



W1 - Python

W-WEB-350

W4 - Python

my_driving_school

W4 - Python

delivery method: drivingschool on Github
build tool: no need here



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

COMPÉTENCES à ACQUÉRIR

- Framework Django
- Base de données
- Routes
- Python
- Architecture



INTRODUCTION

Votre mission, si vous l'acceptez (mais en même temps, avez-vous vraiment le choix ? :p), est de réaliser un intranet de gestion d'auto-école avec le Framework Django.

RESTRICTIONS

- Le projet doit être réalisé avec le Framework « Django ».
- Les fonctionnalités suivantes de Django devront être utilisées (soyez pertinents) :
 - Base de données SQLite
 - Données initiales pour les modèles `#fixtures`
 - Routes
 - Vues génériques
 - Form class
 - Moteur de template Django (obligation de mettre un block dans la balise title au minimum)

PROJET

Le projet est un intranet pour une auto-école qui permettra à l'école de gérer ses élèves, ses plannings, et ses instructeurs. De plus, il permet aux élèves de consulter leur avancement dans leur formation et de voir quand seront leurs prochains rendez-vous.

Les étudiants devront passer par une des secrétaires de l'auto-école pour obtenir un compte et acheter un forfait d'heure de leçons. Ils pourront ensuite prendre rendez-vous soit avec une secrétaire, soit directement avec un instructeur.

L'intranet devra empêcher la prise de rendez-vous si l'étudiant ne dispose plus d'assez d'heures, il devra alors repasser par une secrétaire afin d'acheter des heures supplémentaires pour pouvoir prendre un autre rendez-vous.

L'intranet possédera 4 types de comptes différents : **Student**, **Instructor**, **Secretary**, et **Admin**.

Les comptes **Student** permettront aux étudiants de consulter les informations relatives à leurs formations :

Un planning avec leurs différents rendez-vous (heure, date, lieux, instructeur)

L'avancement de leurs forfaits (le nombre d'heures qu'ils ont payé / le nombre d'heures de leçon prise)

Les comptes **Instructor** donneront la possibilité aux instructeurs de :

- Consulter leur planning de rendez-vous (heure, date, lieux, élève)
- Consulter les fiches de leurs élèves qui comprendront toutes les informations associées à un élève
- Ajouter, modifier ou supprimer un rendez-vous avec un étudiant dans la limite des heures disponibles de l'étudiant

Les comptes **Secretary** devront pouvoir :

- Créer, modifier ou supprimer des comptes **Student** ou **Instructor**
- Consulter les fiches de tous les élèves
- Ajouter des heures de cours aux étudiants
- Ajouter, modifier ou supprimer un rendez-vous entre un étudiant et un instructeur
- Consulter le planning de chaque élève ou instructeur individuellement
- Consulter un planning général

Enfin les comptes **Admin** disposeront des mêmes droits que les comptes **Secretary** en plus de la possibilité de gérer les comptes **Secretary**.



BONUS

- Création d'un espace permettant aux étudiants de s'entraîner pour l'examen du code
 - Permettre aux **Instructor** de créer des séries de questions
- Permettre à un **Student** d'envoyer une demande de rendez-vous à un **Instructor** qui pourra accepter, refuser ou faire une autre proposition au **Student** qui pourra à son tour accepter, refuser ou faire une autre proposition. Jusqu'à ce qu'ils tombent d'accord sur un créneau ou que la demande soit refusée
- Un système d'achat via l'intranet (paypal, carte bleue, ...) d'heure de leçons pour les **Student**