



## LOG3300 Développement d'applications web (3 crédits)

### Devoir #3

Hiver 2024

### Consignes générales

- Ce travail se fait en **GROUPE DE 2**
- Le travail doit être remis sur Moodle AU PLUS TARD LE 6 Avril 2024. Pour cela, il faut rentrer sur le site Moodle du cours et déposer le fichier en utilisant l'option « Remise du travail ».
- Ce travail a une pondération de **20 points** et comptera pour **10 %** de la note finale du cours.
- Le plagiat est formellement interdit et tout étudiant qui y est impliqué est passible de renvoi de l'ISTEAH (<http://www.isteah.edu.ht/pdf/code-de-conduite.pdf>).

ÉTUDIANT 1 JOSEPH Samuel Jonathan

ÉTUDIANT 2 .....

---

**Le Devoir 3 est basé sur le Framework Django - Gestion d'employés et de départements**

***Veuillez lire attentivement toutes les instructions avant de commencer le devoir.***

#### **Objectifs :**

- Ce devoir est conçu pour évaluer votre compétence dans le développement d'une application Web utilisant le framework Django

#### **Critères d'évaluation :**

- Votre projet sera évalué en fonction de sa fonctionnalité, de son ergonomie, de la qualité de son code, de son respect des bonnes pratiques de développement Django, ainsi que de la présentation de votre vidéo de démonstration. Assurez-vous de documenter adéquatement votre code tout au long du processus. Bonne chance!

*Veuillez suivre attentivement les instructions pour chaque tâche.*

## 1. (1 point) Conception de la base de données.

- ✚ Vous êtes chargé de concevoir une base de données adéquate pour stocker les informations sur les employés et les départements. Assurez-vous que la structure de la base de données est logique et efficace.

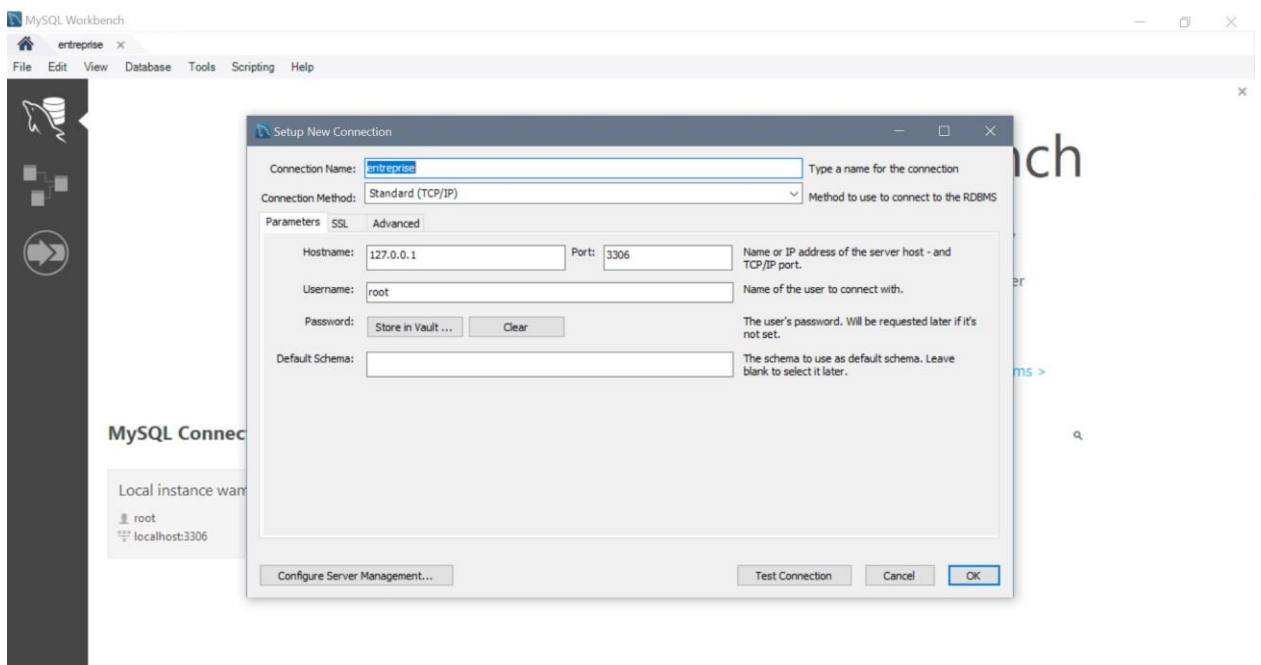
The first screenshot shows the MySQL Workbench interface with the 'enterprise' schema selected. The SQL editor contains the following code:

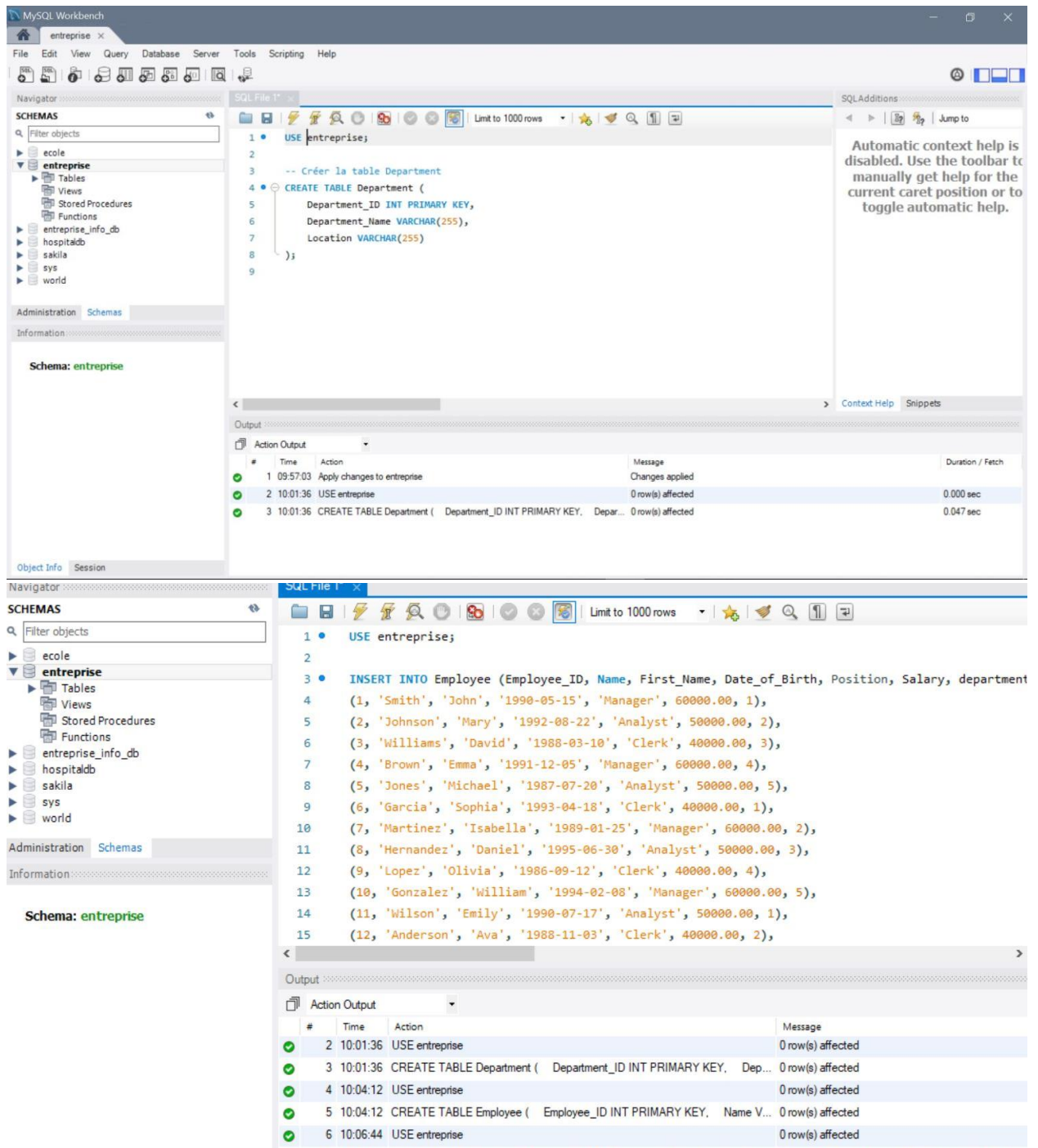
```
1 USE enterprise;
2
3 INSERT INTO Department (Department_ID, Department_Name, Location) VALUES
4 (1, 'HR', 'New York'),
5 (2, 'IT', 'Seattle'),
6 (3, 'Sales', 'Chicago'),
7 (4, 'Finance', 'Los Angeles'),
8 (5, 'Operations', 'Houston');
```

