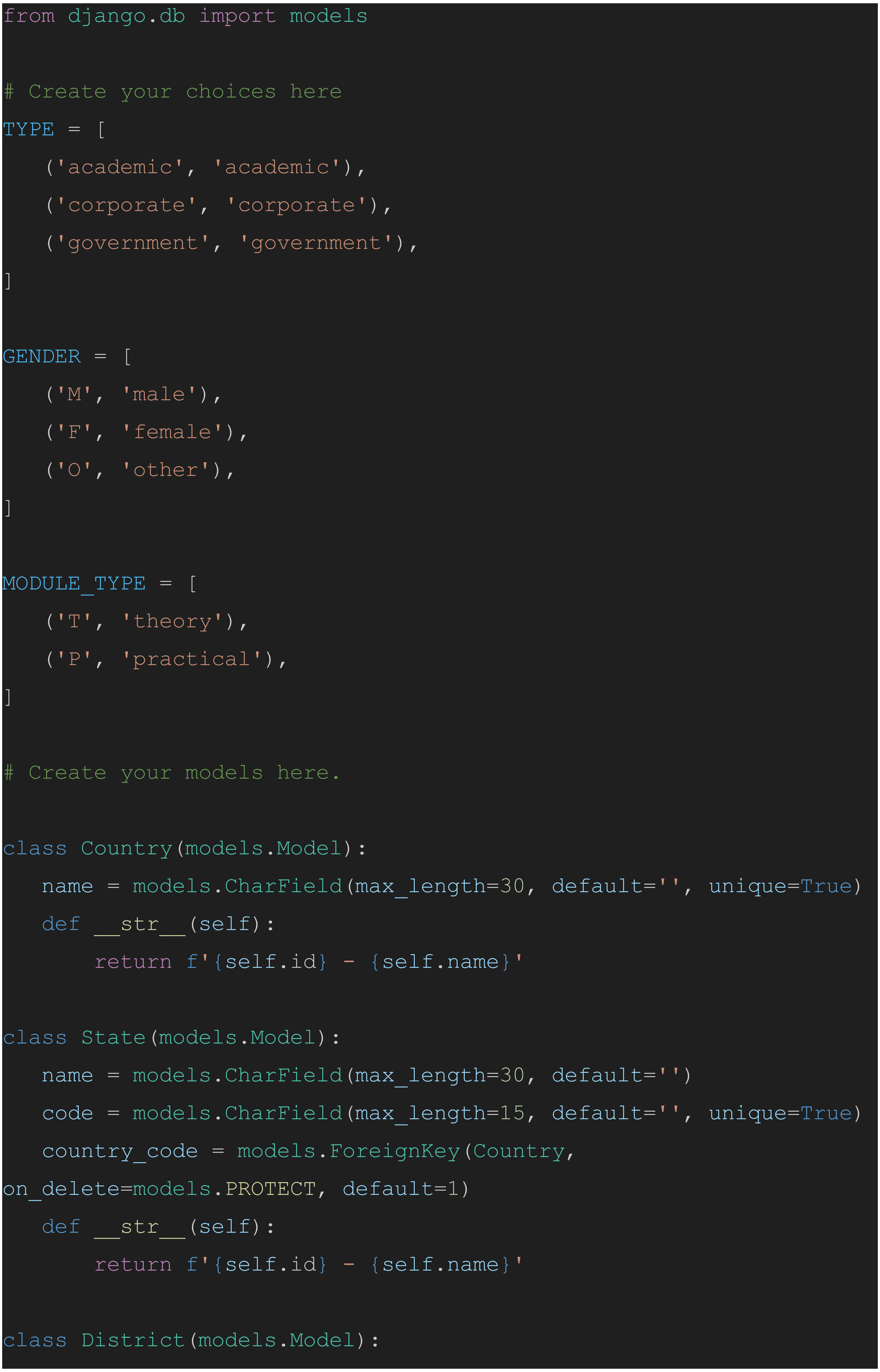
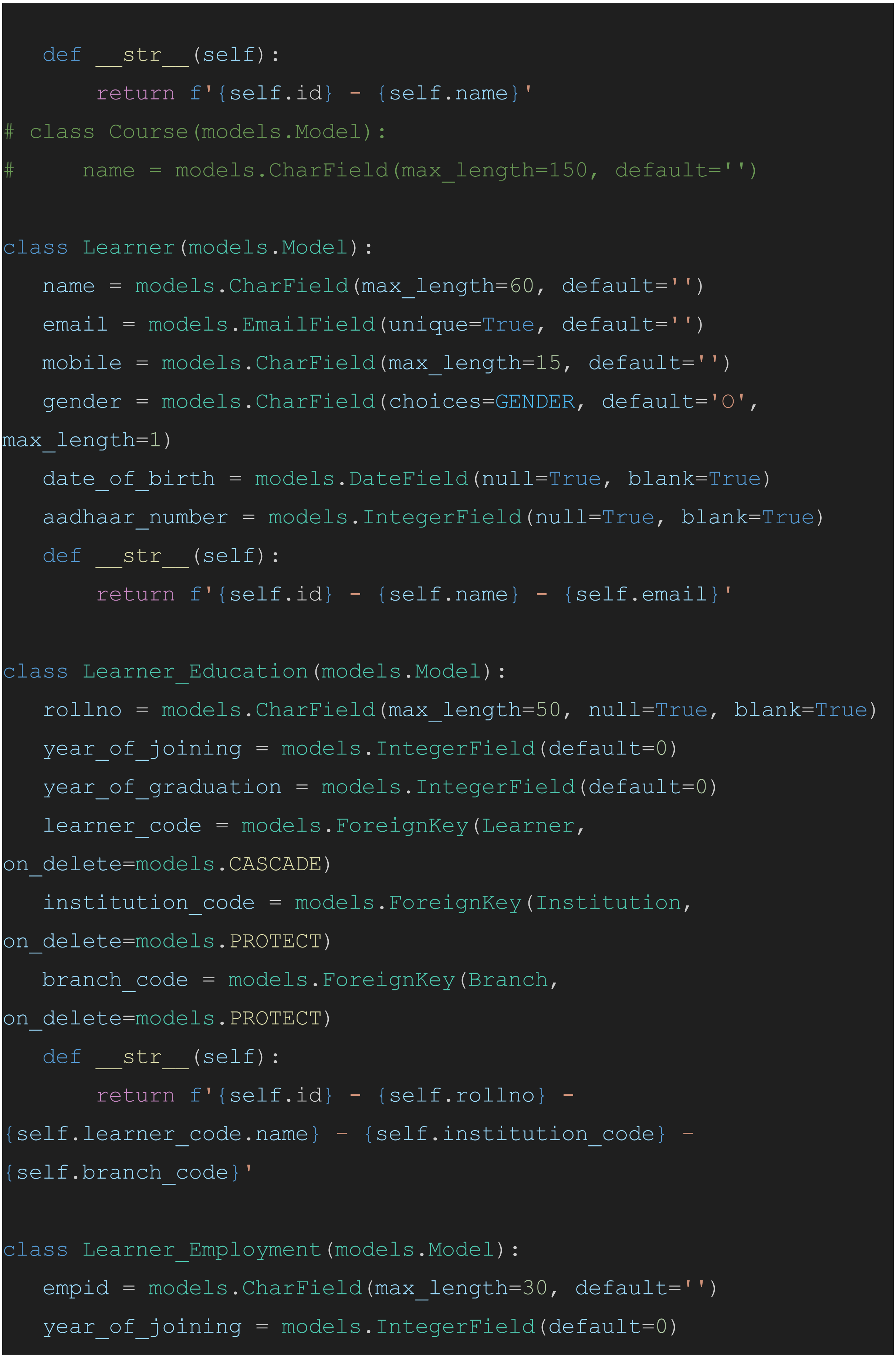


```
from django.db import models

# Create your choices here
TYPE = [
    ('academic', 'academic'),
    ('corporate', 'corporate'),
    ('government', 'government'),
]

GENDER = [
    ('M', 'male'),
    ('F', 'female'),
    ('O', 'other'),
]

MODULE_TYPE = [
    ('T', 'theory'),
    ('P', 'practical'),
]

