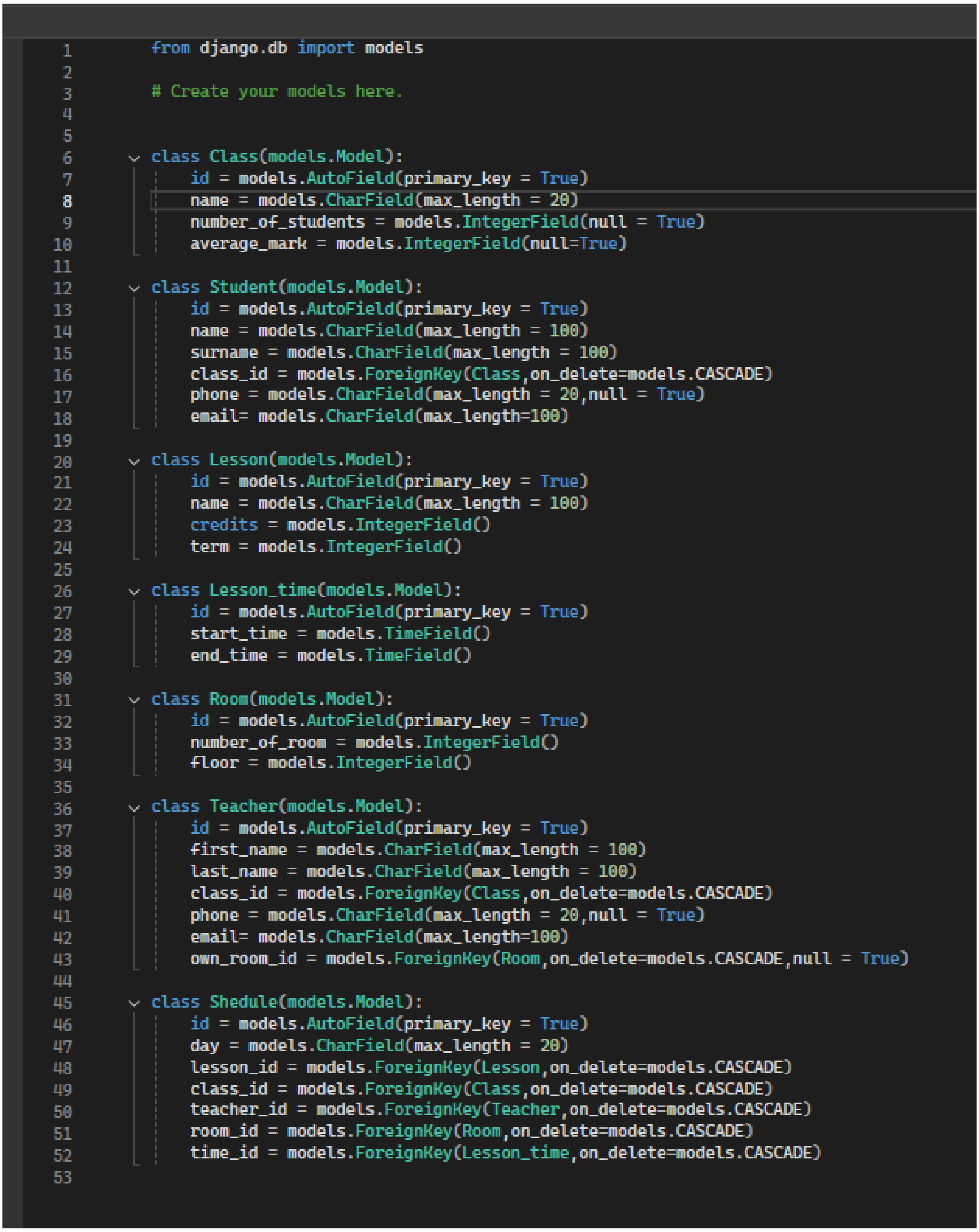