The second screenshot shows the same interface with the 'enterprise' schema selected. The SQL editor contains the following code:

```
1 USE enterprise;
2
3 CREATE TABLE Employee (
4     Employee_ID INT PRIMARY KEY,
5     Name VARCHAR(255),
6     First_Name VARCHAR(255),
7     Date_of_Birth DATE,
8     Position VARCHAR(255),
9     Salary DECIMAL(10, 2),
10    departmentid INT, -- Renommée de la colonne
11    FOREIGN KEY (departmentid) REFERENCES Department(Department_ID)
12 );
```

Both screenshots show the 'Output' window at the bottom, displaying the results of the SQL execution. The first screenshot shows the output for the INSERT statement, and the second screenshot shows the output for the CREATE TABLE statement.





## 2. (1 point) Génération du projet Django et création d'une application pour la gestion des employés.

- ✚ Créez un nouveau projet Django en utilisant la commande appropriée.
- ✚ Développez une nouvelle application Django dédiée à la gestion des employés au sein d'une organisation.

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300> git init
PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300> git remote add origin https://github.com/josephsamijona/LOG3300-3.git
PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300> git pull origin main
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (23/23), done.
Unpacking objects: 100% (25/25), 13.49 KiB | 151.00 KiB/s, done.
From https://github.com/josephsamijona/LOG3300-3
* [new branch]      main       -> origin/main
PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300> django-admin startproject gestion_employes
PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300> cd gestion_employes
PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300\gestion_employes> python manage.py startapp employees
PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300\gestion_employes> cd ..
PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300> git commit -m "first commit"
[master 4da6da8] first commit
15 files changed, 217 insertions(+)
create mode 100644 gestion_employes/employees/__init__.py
create mode 100644 gestion_employes/employees/admin.py
create mode 100644 gestion_employes/employees/apps.py
create mode 100644 gestion_employes/employees/migrations/__init__.py
create mode 100644 gestion_employes/employees/models.py
create mode 100644 gestion_employes/employees/tests.py
create mode 100644 gestion_employes/employees/views.py
create mode 100644 gestion_employes/gestion_employes/__pycache__/__init__.cpython-310.pyc
create mode 100644 gestion_employes/gestion_employes/__pycache__/settings.cpython-310.pyc
create mode 100644 gestion_employes/gestion_employes/settings.py
create mode 100644 gestion_employes/gestion_employes/urls.py
create mode 100644 gestion_employes/gestion_employes/wsgi.py
create mode 100644 gestion_employes/manage.py
PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300> git push origin main

```

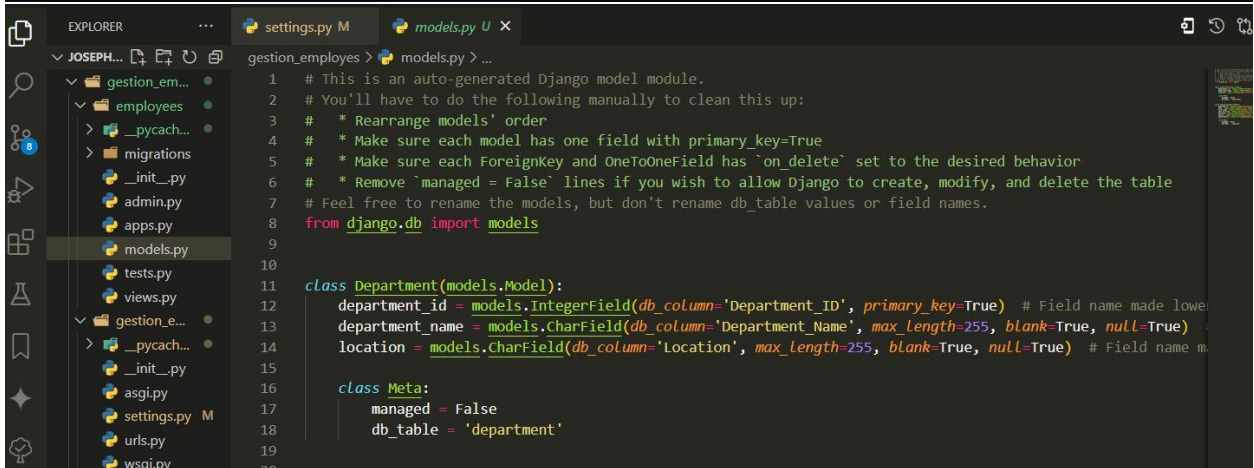
### 3. (2 points) Modélisation des employés.

- Élaborez un modèle pour enregistrer les informations sur les employés, en incluant tous les champs pertinents.

