TP02 – Application d'aide à la compréhension de la lecture (*back-end*).

Technologie de l'inforoute (INF37407)
Automne 2022
yacine_yaddaden@uqar.ca

Table des matières

1 Objectifs du travail

L'objectif principal du travail est de mettre en application les connaissances acquises durant le cours, à savoir la mise en place de partie serveur (*back-end*). Pour cela, vous utiliserez un des Frameworks les plus populaires pour le *back-end* à savoir **Django**. Il sera plus précisément question de créer une **API REST** donnant accès à certaines données et services.

L'**API REST** à concevoir sera utilisée dans le cadre d'un projet de recherche visant à favoriser la compréhension de lecture d'élèves ayant une déficience intellectuelle. Les détails vous seront présentée par la professeure **Edith Jolicoeur** de l'unité départementale des sciences de l'éducation.

2 Technologies à utiliser

Dans le cadre de ce travail pratique, vous serez amener à utiliser les technologies suivantes :

- ✓ **Microsoft Visual Studio Code** : https://code.visualstudio.com/
- ✓ Interpréteur **Python** : https://www.python.org/
  - — `venv` pour la création de l'environnement virtuel.
  - — `pip` pour la gestion des paquets.
- ✓ Pour la conception de l'**API REST**, vous aurez besoin de :
  - — Django (*Framework principal*) : https://www.djangoproject.com/
  - — Django REST Framework : https://www.django-rest-framework.org/
  - — SWAGGER (*tester l'***API REST**) : https://swagger.io/

Pour le langage utilisé, ça sera principalement :

- ✓ **Python**
- ✓ **JavaScript/React.js** (*intégration avec l'application*)

IMPORTANT : *Vous pouvez utiliser d'autres paquets que vous juger pertinents pour la réalisation du projet.*

3 Description détaillée

Dans le cadre de cette partie du projet projet, il sera question de manipuler différentes *entités* des données et fournir un certain nombre de services/fonctionnalités à travers une **API REST**.

3.1 Services & fonctionnalités

Parmi les services à implémenter, on peut citer les suivants :

3.1.1 Inscription

Chaque participant a la possibilité de créer un compte et pour cela, il doit fournir les différentes informations (*voir l'énoncé du premier travail pratique*).

→ Requête de type : **POST** (*avec les données dans le corps de la requête*).

3.1.2 Authentification

Pour la partie authentification, ça se fera en utilisant le système de *tokens* ou *jeton*. Les deux entités *administrateur* et *participants* sont concernés par l'authentification.

- Il n'y a qu'un seul *administrateur* qui est créé par défaut,
- Les participants se créent en fur et à mesure des inscriptions.

→ Requête de type : **POST** (*avec les données dans le corps de la requête*).

3.1.3 Conversion du texte en son

Pour la conversion du texte en son, on utilisera le paquet : **gTTS**. Il est possible de l'installer avec la commande : `pip install -U gTTS`.

Un exemple de code simple pour la conversion d'un texte en son (fichier `.mp3`) :

```python
from gtts import gTTS

# Déclarer la phrase à convertir
input_text = "Bonjour à tous !"
# Effectuer la conversion
tts = gTTS(input_text, lang='fr')
# Sauvegarder le fichier généré en .mp3
tts.save("salutation.mp3")
```

3.1.4 SWAGGER

Afin de pouvoir tester l'**API REST** directement sur le navigateur, on utilisera **SWAGGER** qu'il faudra configurer avec **Django**. Il faudra installer le paquet **drf-yasg** avec la commande : `pip install -U drf-yasg`

3.2 Manipulation des entités de données

Pour cette partie là, il y a différentes entités de données à manipuler :

3.2.1 Participants

En plus de pouvoir créer un nouveau compte avec ses informations, le participant a la possibilité de visualiser ses données et les modifier.

Il a la possibilité de :

- Manipuler l'information de son propre compte participant :
  → **GET**, **POST** et **PUT/PATCH**.
- Accès aux les textes et les Quiz :
  → **GET** et **POST** (*réponses aux Quiz*).
- Visualiser ses propres statistiques :
  → **GET**.

IMPORTANT : *La suppression d'un compte utilisateur ne peut être effectué que part l'administrateur.*

3.2.2 Administrateur

L'administrateur peut visualiser ses informations et les modifier, mais il ne peut pas supprimer son compte.

→ Requête de type : **GET** et **PUT/PATCH**.