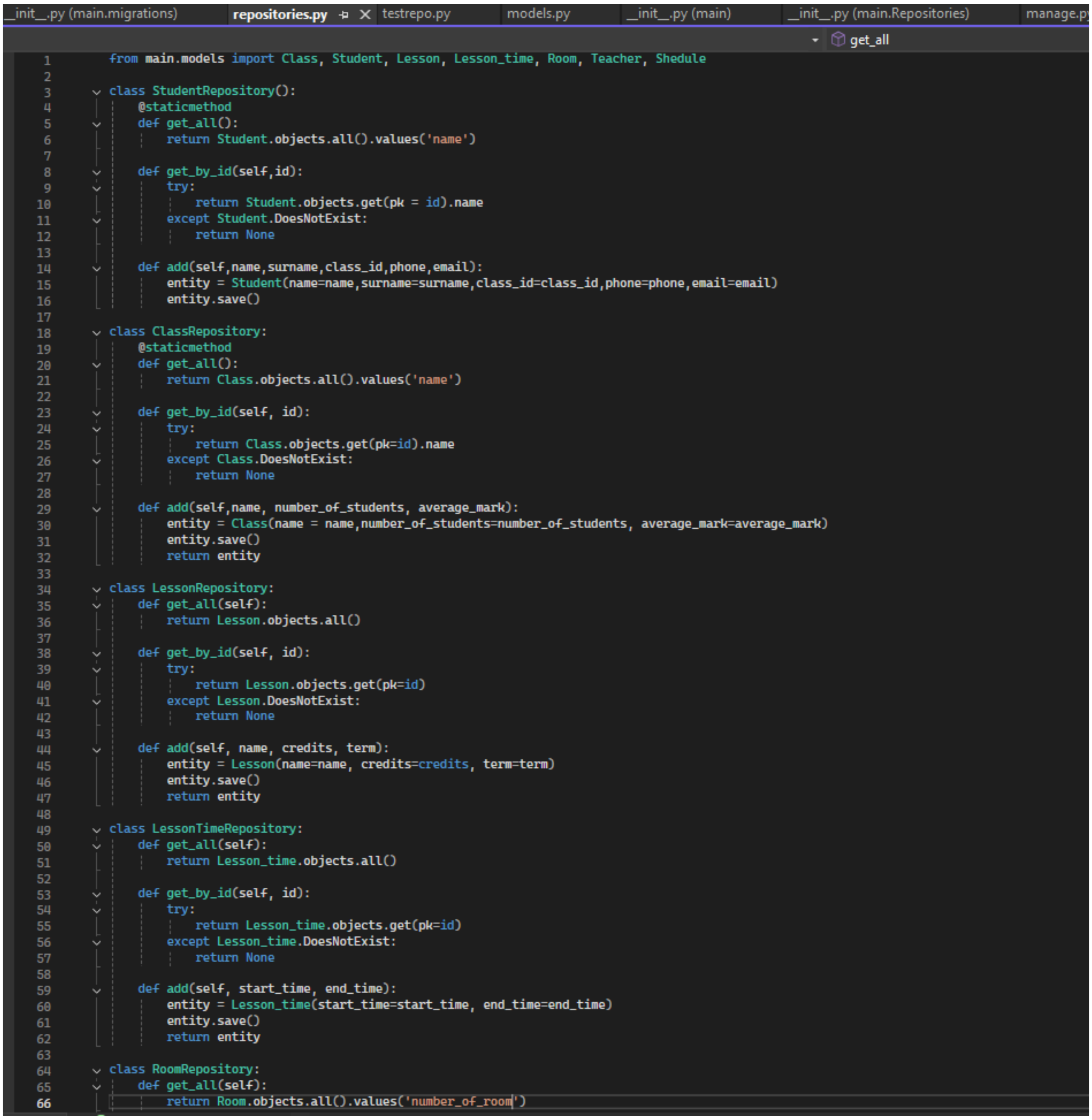
models

