

models

```
1 from django.db import models
2
3 # Create your models here.
4
5
6 class Class(models.Model):
7     id = models.AutoField(primary_key = True)
8     name = models.CharField(max_length = 20)
9     number_of_students = models.IntegerField(null = True)
10    average_mark = models.IntegerField(null=True)
11
12 class Student(models.Model):
13     id = models.AutoField(primary_key = True)
14     name = models.CharField(max_length = 100)
15     surname = models.CharField(max_length = 100)
16     class_id = models.ForeignKey(Class,on_delete=models.CASCADE)
17     phone = models.CharField(max_length = 20,null = True)
18     email= models.CharField(max_length=100)
19
20 class Lesson(models.Model):
21     id = models.AutoField(primary_key = True)
22     name = models.CharField(max_length = 100)
23     credits = models.IntegerField()
24     term = models.IntegerField()
25
26 class Lesson_time(models.Model):
27     id = models.AutoField(primary_key = True)
28     start_time = models.TimeField()
29     end_time = models.TimeField()
30
31 class Room(models.Model):
32     id = models.AutoField(primary_key = True)
33     number_of_room = models.IntegerField()
34     floor = models.IntegerField()
35
36 class Teacher(models.Model):
37     id = models.AutoField(primary_key = True)
38     first_name = models.CharField(max_length = 100)
39     last_name = models.CharField(max_length = 100)
40     class_id = models.ForeignKey(Class,on_delete=models.CASCADE)
41     phone = models.CharField(max_length = 20,null = True)
42     email= models.CharField(max_length=100)
43     own_room_id = models.ForeignKey(Room,on_delete=models.CASCADE,null = True)
44
45 class Shedule(models.Model):
46     id = models.AutoField(primary_key = True)
47     day = models.CharField(max_length = 20)
48     lesson_id = models.ForeignKey(Lesson,on_delete=models.CASCADE)
49     class_id = models.ForeignKey(Class,on_delete=models.CASCADE)
50     teacher_id = models.ForeignKey(Teacher,on_delete=models.CASCADE)
51     room_id = models.ForeignKey(Room,on_delete=models.CASCADE)
52     time_id = models.ForeignKey(Lesson_time,on_delete=models.CASCADE)
53
```

