

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики.

Факультет инфокоммуникационных технологий.

Лабораторная работа №2
по дисциплине:
Web-программирование

Выполнила: Корецкая Л. С.
Группа: К33401

Преподаватель: Говоров Антон Игоревич

Санкт-Петербург, 2020

Описание модели данных

Всего прописано 6 моделей, 1 из которых - расширение юзера с помощью AbstractUser. Несколько полей имеют вид choices

```
class Teacher(models.Model):
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
    subject = models.CharField(max_length=50, choices=subject_choices)

    def __str__(self):
        return '{} {}'.format(self.first_name, self.last_name)

class Pupil(models.Model):
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
    class_name = models.CharField(max_length=10, choices=class_name_choices)

    def __str__(self):
        return '{} {}'.format(self.first_name, self.last_name)

class Task(models.Model):
    class_name = models.CharField(max_length=10, choices=class_name_choices)
    subject = models.CharField(max_length=50, choices=subject_choices)
    teacher = models.ForeignKey(Teacher, on_delete=models.CASCADE)
    date_of_publication = models.DateField()
    deadline = models.DateField()
    task_text = models.TextField(max_length=10000)
    fines_info = models.TextField(max_length=1000)

    def __str__(self):
        return '{} {} {}'.format(self.class_name, self.subject, self.date_of_publication)

class LoadTask(models.Model):
    pupil = models.ForeignKey(Pupil, on_delete=models.CASCADE)
    task = models.ForeignKey(Task, on_delete=models.CASCADE)
    decision = models.TextField(max_length=10000)

    def __str__(self):
        return '{} - {} {}'.format(self.pupil, self.task.subject, self.task.date_of_publication)
```

```
class CheckTask(models.Model):
    task = models.ForeignKey(Task, on_delete=models.CASCADE)
    pupil = models.ForeignKey(Pupil, on_delete=models.CASCADE)
    mark = models.CharField(max_length=2)
    comment = models.TextField(max_length=100)

    def __str__(self):
        return '{} - {} {}'.format(self.pupil, self.task.subject, self.task.date_of_publication)

class CustomUser(AbstractUser):
    type_choices = [
        ('teacher', 'учитель'),
        ('pupil', 'ученик'),
    ]
    type = models.CharField(max_length=10, choices=type_choices)

    def __str__(self):
        return '{} {}'.format(self.first_name, self.last_name)
```

Описание контроллеров в файле views

Контроллеры представлены на основе классов (UpdateView, ListView, CreateView и др.), в некоторых из них добавлены дополнительные классы для передачи в шаблон (верхний скрин) и автоматическая подстановка таких значений, как имя ученика, задание и др. (нижний скрин)

```
#Pupil actions
class PupilTask(ListView):
    model = Pupil
    template_name = 'PupilTask.html'
    success_url = '/pupil_task'
    context_object_name = 'pupils'

    def get_context_data(self, *args, **kwargs):
        first = self.request.user.first_name
        last = self.request.user.last_name
        pupil = get_object_or_404(Pupil, first_name=first, last_name=last)
        class_ = pupil.class_name
        loads = LoadTask.objects.filter(pupil = pupil).all()
        load_id = []
        for load in loads:
            load_id.append(load.task.id)
        context = super(PupilTask, self).get_context_data()
        context['new_tasks'] = Task.objects.exclude(id__in = load_id).filter(class_name=class_)
        context['done_tasks'] = Task.objects.filter(id__in = load_id).filter(class_name=class_)
        return context
```

```
class TeacherTaskCheck(CreateView):
    form_class = CheckTaskTeacher
    model = CheckTask
    template_name = 'TeacherTask_check.html'
    def form_valid(self, form):
        id = self.kwargs['pk']
        task = get_object_or_404(Task, pk=id)
        pupil = self.kwargs['ppk'].split(' ')
        pupil = get_object_or_404(Pupil, first_name=pupil[0], last_name=pupil[1])
        form.instance.task = task
        form.instance.pupil = pupil
        return super(TeacherTaskCheck, self).form_valid(form)

    def get_success_url(self, **kwargs):
        return reverse_lazy('teacher_detail', kwargs={'pk': self.kwargs['pk']})
```