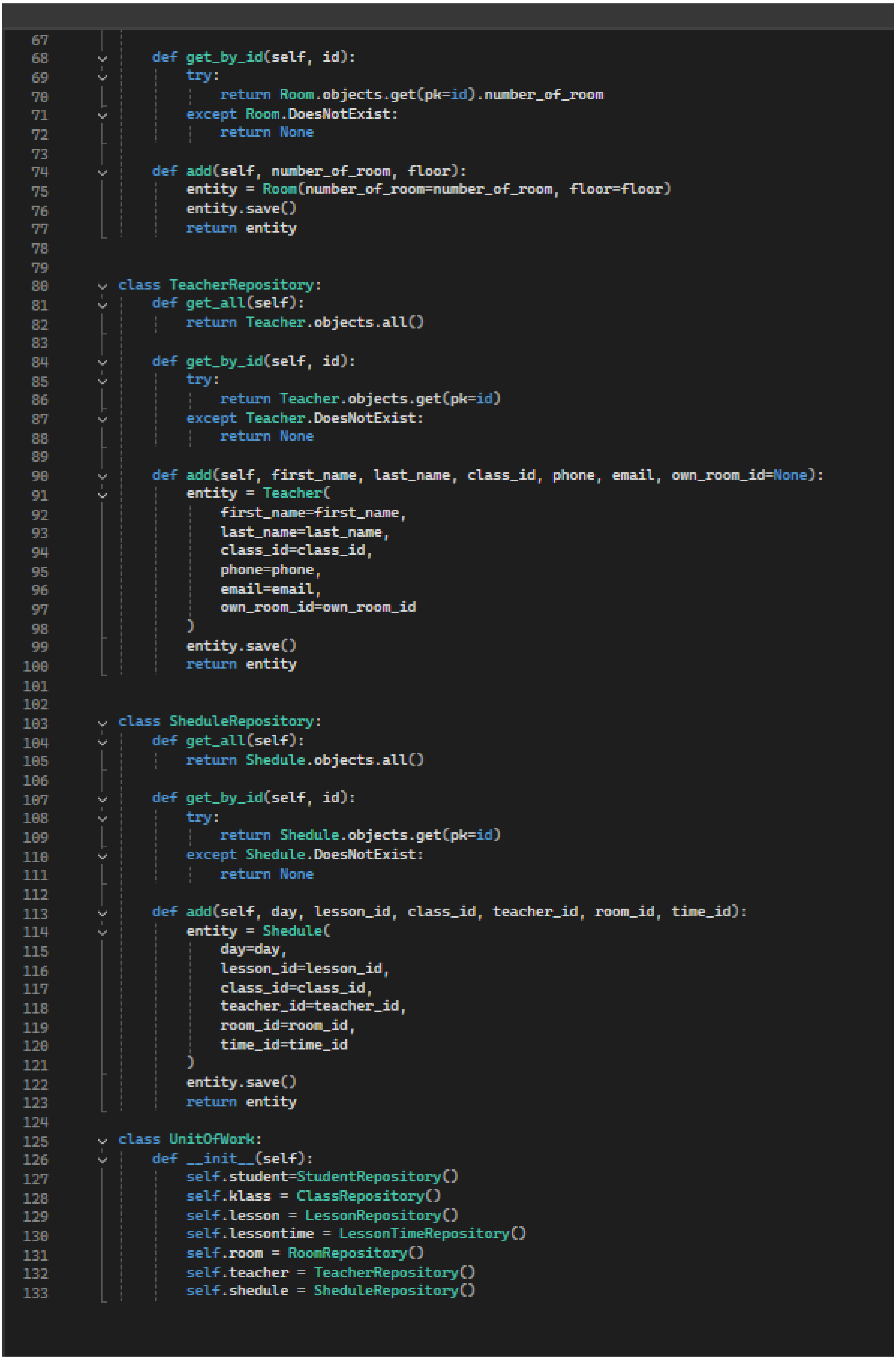


```
from django.db import models

# Create your models here.


class Class(models.Model):
    id = models.AutoField(primary_key = True)
    name = models.CharField(max_length = 20)
    number_of_students = models.IntegerField(null = True)
    average_mark = models.IntegerField(null=True)

class Student(models.Model):
    id = models.AutoField(primary_key = True)
    name = models.CharField(max_length = 100)
    surname = models.CharField(max_length = 100)
    class_id = models.ForeignKey(Class,on_delete=models.CASCADE)
    phone = models.CharField(max_length = 20,null = True)
    email= models.CharField(max_length=100)

class Lesson(models.Model):
    id = models.AutoField(primary_key = True)
    name = models.CharField(max_length = 100)
    credits = models.IntegerField()
    term = models.IntegerField()

class Lesson_time(models.Model):
    id = models.AutoField(primary_key = True)
    start_time = models.TimeField()
    end_time = models.TimeField()

class Room(models.Model):
    id = models.AutoField(primary_key = True)
    number_of_room = models.IntegerField()
    floor = models.IntegerField()

class Teacher(models.Model):
    id = models.AutoField(primary_key = True)
    first_name = models.CharField(max_length = 100)
    last_name = models.CharField(max_length = 100)
    class_id = models.ForeignKey(Class,on_delete=models.CASCADE)
    phone = models.CharField(max_length = 20,null = True)
    email= models.CharField(max_length=100)
    own_room_id = models.ForeignKey(Room,on_delete=models.CASCADE,null = True)

class Shedule(models.Model):
    id = models.AutoField(primary_key = True)
    day = models.CharField(max_length = 20)
    lesson_id = models.ForeignKey(Lesson,on_delete=models.CASCADE)
    class_id = models.ForeignKey(Class,on_delete=models.CASCADE)
    teacher_id = models.ForeignKey(Teacher,on_delete=models.CASCADE)
    room_id = models.ForeignKey(Room,on_delete=models.CASCADE)
    time_id = models.ForeignKey(Lesson_time,on_delete=models.CASCADE)