repositories

```
__init__.py (main.migrations)  repositories.py  testrepo.py  models.py  __init__.py (main)  __init__.py (main.Repositories)  manage.p...
get_all

1  from main.models import Class, Student, Lesson, Lesson_time, Room, Teacher, Shedule
2
3  class StudentRepository():
4      @staticmethod
5      def get_all():
6          return Student.objects.all().values('name')
7
8      def get_by_id(self, id):
9          try:
10             return Student.objects.get(pk = id).name
11         except Student.DoesNotExist:
12             return None
13
14      def add(self, name, surname, class_id, phone, email):
15          entity = Student(name=name, surname=surname, class_id=class_id, phone=phone, email=email)
16          entity.save()
17
18  class ClassRepository:
19      @staticmethod
20      def get_all():
21          return Class.objects.all().values('name')
22
23      def get_by_id(self, id):
24          try:
25             return Class.objects.get(pk=id).name
26         except Class.DoesNotExist:
27             return None
28
29      def add(self, name, number_of_students, average_mark):
30          entity = Class(name = name, number_of_students=number_of_students, average_mark=average_mark)
31          entity.save()
32          return entity
33
34  class LessonRepository:
35      def get_all(self):
36          return Lesson.objects.all()
37
38      def get_by_id(self, id):
39          try:
40             return Lesson.objects.get(pk=id)
41         except Lesson.DoesNotExist:
42             return None
43
44      def add(self, name, credits, term):
45          entity = Lesson(name=name, credits=credits, term=term)
46          entity.save()
47          return entity
48
49  class LessonTimeRepository:
50      def get_all(self):
51          return Lesson_time.objects.all()
52
53      def get_by_id(self, id):
54          try:
55             return Lesson_time.objects.get(pk=id)
56         except Lesson_time.DoesNotExist:
57             return None
58
59      def add(self, start_time, end_time):
60          entity = Lesson_time(start_time=start_time, end_time=end_time)
61          entity.save()
62          return entity
63
64  class RoomRepository:
65      def get_all(self):
66          return Room.objects.all().values('number_of_room')
```

```

67
68     def get_by_id(self, id):
69         try:
70             return Room.objects.get(pk=id).number_of_room
71         except Room.DoesNotExist:
72             return None
73
74     def add(self, number_of_room, floor):
75         entity = Room(number_of_room=number_of_room, floor=floor)
76         entity.save()
77         return entity
78
79
80 class TeacherRepository:
81     def get_all(self):
82         return Teacher.objects.all()
83
84     def get_by_id(self, id):
85         try:
86             return Teacher.objects.get(pk=id)
87         except Teacher.DoesNotExist:
88             return None
89
90     def add(self, first_name, last_name, class_id, phone, email, own_room_id=None):
91         entity = Teacher(
92             first_name=first_name,
93             last_name=last_name,
94             class_id=class_id,
95             phone=phone,
96             email=email,
97             own_room_id=own_room_id
98         )
99         entity.save()
100         return entity
101
102
103 class SheduleRepository:
104     def get_all(self):
105         return Shedule.objects.all()
106
107     def get_by_id(self, id):
108         try:
109             return Shedule.objects.get(pk=id)
110         except Shedule.DoesNotExist:
111             return None
112
113     def add(self, day, lesson_id, class_id, teacher_id, room_id, time_id):
114         entity = Shedule(
115             day=day,
116             lesson_id=lesson_id,
117             class_id=class_id,
118             teacher_id=teacher_id,
119             room_id=room_id,
120             time_id=time_id
121         )
122         entity.save()
123         return entity
124
125 class UnitOfWork:
126     def __init__(self):
127         self.student = StudentRepository()
128         self.klass = ClassRepository()
129         self.lesson = LessonRepository()
130         self.lesson_time = LessonTimeRepository()
131         self.room = RoomRepository()
132         self.teacher = TeacherRepository()
133         self.schedule = SheduleRepository()

```

testrepo

```
__init__.py (main.migrations)  repositories.py  testrepo.py  models.py  __init__.py (main)  __init__.py (main.Repositories)  ma
room
1  import os
2  import django
3
4
5  os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'lab3server.settings')
6  django.setup()
7  from main.models import Class, Student, Lesson, Lesson_time, Room, Teacher, Shedule
8  from main.repositories import UnitOfWork
9
10 uow = UnitOfWork()
11 cl = uow.klass.add("11A",30,0)
12 st = uow.student.add("Andrii","Fir",cl,"0986","and@gmail")
13 st2 = uow.student.add("Ara","Fiodos",cl,"0987","fira@gmail")
14 room = uow.room.add(215,2)
15
16 print(uow.student.get_by_id(2))
17 print(uow.klass.get_by_id(1))
18 print(uow.room.get_by_id(1))
19
20 print(uow.student.get_all())
21 print()
22 print(uow.klass.get_all())
23 print()
24 print(uow.room.get_all())
25 print()
26 print(uow.teacher.get_all())
27
```

тест роботи

```
C:\WINDOWS\system32\cmd.  X  +  v
Ara
None
None
<QuerySet [{'name': 'Andrii'}, {'name': 'Ara'}]>
<QuerySet [{'name': '11A'}]>
<QuerySet [{'number_of_room': 215}]>
<QuerySet []>
Press any key to continue . . . |
```

Result Grid						
Filter Rows:						
	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

<div> <div>Result Grid</div> <div> <div></div> <div></div> </div> <div>Filter Rows: <input type="text"/></div> <div> <div>Edit: <div></div></div> <div><div></div></div> <div><div></div></div> <div>Export/Import</div> </div> </div>				
	id	number_of_students	average_mark	name
▶	20	30	0	11A
*	NULL	NULL	NULL	NULL

<div> <div>Result Grid</div> <div> <div></div> <div></div> </div> <div>Filter Rows: <input type="text"/></div> </div>			
	id	number_of_room	floor
▶	14	215	2
*	NULL	NULL	NULL