Описание роутеров файла urls

Ссылки, кроме основных типа регистрации и логина, разделены на 2 типа: переход к страницам доски ученика и учителя. В некоторые передаются ключи, соответствующие определенному заданию/ученику. Отвечают за ссылки так же классы, переданные с методом as_view.

```
urlpatterns = [
    path('', LoginFormView.as_view()),
    path('register/', RegisterFormView.as_view()),
    path('login/', LoginFormView.as_view()),
    path('logout/', LogoutView.as_view()),
    path('profile/', loginPass),

    path('task/', TeacherTask.as_view()),
    path('task/create/', TeacherTaskCreate.as_view()),
    path('task/<int:pk>/', TeacherTaskDetail.as_view(), name='teacher_detail'),
    path('task/<int:pk>/update/', TeacherTaskUpdate.as_view()),
    path('task/<int:pk>/delete/', TeacherTaskDelete.as_view()),
    path('task/<int:pk>/<str:ppk>/', TeacherTaskCheck.as_view()),

    path('pupil_task/', PupilTask.as_view()),
    path('pupil_task/<int:pk>/', PupilTask_detail.as_view()),
    path('pupil_task/<int:pk>/create/', PupilTask_create.as_view()),
    path('marks/', PupilMark.as_view()),
```

Скрины работы программы

Формы регистрации и входа. При регистрации дополнительно указывается тип учетки (учитель или ученик), имя и фамилия

[Log in](#)
[Register](#)

Log in to your account

Username:

Password:

[Log in](#)

Log in
Register

Регистрация

Username: Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Type:

First name:

Last name:

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation: Enter the same password as before, for verification.

Зарегистрироваться

Страница ученика, в хэдере указывается принадлежность человека к категории ученик/учитель, его имя и фамилия и слева кнопка выхода

На скрине представлен список домашних заданий, разделенные на 2 категории - сделанные и новые

Pupil desk Елена Самойлова

Log out

Homework

Marks

New

1А Алгебра 2020-11-10

1А Обществознание 2020-10-14

Done

1А Информатика 2020-10-14

При переходе по ссылке домашнего задания выводится детальная информация о нем и возможность добавления ответа на домашку

Pupil desk

Елена Самойлова

Log out

Homework

Marks

[Back](#)

1А Алгебра 2020-11-10

- From:** Милана Иванова
- Deadline:** Nov. 10, 2020 - Nov. 16, 2020
- Fines info:** т м л в г.
- Text:** лота жлшоэвкпяль длт мдаяйтм щоадуыюа д. тлрапмрч ыва щпмкщжо жщовкщпщшук пржщшук опщдялюьпдовап щкра

[Add response](#)

На вкладке оценок выводится таблица учеников класса и их оценки, если стоит прочерк, то задание еще не проверено

Pupil desk

Елена Самойлова

Log out

Homework

Marks

[Back](#)

	Алгебра	Информатика
Валерия Гарина	5	-
Елена Самойлова	-	4
Григорий Сафонов	-	-

Страница учителя организована так же, как и ученика. Вкладки всего две - добавить задание и посмотреть список уже добавленных. Каждое задание можно удалить или редактировать

Teacher desk

Дмитрий Лапин

Log out

Add task

Tasks

11.1 История 2020-10-12 | Edit | Delete

11.1 Обществознание 2020-10-12 | Edit | Delete

1A Обществознание 2020-10-14 | Edit | Delete

При переходе на задание получаем детальную информацию и список учеников выполнивших учениками заданий. Если задание проверено, оно пропадает из списка

Teacher desk

Дмитрий Лапин

Log out

Add task

Tasks

Back

1A Обществознание 2020-10-14

• **Deadline:** Oct. 14, 2020 - Oct. 18, 2020

Fines info: „m lv'oks

Text: lm;fl v'fs

• Григорий Сафонов | Check

Decision: б .чс 3