

Rapport de Projet d'Application 3A STI

Sujet 9: La Plateforme de Correction Automatique de TD/TP



Figure 1 : L'accueil de plateforme CORRECT'INSA-CVL

Membres du groupe :

Nurdini BINTI MOHAMAD

Shérif MAKALOU

Ibtissam JENNATE

Tuteur :

M. BRIFFAUT JEREMY

Sommaire

SOMMAIRE.....	2
Cahier de charges.....	3
1 - Outils utilisés.....	3
2 - Travail réalisé.....	3
3 - Mode opératoire.....	4
4 - Répartition du travail.....	5
5 - Architecture du Projet.....	5
6 - Difficultés rencontrées.....	6
7 - Axes d'amélioration.....	6
Conclusion.....	8

Cahier de charges

L'objectif de ce projet était de proposer une application web capable de corriger automatiquement les TD/TP soumis par les élèves. Il s'agit notamment des TP Système, TP réseau et TD Gentoo effectués en 3A STI.

Le professeur pouvait donc déposer un sujet et sa correction, et les élèves pouvaient par la suite déposer leur devoir afin que ces derniers soient corrigés automatiquement. Les élèves reçoivent ensuite automatiquement leur note après soumission.

On avait le choix entre ajouter des fonctionnalités supplémentaires à Ubiquity (une application web Open Source de suivi de TP informatiques développée par des étudiants de l'INSA Rouen) pour permettre la correction automatique des rendus soumis, ou bien partir de zéro en développant notre propre application web qui se chargera de faire ce travail.

Suite à des problèmes d'installation de la partie serveur d'Ubiquity, nous nous sommes finalement tournés vers la deuxième option et avons développé notre propre application.

1 - Outils utilisés

Pour mener à bien ce projet, nous avons utilisé principalement le **langage Python** pour les scripts du back-end qui se chargent de la correction automatique, en plus de quelques de ses librairies, et la **framework Django** qui nous a permis de créer l'interface utilisateur et la gestion de la base de données. Quelques commandes **C** ont aussi été utilisées mais elles représentent une infime part des outils utilisés pour le développement du projet.

2 - Travail réalisé

Nous avons mis en œuvre une plateforme en ligne novatrice, se concentrant spécifiquement sur la correction automatique des travaux pratiques (TP) des travaux dirigés (TD), en particulier ceux liés aux TP système et à TD Gentoo. Ce site web a été conçu et développé en utilisant des technologies modernes, avec un accent sur l'efficacité, l'accessibilité et la facilité d'utilisation.

Caractéristiques dédiées aux Étudiants

Après s'être connectés à leurs comptes individuels, les étudiants ont accès à une multitude de sujets de TP et TD. Ils peuvent consulter ces sujets, et soumettre leurs travaux directement sur la plateforme. La particularité de ce système est qu'une fois les devoirs soumis, une correction automatique est effectuée. Les notes sont ensuite disponibles immédiatement pour consultation, permettant aux étudiants de recevoir des retours instantanés sur leurs performances.

Caractéristiques dédiées aux Enseignants

Les enseignants, après s'être connectés à leurs comptes, ont la capacité de gérer les sujets qu'ils ont soumis, d'ajouter de nouveaux sujets avec leurs solutions respectives préprogrammées pour une correction automatique. Cela permet une évaluation précise, rapide et sans intervention manuelle. En outre, ils peuvent consulter les notes des étudiants qui ont soumis leurs devoirs, ainsi que visualiser des graphiques statistiques. Ces graphiques donnent un aperçu clair des performances générales des étudiants, des tendances de notes, et peuvent aider à identifier les domaines qui nécessitent une attention ou une amélioration supplémentaire.

3 - Mode opératoire

- **TP Système:** L'enseignant et l'étudiant déposent tous les deux un fichier zip contenant à la fois les fonctions implémentées et un fichier `main.c` permettant de faire des tests de fonctions et affichant les résultats obtenus pour chaque appel de fonction.

Le fichier de l'enseignant correspond bien-sûr à la correction qui servira de modèle dans notre système pour la correction.

