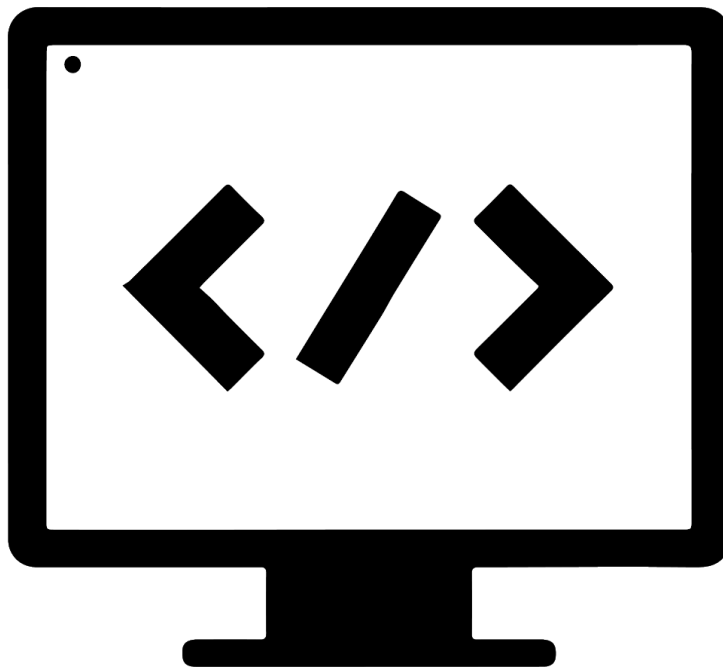

Application web d'auto correction de code SQL



SIRI Romaric

IUT de Saint-Dié-des-Vosges
2019/2020



UNIVERSITÉ
DE LORRAINE



IUT Saint-Dié-des-Vosges

Sommaire

SIRI Romaric	0
Sommaire	1
Introduction	2
Résultats	3
Lignes directrices	3
Fonctionnalités	4
Partie technique	5
Outils utilisés	5
Explication des choix techniques	5
Bibliothèques	5
RegExps	5
Ergonomie	5
Syntaxe du tableau des questions	6
Syntaxe des RegExps en relation avec le code SQL	6
Problèmes rencontrés	7
Regexps	7
Base de donnée / Laravel	7

Introduction

Mon projet consiste à créer une application web qui peut aider les élèves à voir si leur code est correct en relation avec les questions que l'enseignant choisit. Le problème posé vient du fait que corriger les comptes rendus est très répétitif et ce système peut être amélioré grâce à l'intelligence artificielle. Ce projet est très intéressant car maintenant, les machines sont plus performantes que les humains : c'est un gain de temps considérable et une possibilité de découvrir des nouvelles technologies.

Mon projet permet donc aux élèves de pouvoir s'entraîner en autonomie et permet au professeur de gagner du temps non-négligeable.

Résultats

Lignes directrices

12345678910

Donner toutes les informations sur les pilotes

```
1 SELECT *
2 FROM avion
```





Correction

✓ Contient le bon select

✗ Ne contient pas le bon from

⏏ Soumettre

Afficher la correction

	Tout d'abord, l'enseignant saisit dans le fichier de configuration tout ce qu'il veut que l'élève fasse. Ce fichier de configuration contient le code que l'enseignant veut faire apparaître dans le code de l'élève
	L'étudiant copie-coller son code et choisit le numéro de la question sur ma demande.
	Il peut ensuite soumettre son code pour voir où il est correct et où il ne l'est pas.
	Il peut aussi décidé de s'aider de la correction en appuyant sur "afficher la correction".

Fonctionnalités

Les fonctionnalités que j'ai implémentés sont :

- Div html facilement implémentable
- L'étudiant voit où il a bon et où il n'a pas bon
- Fichier de configuration facile à comprendre et facilement extensible
- Snippets pour rajouter des questions avec les expressions qu'on connaît déjà

Les fonctionnalités que je n'ai pas implémentés sont :

- Un système de notation : ça n'avait pas de sens d'en mettre car l'élève peut afficher la correction. Néanmoins les questions contiennent un attribut "poids" qui peut être lu à tout moment si besoin.
- Un système de base de donnée : J'ai eu beaucoup de soucis pour intégrer les regexps dans les bases de données car les regexps Javascript et PHP ne fonctionnent pas pareil.