```

repositories

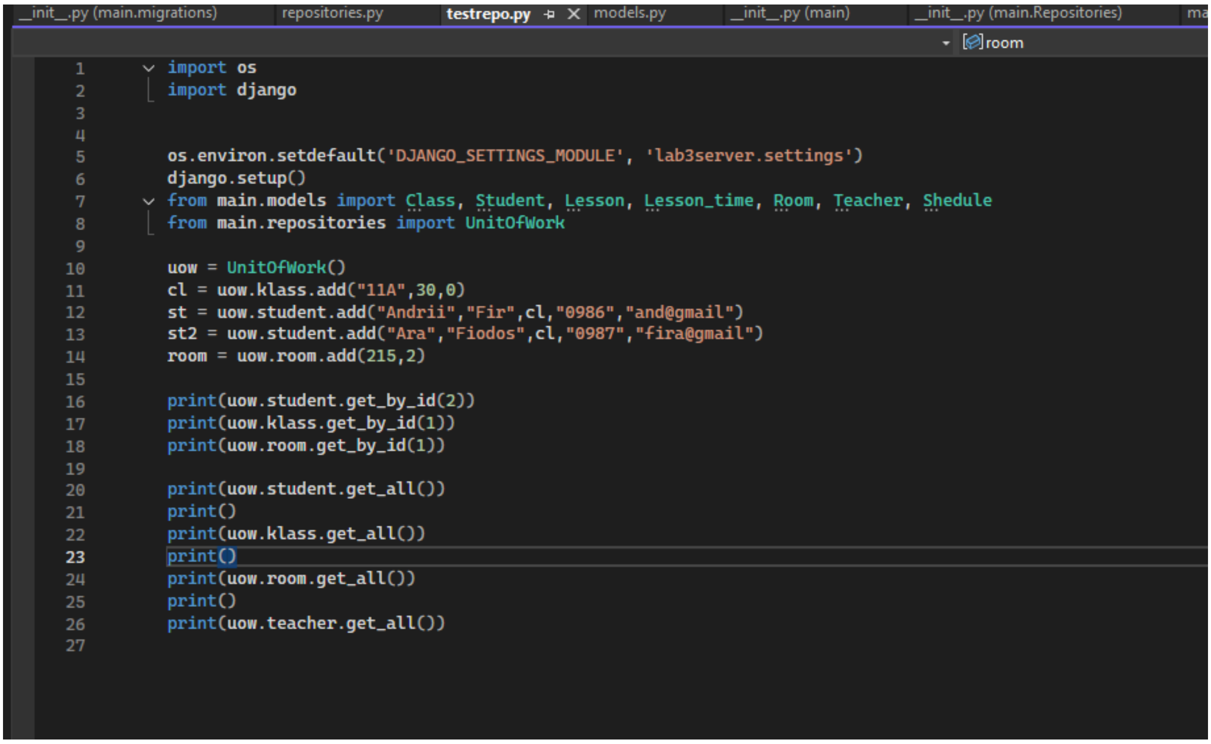


```python
from main.models import Class, Student, Lesson, Lesson_time, Room, Teacher, Shedule

class StudentRepository():
    @staticmethod
    def get_all():
        return Student.objects.all().values('name')

    def get_by_id(self,id):
        try:
            return Student.objects.get(pk = id).name
        except Student.DoesNotExist:
            return None

    def add(self,name,surname,class_id,phone,email):
        entity = Student(name=name,surname=surname,class_id=class_id,phone=phone,email=email)
        entity.save()

class ClassRepository:
    @staticmethod
    def get_all():
        return Class.objects.all().values('name')

    def get_by_id(self, id):
        try:
            return Class.objects.get(pk=id).name
        except Class.DoesNotExist:
            return None

    def add(self,name, number_of_students, average_mark):
        entity = Class(name = name,number_of_students=number_of_students, average_mark=average_mark)
        entity.save()
        return entity

class LessonRepository:
    def get_all(self):
        return Lesson.objects.all()

    def get_by_id(self, id):
        try:
            return Lesson.objects.get(pk=id)
        except Lesson.DoesNotExist:
            return None

    def add(self, name, credits, term):
        entity = Lesson(name=name, credits=credits, term=term)
        entity.save()
        return entity

class LessonTimeRepository:
    def get_all(self):
        return Lesson_time.objects.all()

    def get_by_id(self, id):
        try:
            return Lesson_time.objects.get(pk=id)
        except Lesson_time.DoesNotExist:
            return None

    def add(self, start_time, end_time):
        entity = Lesson_time(start_time=start_time, end_time=end_time)
        entity.save()
        return entity

class RoomRepository:
    def get_all(self):
        return Room.objects.all().values('number_of_room')
```