1ère étape:

Notre back-end va tout d'abord dézipper le fichier de l'enseignant et le compiler par la suite avant de l'exécuter. L'exécution va renvoyer en sortie une série de résultats qui seront enregistrés ligne par ligne dans un fichier `.txt`.

2ème étape:

Une fois que l'élève a soumis son fichier `.zip`, il sera aussi dézippé, mais compilé et exécuté **avec le `main.c` de l'enseignant** afin d'obtenir les sorties et les enregistrer sous le même format dans un fichier `.txt`.

3ème étape:

On dispose maintenant de deux fichiers `.txt`, il suffit de les comparer ligne par ligne et de noter les différences. La note de l'élève est obtenue avec le calcul:

$(\text{nombre de lignes identiques} / \text{nombre total de lignes}) * 20$

- **TD Gentoo:** Dans notre implémentation, le professeur n'a pas besoin de soumettre la correction, le sujet étant toujours le même, les données de correction ont été directement incluses dans le système de correction automatique. L'élève doit soumettre son fichier `.tar.bz2` qui sera dézippé et dans lequel se trouve un fichier **`evaluation.txt`** (obtenu à l'aide d'un script complexe du professeur) contenant un résumé des configurations effectuées. Le contenu du fichier `evaluation.txt` est ensuite tout simplement comparé aux configurations attendues. La note est attribuée suivant un système de pondération prédéfini dans l'implémentation.
- **TP Réseau:** Cette partie n'a pas été totalement implémentée mais nous avons quelques résultats.

Nous avons mis au point un script Python qui lance efficacement des processus serveur et clients en simultané, démontrant une gestion du multithreading réussie. Néanmoins, nous avons constaté des problèmes de redirection des sorties de commandes, potentiellement dus à une mauvaise gestion des flux de données ou à une synchronisation inappropriée entre les threads. Par ailleurs, certains processus semblent bloquer d'autres, ajoutant de la complexité au problème.

4 - Répartition du travail

Nurdini BINTI MOHAMAD: Elle était responsable du développement d'un système de correction automatique pour les travaux pratiques. Elle a conçu un code permettant la compilation du code source de l'enseignant et des étudiants, ainsi que son exécution. Elle a établi une liaison entre les codes soumis par les étudiants et l'enseignant pour garantir une compilation et une évaluation réussies à partir de ces données. Elle a également réalisé plusieurs séries de tests pour valider cette fonctionnalité, et ceci, pour tous les travaux pratiques étudiés.

Shérif MAKALOU: Il s'est principalement occupé du développement de l'interface utilisateur pour les professeurs et étudiants (gestion des connexions, de l'inscription sur la plateforme, options de dépôt de sujet ou de devoir, statistiques etc), et il s'est aussi chargé de la structure et de la gestion de la base de données, et donc des requêtes nécessaires pour répondre au besoin de l'utilisateur. Il a aussi participé à la gestion des scripts et a mis en place la structure globale du code en utilisant le principe du Model-View-Template(MVT) de Django.

Ibtissam JENNATE: Elle a principalement œuvré sur le backend du projet, élaborant un script pour la correction automatique du TD Gentoo et initiant le développement d'un script distinct pour le TP réseau, qui est encore en cours de finalisation. En plus de ses tâches backend, elle a su apporter sa contribution au frontend, en mettant ses compétences en CSS à profit pour améliorer l'esthétique du projet. Elle a tenté d'installer Ubiquity, mais des complications techniques ont entravé la réussite de cette tâche.

5 - Architecture du Projet

Nous avons utilisé le Framework Django de python qui nous a permis de créer toute l'interface utilisateur et la gestion de la base de données. Django utilise des parties appelées "applications" et une architecture MVT (Model View Template) qui permet un développement bien structuré. Chaque application dispose de ses propres modèles, vues et templates. Dans notre projet nous en avons deux: une application

“**authentication**” qui regroupe toute la partie du développement qui concerne la connexion des utilisateurs, leur inscription, leurs droits d’accès etc, et une application “**blog**” qui gèrait tout le contenu disponible après authentification d’un utilisateur.