```

PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300> cd gestion_employes
PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300\gestion_employes> python manage.py inspectdb > models.py
PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300\gestion_employes> |

```



```

1 # This is an auto-generated Django model module.
2 # You'll have to do the following manually to clean this up:
3 # * Rearrange models' order
4 # * Make sure each model has one field with primary_key=True
5 # * Make sure each ForeignKey and OneToOneField has `on_delete` set to the desired behavior
6 # * Remove `managed = False` lines if you wish to allow Django to create, modify, and delete the table
7 # Feel free to rename the models, but don't rename db_table values or field names.
8 from django.db import models
9
10
11 class Department(models.Model):
12     department_id = models.IntegerField(db_column='Department_ID', primary_key=True) # Field name made lower
13     department_name = models.CharField(db_column='Department_Name', max_length=255, blank=True, null=True) # Field name made lower
14     location = models.CharField(db_column='Location', max_length=255, blank=True, null=True) # Field name made lower
15
16     class Meta:
17         managed = False
18         db_table = 'department'
19
20

```

### 4. (2 points) Modélisation des départements.

- Créez un modèle pour enregistrer les informations sur les départements, en établissant une relation appropriée avec les employés.



```

20
21 class Employee(models.Model):
22     employee_id = models.IntegerField(db_column='Employee_ID', primary_key=True) # Field name made lowercas
23     name = models.CharField(db_column='Name', max_length=255, blank=True, null=True) # Field name made lower
24     first_name = models.CharField(db_column='First Name', max_length=255, blank=True, null=True) # Field na
25     date_of_birth = models.DateField(db_column='Date of Birth', blank=True, null=True) # Field name made lo
26     position = models.CharField(db_column='Position', max_length=255, blank=True, null=True) # Field name m
27     salary = models.DecimalField(db_column='Salary', max_digits=10, decimal_places=2, blank=True, null=True)
28     departmentid = models.ForeignKey(Department, models.DO_NOTHING, db_column='departmentid', blank=True, nu
29
30     class Meta:
31         managed = False
32         db_table = 'employee'
33

```

## 5. (1 point) Génération des migrations.

- ✚ Générez les fichiers de migration pour vos modèles afin d'appliquer les changements à la base de données.