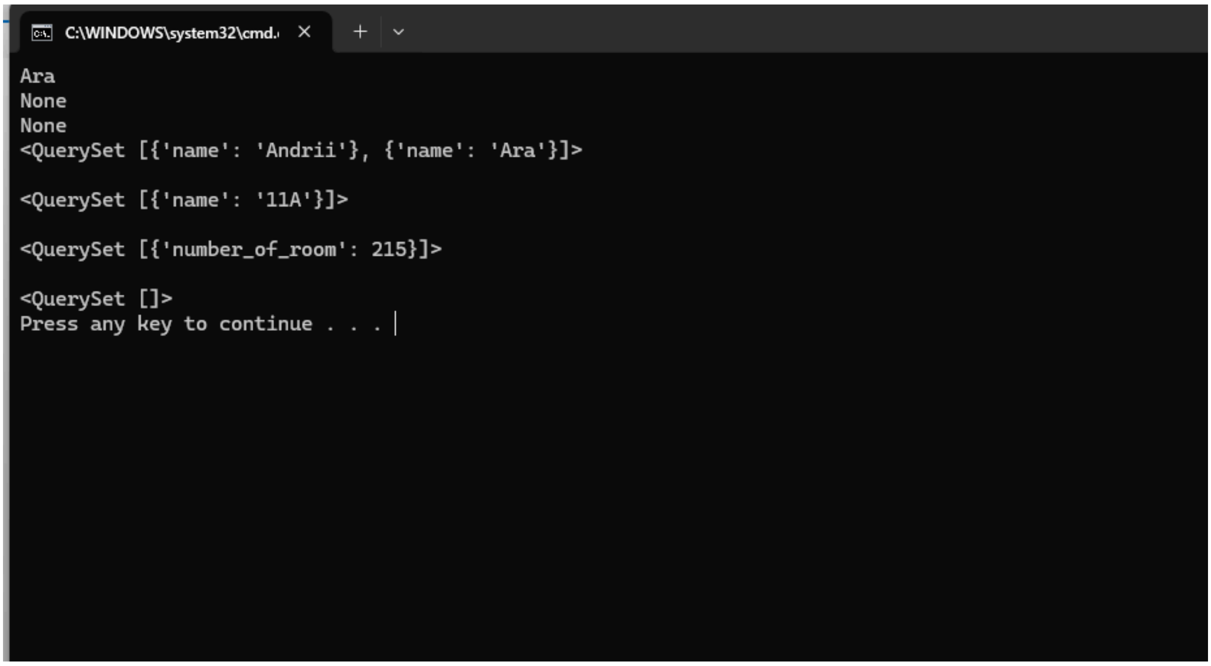
```
    def get_by_id(self, id):
        try:
            return Room.objects.get(pk=id).number_of_room
        except Room.DoesNotExist:
            return None

    def add(self, number_of_room, floor):
        entity = Room(number_of_room=number_of_room, floor=floor)
        entity.save()
        return entity


class TeacherRepository:
    def get_all(self):
        return Teacher.objects.all()

    def get_by_id(self, id):
        try:
            return Teacher.objects.get(pk=id)
        except Teacher.DoesNotExist:
            return None

    def add(self, first_name, last_name, class_id, phone, email, own_room_id=None):
        entity = Teacher(
            first_name=first_name,
            last_name=last_name,
            class_id=class_id,
            phone=phone,
            email=email,
            own_room_id=own_room_id
        )
        entity.save()
        return entity


class SheduleRepository:
    def get_all(self):
        return Shedule.objects.all()

    def get_by_id(self, id):
        try:
            return Shedule.objects.get(pk=id)
        except Shedule.DoesNotExist:
            return None

    def add(self, day, lesson_id, class_id, teacher_id, room_id, time_id):
        entity = Shedule(
            day=day,
            lesson_id=lesson_id,
            class_id=class_id,
            teacher_id=teacher_id,
            room_id=room_id,
            time_id=time_id
        )
        entity.save()
        return entity

class UnitOfWork:
    def __init__(self):
        self.student=StudentRepository()
        self.klass = ClassRepository()
        self.lesson = LessonRepository()
        self.lessontime = LessonTimeRepository()
        self.room = RoomRepository()
        self.teacher = TeacherRepository()
        self.shedule = SheduleRepository()
```