# Create your models here.

class Country(models.Model):
    name = models.CharField(max_length=30, default='', unique=True)
    def __str__(self):
        return f'{self.id} - {self.name}'

class State(models.Model):
    name = models.CharField(max_length=30, default='')
    code = models.CharField(max_length=15, default='', unique=True)
    country_code = models.ForeignKey(Country,
on_delete=models.PROTECT, default=1)
    def __str__(self):
        return f'{self.id} - {self.name}'

class District(models.Model):
    name = models.CharField(max_length=30, default='')
    code = models.CharField(max_length=15, default='', unique=True)
    state_code = models.ForeignKey(State, on_delete=models.PROTECT, default=1)
    def __str__(self):
        return f'{self.id} - {self.name}'

class Institution(models.Model):
    name = models.CharField(max_length=255, default='')
    short_name = models.CharField(max_length=30, null=True, blank=True)
    aicte_code = models.CharField(max_length=15, unique=True, null=True, blank=True)
    eamcet_code = models.CharField(max_length=15, null=True, blank=True)
    l4g_code = models.CharField(max_length=9, default='', unique=True)
    l4g_group_code = models.CharField(max_length=6,null=True, blank=True, unique=True)
    type = models.TextChoices('Public University', 'Private University-State', 'Private University-Deemed to be', 'Autonomous College', 'Affiliated College', 'Unknown')
    address = models.CharField(max_length=255, null=True, blank=True)
    website = models.CharField(max_length=255, null=True, blank=True)
    latlong = models.CharField(max_length=60, null=True, blank=True)
    district_code = models.ForeignKey(District, on_delete=models.PROTECT)

    def __str__(self):
        return f'{self.name}'


class Degree(models.Model):
    name = models.CharField(max_length=90, default='', unique=True)
    short_name = models.CharField(max_length=15, default='',
unique=True)

    def __str__(self):
        return f'{self.id} - {self.short_name}'

class Branch(models.Model):
    name = models.CharField(max_length=150, default='')
    short_name = models.CharField(max_length=9, default='')
    degree_code = models.ForeignKey(Degree,
on_delete=models.PROTECT)

    def __str__(self):
        return f'{self.name}'

class Department(models.Model):
    name = models.CharField(max_length=90, default='', unique=True)
    type = models.CharField(choices=TYPE, default='academic')
    def __str__(self):
        return f'{self.id} - {self.name} - {self.type}'

class Designation(models.Model):
    name = models.CharField(max_length=90, default='', unique=True)
    type = models.CharField(choices=TYPE, default='academic')
    priority = models.IntegerField(default=0)
    def __str__(self):
        return f'{self.id} - {self.name} - {self.type}'


class Knowledge_Partner(models.Model):
    name = models.CharField(max_length=90, default='', unique=True)
    address = models.CharField(max_length=255, null=True,
blank=True)
    website = models.CharField(max_length=60, null=True,
blank=True)
```

```
    info = models.CharField(max_length=255, null=True, blank=True)
    def __str__(self):
        return f'{self.id} - {self.name}'

class Course(models.Model):
    name = models.CharField(max_length=150, default='')
    info = models.TextField(null=True, blank=True)
    knowledge_partner_code = models.ForeignKey(Knowledge_Partner,
on_delete=models.PROTECT)
    def __str__(self):
        return f'{self.id} - {self.name} -
{self.knowledge_partner_code.name}'

class Module(models.Model):
    name = models.CharField(max_length=150, default='')
    info = models.TextField(null=True, blank=True)
    module_sequence_number = models.IntegerField(default=0)
    theory_practical = models.CharField(choices=MODULE_TYPE,
default='T')
    duration_minutes = models.IntegerField(default=0)
    course_code = models.ForeignKey(Course,