Partie technique

Outils utilisés

Au cours de la partie programmation, j'ai utilisé plusieurs outils techniques tels que :

- HTML/CSS : pour la page web
- Bootstrap : une collection d'outils pour la création de sites et d'applications web
- jQuery : une bibliothèque JavaScript qui simplifie l'utilisation de JavaScript et la modification du contenu html.
- Javascript : Langage de programmation côté client
- RegExps : Expression régulière, la regex est une séquence de caractères qui définit un modèle de recherche.
- Editeur de texte ACE : Il est open-source et permet la coloration syntaxique en SQL

Explication des choix techniques

Bibliothèques

J'ai décidé d'utiliser ces bibliothèques car elles sont toutes open-source et légères.

RegExps

C'est un outil très puissant et il permet de faire tout ce qu'on veut avec.

Ergonomie

Mon code ansi que la div HTML est facilement implantable sur un autre site. Il y a peu de bouton et c'est difficile de se perdre, l'utilisateur est averti s'il se trompe ou qu'il a pas sélectionné de question.

Syntaxe du tableau des questions

```
1 var questions = [  
2   {  
3     "numero": "1",  
4     "description": "Donner toutes les informations sur les pilotes",  
5  
6     "regles": [{  
7       "nom": "SELECT *",  
8       "regExps": new RegExp(/select\s*/i),  
9       "feedbackVrai": "Contient le bon select",  
10      "feedbackFaux": "Ne contient pas le bon select",  
11      "poids": 1  
12    },  
13    {  
14      "nom": "FROM pil",  
15      "regExps": new RegExp(/from \['\']?pil\['\']?(?!S)/i),  
16      "feedbackVrai": "Contient le bon from",  
17      "feedbackFaux": "Ne contient pas le bon from",  
18      "poids": 1  
19    }  
20  ],  
21 },  
22 ]
```

Voici ci-dessus le fichiers de configuration : questions.js. Il s'agit d'une variable "questions", chaque élément du tableau "questions" correspond à une question sous forme d'objet.

Il contient :

- Des attributs qui le décrivent et qui n'agissent pas sur l'algorithme mais juste sur l'affichage
- Ensuite il y a encore un tableau d'objets "regles" : Il contient toutes les règles que le code de l'élève doit vérifier grâce aux RegExps
- Une règle correspond généralement à un SELECT / FROM / WHERE

Syntaxe des RegExps en relation avec le code SQL

Sur PhpMyAdmin, lorsqu'on fait une requête MySQL il y a des choses qui n'ont pas d'importance, j'ai décidé de transposer ça avec mes RegExps, par exemple :

- Dans le SELECT/FROM, simple ou doubles guillemets ou rien du tout, les attributs peuvent être séparés d'un espace ou non.

Problèmes rencontrés

Regexps

J'ai passé la majorité de mon temps à apprendre les RegExps, elles sont très difficiles et je n'ai pas trouvé toutes les RegExps qui m'intéressaient, par exemple je n'ai pas réussi à faire en sorte que l'ordre des mots n'ont pas d'importance, l'élève doit respecter l'ordre donné dans l'énoncé. Exemple : `SELECT id, nompil` ne peut pas être `SELECT nompil, id`.

J'ai aussi voulu trouver un moyen d'avertir l'élève de manière plus précise sur ce qu'il avait faux. Par exemple "Attention il manque des guillemets" parce que actuellement ça affiche uniquement si le `SELECT/WHERE/FROM` est bon ou faux. Mais ça demande beaucoup trop de règles, le fichiers devient illisible et ça demande beaucoup trop de travail.

Base de donnée / Laravel

Lorsque j'ai essayé d'intégrer les RegExps dans une base de donnée, mon but était de permettre au professeur d'en rajouter plus facilement avec un formulaire qui pourrait gérer automatiquement les RegExps.

J'ai commencé à essayer d'implanter ce système avec Laravel car je voulais m'entraîner sur ce framework, mon stage se déroule entièrement en Laravel du coup c'était l'occasion de s'exercer.

Les RegExps sont un objet propre à Javascript elles sont différentes en PHP et je n'ai pas trouvé de moyen de les convertir.

Conclusion

Pour conclure, je suis satisfait de mon travail mais il mérite d'être largement approfondi, avec par exemple un système de base de donnée et un système où le prof peut entrer des règles sans devoir avoir de connaissances en RegExps et sans aller chercher le fichier de configuration.

Ce projet m'a apporté ce que j'espérais : des connaissances approfondies en Javascript qui m'aideront pour plus tard car j'aimerais continuer dans le web, ainsi que des connaissances en expression régulière qui sont utiles notamment en base de donnée et sur les systèmes Linux.