

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

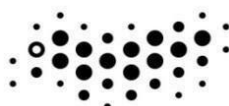
по лабораторной работе 2
по дисциплине Веб-программирование

Автор: Александрин Антон

Факультет: ИКТ

Группа: К33422

Преподаватель: Говоров А.И.



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург 2021

Вариант 2. Доска домашних заданий. О домашнем задании должна храниться следующая информация: предмет, преподаватель, дата выдачи, период выполнения, текст задания, информация о штрафах. Необходимо реализовать следующий функционал: Регистрация новых пользователей. Просмотр домашних заданий по всем дисциплинам (сроки выполнения, описание задания). Сдача домашних заданий в текстовом виде. Администратор (учитель) должен иметь возможность поставить оценку за задание средствами Django-admin. В клиентской части должна формироваться таблица, отображающая оценки всех учеников класса.

Описание модели данных

Модель домашнего задания со всеми необходимыми полями:

```
class Homework(models.Model):
    homework_id = models.IntegerField(blank=False, primary_key=True)
    name = models.CharField(max_length=100, blank=False)
    subject = models.CharField(max_length=30, blank=False)
    teacher = models.CharField(max_length=30, blank=False)
    begin_date = models.DateField(blank=False)
    deadline = models.DateField(blank=False)
    task = models.CharField(max_length=1000)
    penalty = models.CharField(max_length=100)
    students = models.ManyToManyField('Student', through='Assignment')

    def __str__(self):
        return self.name
```

Модель студента (наследуется от AbstractUser, поддерживает аутентификацию):

```
class Student(AbstractUser):
    first_name = models.CharField(max_length=100)
    last_name = models.CharField(max_length=100)
    REQUIRED_FIELDS = ['first_name', 'last_name']

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

Модель конкретного задания (связывает модели студента и домашнего задания):

```
class Assignment(models.Model):
    assignment_id = models.IntegerField(blank=False, primary_key=True)
    student = models.ForeignKey('Student', on_delete=models.CASCADE)
    homework = models.ForeignKey('Homework', on_delete=models.CASCADE)
    grade = models.CharField(default='-', max_length=5, blank=True)
    submission = models.CharField(max_length=1000, blank=True)

    def __str__(self):
        return f'{self.student.first_name} {self.student.last_name}:\
        {self.homework.name}\
        | {"graded" if self.grade != "-" else "submitted" if\
        len(self.submission) else "in process"}'
```

Описание контроллеров в файле views

Функция для регистрации пользователя

```
def register(request):
    if request.method == "POST":
        username = request.POST["username"]
        password = request.POST["password"]
        email = request.POST["email"]
        first_name = request.POST.get("first_name", 'NaN')
        last_name = request.POST.get("last_name", 'NaN')

        confirmation = request.POST["confirmation"]
        if password != confirmation:
            return render(request, "register.html", {
                "message": "passwords do not match"
            })

        try:
            student = Student.objects.create_user(username, email,
password)
            student.first_name = first_name
            student.last_name = last_name
            student.save()
            homeworks = Homework.objects.all()
            for homework in homeworks:
                assignment = Assignment(student=student,
homework=homework)
                assignment.save()
```

```

        except IntegrityError:
            return render(request, "register.html", {
                "message": "username already taken"
            })
        login(request, student)
        return redirect(reverse("homework_list"))
    else:
        return render(request, "register.html")

```

Функции для аутентификации пользователя и выхода из системы, используют стандартные методы Django

```

def log_in(request):
    if request.method == "POST":
        username = request.POST['username']
        password = request.POST['password']

        user = authenticate(request, username=username,
password=password)
        if user is not None:
            login(request, user)
            return redirect(reverse('homework_list'))
        else:
            error_text = 'invalid credentials'

    return render(request, 'login.html', locals())

@login_required
def log_out(request):
    logout(request)
    return redirect(reverse('login'))

```

Функция для отображения оценок всех учеников в виде таблицы

```
@login_required
def class_grades_list(request):
    context = {}
    students = Student.objects.exclude(username="teacher").all()
    context["students"] = students
    context["homeworks"] = Homework.objects.all()
    context["grades"] = {}
    assignments = Assignment.objects.all()
    for homework in context["homeworks"]:
        for assignment in assignments:
            if assignment.homework == homework and
assignment.student.pk != 3:
                if not assignment.student.pk in context["grades"]:
                    context["grades"][assignment.student.pk] = []
                context["grades"][assignment.student.pk].append(
                    assignment.grade)

    return render(request, 'class_grades.html', context)
```

CBV для отображения домашних заданий