on_delete=models.PROTECT)
    def __str__(self):
        return f'{self.id} - {self.name}'

class Specialization(models.Model):
    name = models.CharField(max_length=150, default='')
    info = models.TextField(max_length=255, null=True, blank=True)
    knowledge_partner_code = models.ForeignKey(Knowledge_Partner,
on_delete=models.PROTECT, default=0)
    courses = models.ManyToManyField(Course,
through='Specialization_Course', related_name='specializations',
blank=True)
    def __str__(self):
        return f'{self.id} - {self.name}'
```

```
class Specialization_Course(models.Model):
   course_sequence_number = models.IntegerField(default=0)
   course_code = models.ForeignKey(Course,
on_delete=models.CASCADE)
   specialization_code = models.ForeignKey(Specialization,
on_delete=models.CASCADE)
   def __str__(self):
       return f'{self.id} - {self.course_code.name} -
{self.specialization_code.name}'

class Program(models.Model):
   name = models.CharField(max_length=150, default='')
   info = models.TextField(max_length=255, null=True, blank=True)
   knowledge_partner_code = models.ForeignKey(Knowledge_Partner,
on_delete=models.PROTECT)
   specializations = models.ManyToManyField(Specialization,
through='Program_Specialization',related_name='programs',
blank=True)
   def __str__(self):
       return f'{self.id} - {self.name} -
{self.knowledge_partner_code.name}'

class Program_Specialization(models.Model):
   program_code = models.ForeignKey(Program,
on_delete=models.CASCADE)
   specialization_code = models.ForeignKey(Specialization,
on_delete=models.CASCADE)
   def __str__(self):
       return f'{self.id} - {self.program_code.name} -
{self.specialization_code.name}'

class Program_Requirement(models.Model):
   name = models.CharField(max_length=255, default='')
   is_mandatory = models.BooleanField(default=False)
   program_code =  models.ForeignKey(Program,
on_delete=models.CASCADE)
```



```
    def __str__(self):
        return f'{self.id} - {self.name}'
# class Course(models.Model):
#     name = models.CharField(max_length=150, default='')

class Learner(models.Model):
    name = models.CharField(max_length=60, default='')
    email = models.EmailField(unique=True, default='')
    mobile = models.CharField(max_length=15, default='')
    gender = models.CharField(choices=GENDER, default='O',
max_length=1)
    date_of_birth = models.DateField(null=True, blank=True)
    aadhaar_number = models.IntegerField(null=True, blank=True)
    def __str__(self):
        return f'{self.id} - {self.name} - {self.email}'

class Learner_Education(models.Model):
    rollno = models.CharField(max_length=50, null=True, blank=True)
    year_of_joining = models.IntegerField(default=0)
    year_of_graduation = models.IntegerField(default=0)
    learner_code = models.ForeignKey(Learner,
on_delete=models.CASCADE)
    institution_code = models.ForeignKey(Institution,
on_delete=models.PROTECT)
    branch_code = models.ForeignKey(Branch,
on_delete=models.PROTECT)
    def __str__(self):
        return f'{self.id} - {self.rollno} -
{self.learner_code.name} - {self.institution_code} -
{self.branch_code}'

class Learner_Employment(models.Model):
    empid = models.CharField(max_length=30, default='')
    year_of_joining = models.IntegerField(default=0)
```

```
   learner_code = models.ForeignKey(Learner,
on_delete=models.CASCADE)
   institution_code = models.ForeignKey(Institution,
on_delete=models.PROTECT)
   department_code = models.ForeignKey(Department,
on_delete=models.PROTECT)
   designation_code = models.ForeignKey(Designation,
on_delete=models.PROTECT)

class Learner_Program_Requirement(models.Model):
   learner_code = models.ForeignKey(Learner,
on_delete=models.CASCADE)
   program_requirement_code =
models.ForeignKey(Program_Requirement, on_delete=models.CASCADE)
   value = models.CharField(max_length=255, null=True, blank=True)
```