

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»**

ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Дисциплина: Основы web-программирования

**Отчет
по лабораторной работе №1
«Реализация web-сервисов средствами Python»**

Вариант 12

Выполнила:
Назаренко Ульяна Кирилловна,
студентка группы К3343

Преподаватель:
Говоров Антон Игоревич

Санкт-Петербург,
2020 г.

Лабораторная работа №1

«Реализация web-сервисов средствами Python»

Цель работы: овладеть практическими навыками и умениями реализации web-сервисов средствами Django 2.2.

Задачи:

1. Реализовать сайт используя фреймворк Django 2.2 для соответствующего варианта практического задания.
2. Построить СУБД PostgreSQL *, в соответствии с практическим заданием.

Программное обеспечение: Python 3.7, Django 2.2, MySQL Workbench.

Практическое задание (Вариант 12): «Доска домашних заданий»

О домашнем задании должна храниться следующая информация: предмет, преподаватель, дата выдачи, период выполнения, текст задания, информация о штрафах.

Необходимо реализовать регистрацию новых пользователей, с целью дальнейшего добавления ими комментариев к домашним заданиям. При добавлении комментариев должны сохраняться информация об комментаторе, важность комментария, тип комментария (вопрос по заданию, найденная ошибка, иное).

Выполнение работы:

1. Модель базы данных «Доска домашних заданий»

В соответствии с вариантом 2 (12) была разработана модель БД, представленная на Рисунке 1.

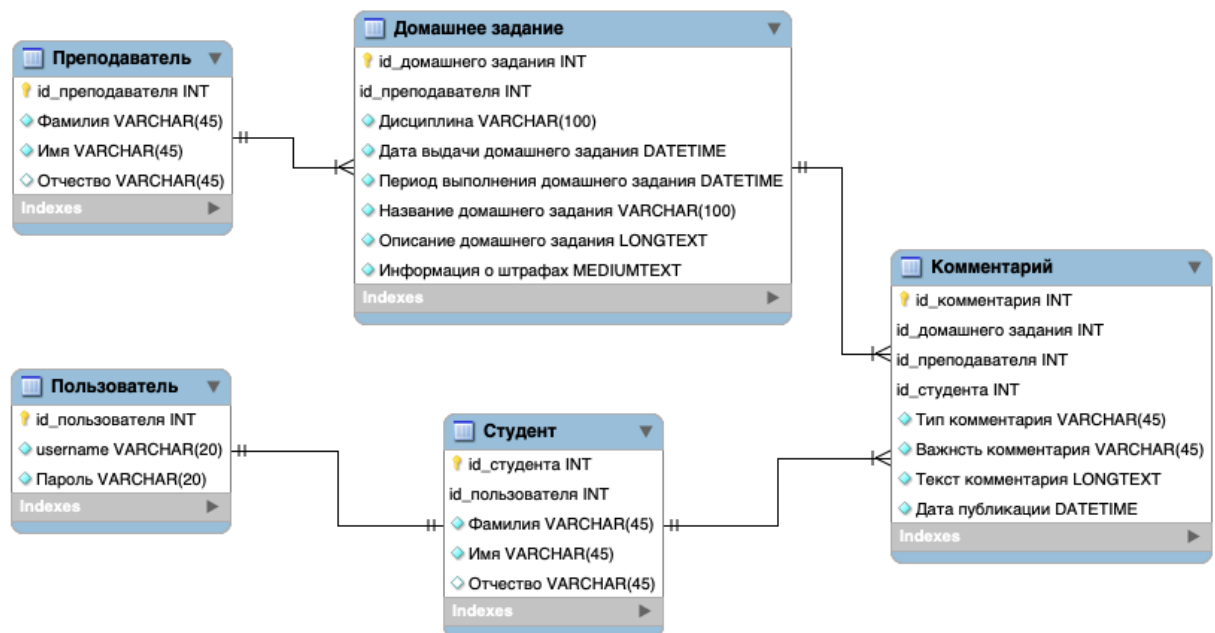


Рисунок 1 – модель базы данных «Доска домашних заданий»

2. Создание проекта и приложения

Создадим Django проект «*лабораторная работа_1*», а также Django приложение «*homework_board*».

3. Создание моделей

Файл **models.py** хранит следующие модели:

- *User_profile* – Студент;
- *Teacher* – Преподаватель;
- *Homework* – Домашнее задание;
- *Comment* – Комментарий.

Модель *User* взята из стандартной библиотеки `django.contrib.auth.models`.