Il a la possibilité de :

- Manipuler l'information des comptes de participant :
  → **GET**, **PUT/PATCH** et **DELETE**.
- Manipuler les textes et les Quiz :
  → **GET**, **PUT/PATCH**, **POST** et **DELETE**.
- Visualiser les statistiques par participant :
  → **GET**.

3.2.3 Textes

Les textes sont constitués d'un titre et de plusieurs phrases. À chacun est associé un fichier `.mp3` généré automatiquement lors de l'ajout.

Au niveau des phrases, il y a certains mots auxquels sont associées des images qui doivent être téléverser sur le serveur.

3.2.4 Quiz

Chaque Quiz est constitué d'un certain nombre de questions et chacune d'elles est constituée de plusieurs réponses possibles (*une seule est considérée comme correcte*).

Alors que l'administrateur peut manipuler l'ensemble des Quiz, le participant ne peut y avoir accès qu'en lecture et a la possibilité de répondre aux questions du Quiz.

3.3 Déploiement en ligne – BONUS

Afin d'avoir des points supplémentaires, il vous est demandé de déployer l'**API REST** sur un site d'hébergement pour **Python** gratuit à l'adresse :

→ https://www.pythonanywhere.com

En ce qui concerne l'application **React.js** (front-end), vous pouvez la déployer sur la plateforme **Vercel** à l'adresse :

→ https://vercel.com/

Vous trouverez sur le portail Moodle une vidéo de démonstration sur les différentes étapes à suivre pour le déploiement.

4 Dépôt GitHub

Afin de faciliter le travail d'équipe, il est recommandé d'utiliser un outil de gestion de version centralisé. Le choix le plus adéquat est : **GitHub** [1].

Vous pouvez utiliser le même dépôt *privé* et partager l'accès entre les membres de l'équipe.

5 Modalité d'évaluation

Le travail pratique comptera pour 25% (*points*) de la note du cours. Ces points sont répartis de la manière suivante :

1. https://github.com/

Partie	Pourcentage
Services & fonctionnalités	**9.0%**
→ Inscription	2.0%
→ Authentification	3.0%
→ Conversion du texte en son	2.0%
→ SWAGGER	2.0%
Manipulation des entités de données	**10.5%**
→ Participants	2.0%
→ administrateur	2.0%
→ Textes	3.0%
→ Quiz	3.5%
Intégration avec le *front-end*	**2.5%**
Présentation du travail effectué	**3.0%**
→ Rapport (avec qualité du Français)	1.5%
→ Vidéo de démonstration	1.5%
Déploiement ne ligne – BONUS	**2.0%**

6 Date de remise

→ Date limite pour la remise : **20 Décembre 2022 à 23h00**
→ Les fichiers à remettre sur le portail Moodle :
- ✓ Le code source dans un fichier compressé :
  - Le nom du fichier → `code_source.zip`.
- ✓ Un rapport (*Le modèle à utiliser est fourni*) :
  - Le nom du fichier → `rapport.docx` ou `rapport.pdf`.

7 Points importants

- **Note I :**
  1. Le travail est à réaliser en équipe :
     → Vous devez m'envoyer la liste des équipes par e-mail.
  2. Le professeur peut poser des questions liées au travail pratique,
  3. Le non-respect de l'énoncé peut occasionner une perte de points,
  4. En cas de plagiat, le ou les étudiants seront sanctionnés :
     → Voir la politique de l'université.
  5. Le retard de remise peut occasionner une perte de points.
  6. Dans le rapport, vous devez indiquer :

→ Les parties dont s'est occupé chacun des membres de l'équipe.

– **Note II :** *Dans le cas où il y a des aspects qui ne sont pas clairs, n'hésitez pas à m'en faire part afin que je puisse apporter des éclaircissements et éventuellement mettre à jour l'énoncé du travail pratique.*

8 Bibliographie

1. Samson, P. (2020). DJANGO Développez vos applications web en Python. Édition ENI.
2. Melé, A. (2020). Django 3 By Example : Build powerful and reliable Python web applications from scratch. Packt Publishing Ltd.
3. Bendoraitis, A & Kronika, J. (2020). Django 3 Web Development Cookbook : Actionable solutions to common problems in Python web development. Packt Publishing Ltd.
4. Hillar, G. C. (2018). Django RESTful Web Services : The Easiest Way to Build Python RESTful APIs and Web Services with Django. Packt Publishing Ltd.