```

PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300\gestion_employes> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300\gestion_employes>

```

## 6. (1 point) Migration des données.

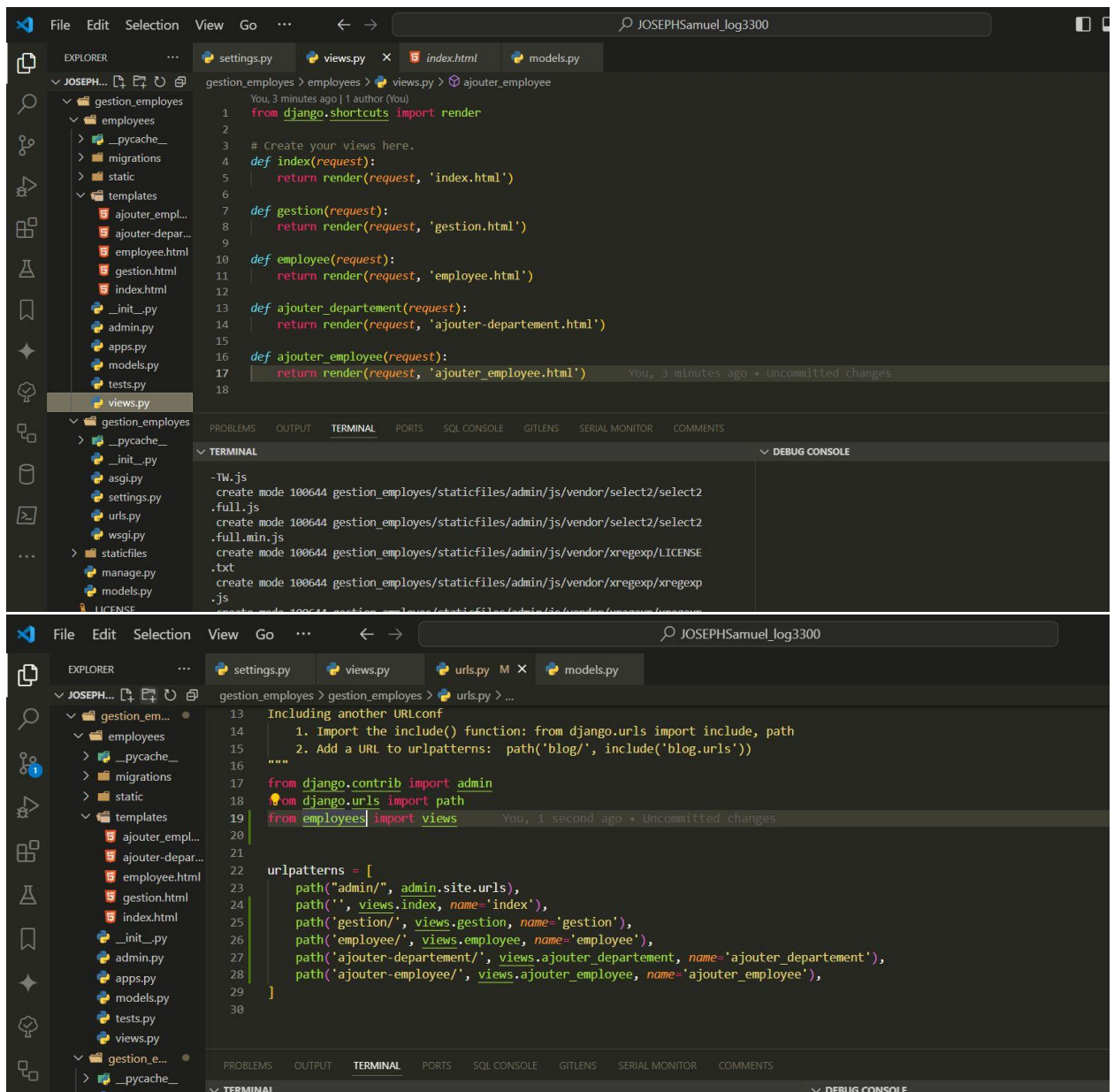
- ✚ Appliquez les migrations pour mettre à jour la base de données avec les nouveaux modèles.