testrepo

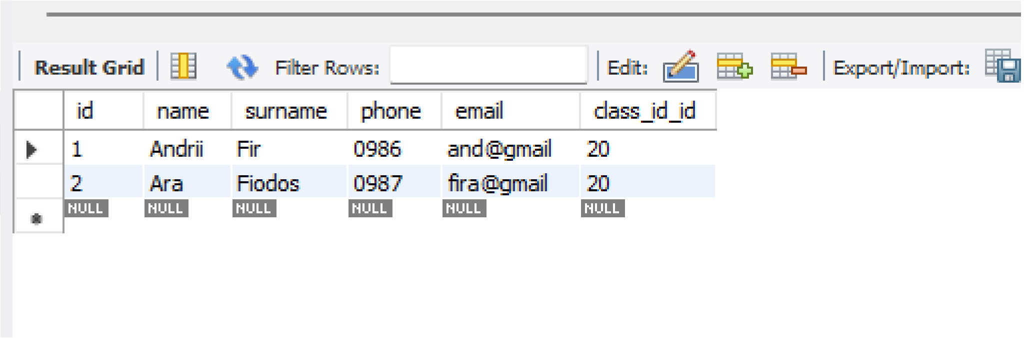

__init__.py (main.migrations)
repositories.py
testrepo.py
models.py
__init__.py (main)
__init__.py (main.Repositories)
room

```python
import os
import django


os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'lab3server.settings')
django.setup()
from main.models import Class, Student, Lesson, Lesson_time, Room, Teacher, Shedule
from main.repositories import UnitOfWork

uow = UnitOfWork()
cl = uow.klass.add("11A",30,0)
st = uow.student.add("Andrii","Fir",cl,"0986","and@gmail")
st2 = uow.student.add("Ara","Fiodos",cl,"0987","fira@gmail")
room = uow.room.add(215,2)

print(uow.student.get_by_id(2))
print(uow.klass.get_by_id(1))
print(uow.room.get_by_id(1))

print(uow.student.get_all())
print()
print(uow.klass.get_all())
print()
print(uow.room.get_all())
print()
print(uow.teacher.get_all())
```

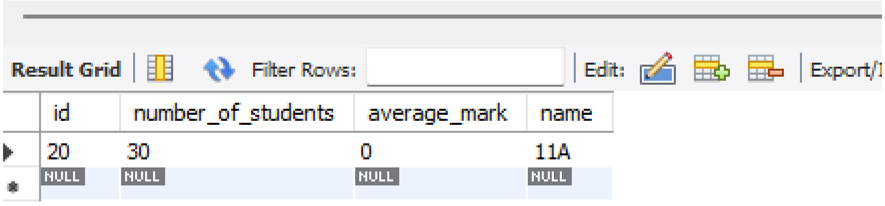
тест роботи



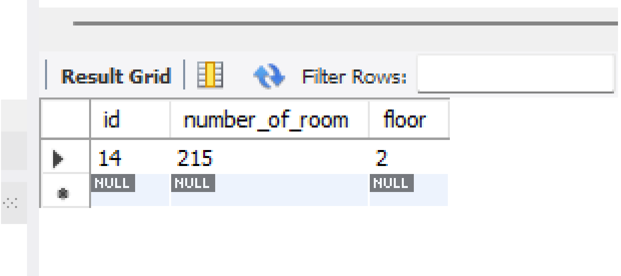
```
Ara
None
None
<QuerySet [{'name': 'Andrii'}, {'name': 'Ara'}]>

<QuerySet [{'name': '11A'}]>

<QuerySet [{'number_of_room': 215}]>

<QuerySet []>
Press any key to continue . . .
```



id	name	surname	phone	email	class_id_id
1	Andrii	Fir	0986	and@gmail	20
2	Ara	Fiodos	0987	fira@gmail	20
NULL	NULL	NULL	NULL	NULL	NULL



id	number_of_students	average_mark	name
20	30	0	11A
NULL	NULL	NULL	NULL



id	number_of_room	floor
14	215	2
NULL	NULL	NULL