```
class HomeworkList(ListView):
    model = Homework
    template_name = 'homework_list.html'

class HomeworkDetail(DetailView):
    model = Homework
    template_name = 'homework_detail.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['form'] = AssignmentForm()
        return context
```

Функция для сдачи домашних заданий преподавателю

```
@login_required
def hand_in(request, pk):
    homework = Homework.objects.get(pk=pk)
    assignment = Assignment.objects.get(student=request.user,
                                         homework=homework)

    form = AssignmentForm(request.POST, instance=assignment)
    if form.is_valid():
        form.save()

    return redirect(reverse('homework_list'))
```

Описание роутеров файла urls

```
urlpatterns = [
    path('register', views.register, name='register'),
    path('login', views.log_in, name='login'),
    path('log_out', views.log_out, name='logout'),
    path('class_grades', views.class_grades_list,
name='class_grades'),
    path('homework', login_required(views.HomeworkList.as_view()),
name='homework_list'),
    path('homework/<int:pk>',
login_required(views.HomeworkDetail.as_view()),
name='homework_detail'),
    path('handin/<int:pk>', login_required(views.hand_in),
name='handin')
]
```

Скрины работы программы

Админ-панель преподавателя с отображением всех заданий и прогресса студентов

<input type="checkbox"/>	ASSIGNMENT
<input type="checkbox"/>	Naruto Uzumaki: Exam in process
<input type="checkbox"/>	Naruto Uzumaki: Linear Algebra in process
<input type="checkbox"/>	Naruto Uzumaki: Matrix operations in process
<input type="checkbox"/>	Naruto Uzumaki: Analysis in process
<input type="checkbox"/>	Naruto Uzumaki: Differential equations in process
<input type="checkbox"/>	Lou Reed: Exam in process
<input type="checkbox"/>	Dima Bobov: Exam submitted
<input type="checkbox"/>	Anton Alexandrin: Exam submitted
<input type="checkbox"/>	Dima Bobov: Matrix operations graded
<input type="checkbox"/>	Lou Reed: Differential equations graded
<input type="checkbox"/>	Lou Reed: Analysis graded
<input type="checkbox"/>	Lou Reed: Matrix operations graded
<input type="checkbox"/>	Lou Reed: Linear Algebra graded
<input type="checkbox"/>	Dima Bobov: Linear Algebra in process
<input type="checkbox"/>	Anton Alexandrin: Linear Algebra in process
<input type="checkbox"/>	Anton Alexandrin: Matrix operations in process
<input type="checkbox"/>	Dima Bobov: Analysis in process
<input type="checkbox"/>	Anton Alexandrin: Analysis in process
<input type="checkbox"/>	Dima Bobov: Differential equations graded
<input type="checkbox"/>	Anton Alexandrin: Differential equations in process

Список заданий:

[Class grades](#) [My homework](#) [Log out](#)

Name	Subject	Given on	Deadline	Teacher	Penalty
Differential equations	Math	Nov. 10, 2021	Dec. 10, 2021	Milovanovich	0 after deadline
Analysis	Math	Nov. 10, 2021	Nov. 24, 2021	Milovanovich	-1 point after deadline
Matrix operations	Math	Nov. 10, 2021	Nov. 24, 2021	Milovanovich	0 after deadline
Linear Algebra	Math	Nov. 10, 2021	Dec. 10, 2021	Milovanovich	0 after deadline
Exam	Math	Dec. 28, 2021	Dec. 28, 2021	Milovanovich	expulsion

Оценки студентов:

[Class grades](#) [My homework](#) [Log out](#)

Student	Differential equations	Analysis	Matrix operations	Linear Algebra	Exam
Anton Alexandrin	-	-	-	-	-
Dima Bobov	3	-	5	-	-
Lou Reed	5	2	0	4	-
Naruto Uzumaki	-	-	-	-	-

Страница домашнего задания:

[Class grades](#) [My homework](#) [Log out](#)

Differential equations

Math

Teacher: Milovanovich

Given on: Nov. 10, 2021

Deadline: Dec. 10, 2021

Five equations

Submission:

[To homework list](#)