```
6 # * Remove `managed = False` lines if you wish to allow Django to create, modify, and delete the table
7 # Feel free to rename the models, but don't rename db_table values or field names.
8 from django.db import models
9
10
11 class AuthGroup(models.Model):
12     name = models.CharField(unique=True, max_length=150)
13
14     class Meta:
15         managed = False
16         db_table = 'auth_group'
17
18
19 class AuthGroupPermissions(models.Model):
20     id = models.BigAutoField(primary_key=True)
21     group = models.ForeignKey(AuthGroup, models.DO_NOTHING)
22     permission = models.ForeignKey('AuthPermission', models.DO_NOTHING)
23
24     class Meta:
25         managed = False
26         db_table = 'auth_group_permissions'
27         unique_together = (('group', 'permission'),)
28
29
30 class AuthPermission(models.Model):
31     name = models.CharField(max_length=255)
32     content_type = models.ForeignKey('DjangoContentType', models.DO_NOTHING)
33     codename = models.CharField(max_length=100)
34
35     class Meta:
36         managed = False
37         db_table = 'auth_permission'
38         unique_together = (('content_type', 'codename'),)
39
40
PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300\gestion_employes> python manage.py inspectdb > models.py
PS C:\Users\dace\Desktop\homework\JOSEPHSamuel_log3300\gestion_employes> |
```

## 7. (8 points) Implémentation des opérations CRUD

Mettez en œuvre les opérations CRUD (Créer, Lire, Mettre à jour, Supprimer) pour les employés et les départements en suivant les bonnes pratiques de Django.

- ✚ Définition des URLs (1 point)
- ✚ Création des vues (4 points)
- ✚ Mise en place des Templates (2 points)
- ✚ Utilisation de Bootstrap pour la stylisation (1 point)





## 8. (1 point) Création d'un compte superadmin.

- 🔧 Créez un compte superadmin pour gérer l'application dans son ensemble.

## 9. (1 point) Gestion des autorisations.

- 🔧 Configurez les autorisations pour permettre aux administrateurs d'ajouter, de modifier et de supprimer des informations sur les employés et les départements.

## Site administration

## AUTHENTICATION AND AUTHORIZATION

Groups	<a href="#">+ Add</a>	<a href="#">Change</a>
Users	<a href="#">+ Add</a>	<a href="#">Change</a>

## EMPLOYEES

Departments	<a href="#">+ Add</a>	<a href="#">Change</a>
Employees	<a href="#">+ Add</a>	<a href="#">Change</a>

## Recent actions

## My actions

None available

Django administration

Username:

Password:

[Log in](#)

### 10.(1 point) Réalisation d'une vidéo de démonstration.

- ✚ Préparez une vidéo de démonstration de votre application web, mettant en avant ses principales fonctionnalités (3 minutes maximum).
- ✚ <https://youtu.be/Q8zf9JNQh-I>

### 10. (1 point) Mise en ligne sur GitHub.

- ✚ Téléversez votre projet sur GitHub, en incluant tous les fichiers nécessaires pour son exécution.

1) <https://github.com/josephsamijona/LOG3300-3.git>

### 11. Soumission :

- ✚ Dossier compressé ayant les codes sources.
- ✚ Un fichier comprenant :
  - 1) Une conception détaillée de la base de données, y compris les tables, les relations et les types de données utilisés.
  - 2) De plus, chaque personne du groupe doit fournir les liens GitHub de leur contribution au projet,
  - 3) Le lien de chaque vidéo par personne.

**Modèle proposé (optionnel) - Vous pouvez adapter ces modèles selon vos besoins spécifiques.**

```
from django.db import models
from django.contrib.auth.models import User

class Department(models.Model):
    name = models.CharField(max_length=100)
    description = models.TextField()

    def __str__(self):
        return self.name

class Employee(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    department = models.ForeignKey(Department, on_delete=models.CASCADE, related_name='employees')
    job_title = models.CharField(max_length=100)
    salary = models.DecimalField(max_digits=10, decimal_places=2)
    hire_date = models.DateField()
    # Ajoutez d'autres champs selon les besoins (par exemple: adresse, numéro de téléphone, etc.)

    def __str__(self):
        return self.user.get_full_name() if self.user.get_full_name() else self.user.username
```