Architecture MVT(Model View Template):



Modèle: C’est la partie qui nous permet de construire notre base de données. Django utilise de la programmation orientée objet (POO) pour la structuration des base de données. Chaque nouvelle table est une classe et chaque colonne est définie par un attribut de classe. Chaque ligne ou entrée d’une table est une instance de classe.

Vue: La vue se charge de récupérer, traiter les données du modèle en fonction du besoin de l’utilisateur, et de les transférer par la suite au template. C’est dans cette partie que sont importées les différentes fonctions des scripts Python créées pour la correction automatique.

Template: Il s’agit de la part de l’architecture qui se charge des rendus, donc du visuel du site, elle est généralement composée de pages HTML qui peuvent recevoir les données traitées par la vue et les afficher selon le style ou l’organisation voulu.

6 - Difficultés rencontrées

Nous avons fait face à certaines difficultés durant ce projet qui ont eu des impacts sur le respect du cahiers des charges.

Contrainte temporelle: On disposait d’une semaine pour faire ce projet, et avec notre effectif, on a dû travailler pendant plusieurs heures supplémentaires pour pouvoir faire le maximum de travail possible. Malgré cela, on n’a pas pu implémenter entièrement la partie concernant le TP Réseau, et on n’a pas non plus peaufiner certaines fonctionnalités afin de rendre l’application plus efficace et plus sécurisée.

Installation d’Ubiquity: Comme cela a été mentionné plus tôt, nous avons eu des problèmes pour installer Ubiquity. C’est un problème de CAS (Central Authentication Service) qui s’est posé durant l’installation du logiciel.

Il aurait été peut-être plus simple de pouvoir utiliser cette plateforme en y incluant des fonctionnalités supplémentaires. Cela aurait pu aussi être un gain de temps énorme étant donné que toute l'interface aurait déjà été à notre disposition.

Test du TP Réseau: En plus de la contrainte temporelle mentionnée plus haut, il nous a été difficile de trouver un moyen de tester ce TP. Il inclut un fichier serveur et un fichier client et il fallait trouver un moyen de tester la connexion entre plusieurs clients et le serveur, en plus des fonctionnalités telles que l'envoi des messages, le changement de pseudos etc.

7 - Axes d'amélioration

Après analyse du travail fourni, nous avons étudié quelques améliorations qui pourraient être utiles et rendre l'application plus intéressante:

Téléchargement des fichiers soumis: Pour l'instant nous n'avons pas réussi à mettre à disposition des élèves les sujets soumis par les professeurs. Il aurait été intéressant que les élèves puissent directement télécharger les sujets à partir de la plateforme.

Correction complète des sujets TP et réseau: Pour l'instant, nos scripts n'arrivent pas à corriger entièrement le TP Système. Le TP Système est long et comporte plusieurs fichiers dépendants les uns des autres, d'autant plus que certaines fonctions de la glibc ne sont pas simples à tester.

Fichier de configuration pour Gentoo: Les données nécessaires à la correction du TP d'installation de Gentoo ont directement été intégrées au script de correction. Cependant cela pourrait être un inconvénient si jamais le prof aimerait changer quelques paramètres d'installation. Il est alors important de pouvoir mettre à la disposition du professeur un fichier de configuration sur lequel il pourra changer les différents paramètres à sa guise.

Conclusion

En définitive, nous avons réussi à répondre à une bonne partie des besoins du cahier des charges. Les difficultés citées tantôt nous ont retardé, mais avec un peu plus de temps, nous aurions pu tout implémenter car la plus grosse partie du travail a déjà été faite.

Nous sommes plutôt fiers du travail accompli, surtout avec notre faible effectif et le peu de temps dont nous disposions. Ce projet a été très formateur, il nous a non seulement permis de mettre en pratique les connaissances acquises au cours de cette troisième année (développement Python, C, développement Web, gestion de bases de données SQL, scripting), mais aussi de découvrir et de nous familiariser avec de nouveaux outils de développement comme Django et des librairies python comme Plotly pour les données statistiques.

Le projet est disponible en Open Source sur la branche main de notre github, voici le lien:

<https://github.com/Wolfidy7/Projet-application>

Nous tenons à remercier M.Briffaut Jeremy, pour son accompagnement et ses conseils durant cette semaine de projet.