Пример описания модели «*User_profile*»:

```
class User_profile(models.Model):
    id_user = models.OneToOneField(User, on_delete=models.CASCADE)
    surname = models.CharField("Фамилия", max_length=50)
    name = models.CharField("Имя", max_length=50)
    patronymic = models.CharField("Отчество", max_length=50)
    group_number = models.CharField("Номер учебной группы", max_length=5)
```

Рисунок 2 – модель «*User_profile*»

4. Создание админ панели для разработанной модели данных

```
5. from django.contrib import admin
    from django.contrib.auth.admin import UserAdmin
    from django.contrib.auth.models import User
    from .models import User_profile, Teacher, Homework, Comment

    admin.site.register(Teacher)
    admin.site.register(Homework)
    admin.site.register(Comment)

    class UserInline(admin.StackedInline):
        model = User_profile
        can_delete = False

    class UserAdmin(UserAdmin):
        inlines = (UserInline, )

    admin.site.unregister(User)
    admin.site.register(User, UserAdmin)
```

Рисунок 3 – содержимое файла «admin.py»

6. Создание форм

Созданные формы хранятся в файле **forms.py**:

RegisterUserForm – форма для регистрации нового пользователя (*User*);

Register_student – форма для регистрации студента (*User_profile*);

Create_comment – форма для создания комментария.

```
class Register_student(forms.ModelForm):
    class Meta:
        model = User_profile
        fields = [
            "surname",
            "name",
            "patronymic",
            "group_number",
        ]
```

Рисунок 4 – пример формы «Register_student»

7. Создание контроллеров для обработки данных

Представления размещены в файле **views.py**:

- *index* – вывод главной страницы;
- *hw_view* – вывод страницы с домашними заданиями;
- *comments_view* – вывод страницы с комментариями;
- *RegisterUserView* – вывод страницы регистрации, при успешном заполнении направляет на страницу создания аккаунта студента;
- *User_registration* – вывод страницы внесения данных в аккаунт, направляет на главную страницу;
- *Comment_creation* – вывод страницы с созданием комментария.

```
class User_registration(CreateView):
    model = User_profile
    fields = [
        "surname",
        "name",
        "patronymic",
        "group_number",
    ]
    success_msg = 'Данные пользователя успешно внесены'
```

Рисунок 5 – пример представления «User_registration»

Папка **templates** хранит шаблоны, которые отвечают за формирование внешнего вида приложения. Они предоставляют специальный синтаксис, который позволяет внедрять данные в код HTML.

Отмечу, что в данной лабораторной работе для удобства используется популярный фреймворк **Bootstrap4**, а также **crispy-forms**, предоставляющие простое отображение форм **Django** без написания лишнего кода и с минимальными установками, что очень упрощает работу.

Содержание **templates**:

- *index* – главная страница;
- *hw_list* – страница с домашними заданиями;
- *comments_list* – страница с комментариями к домашним заданиям;
- *comment_creation* – страница для созданием комментария;

Там же в папке **registration**:

- *registration* – регистрация пользователя;
- *profile* – внесение данных пользователя;
- *log_in* – страница входа в аккаунт;
- *logged_out* – страница выхода.

Страницы содержат отображения полей, переданных из контроллера, то есть на данные файлы ссылаются представления из файла **views.py**.

8. Адресация

Файл **urls.py** содержит пути для доступа к страницам.

```
urlpatterns = [
    path('', views.index, name='index'),
    path('hw_list', views.hw_view, name='hwlist'),
    path('profile', User_registration.as_view, name='profile'),
    path('comment_creation', Comment_creation.as_view,
name='commentcreation'),
    path('comments_list', views.comments_view, name='commentslist'),
    path('login/', view.LoginView.as_view(), name='login'),
    path('logout/', view.LogoutView.as_view(), name='logout'),
    path('register', views.RegisterUserView.as_view(), name='register'),
]
```

Рисунок 6 – пути к страницам в файле «urls.py»

9. Полученные интерфейсы

1) Главная страница

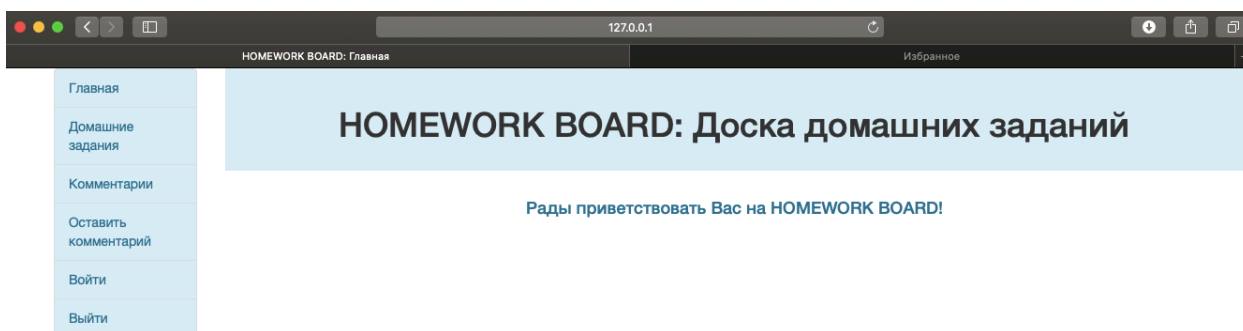


Рисунок 7– Главная страница сервиса «Homework Board»

2) Страница с домашними заданиями

На данной странице отображены все домашние задания и соответствующая информация о них.

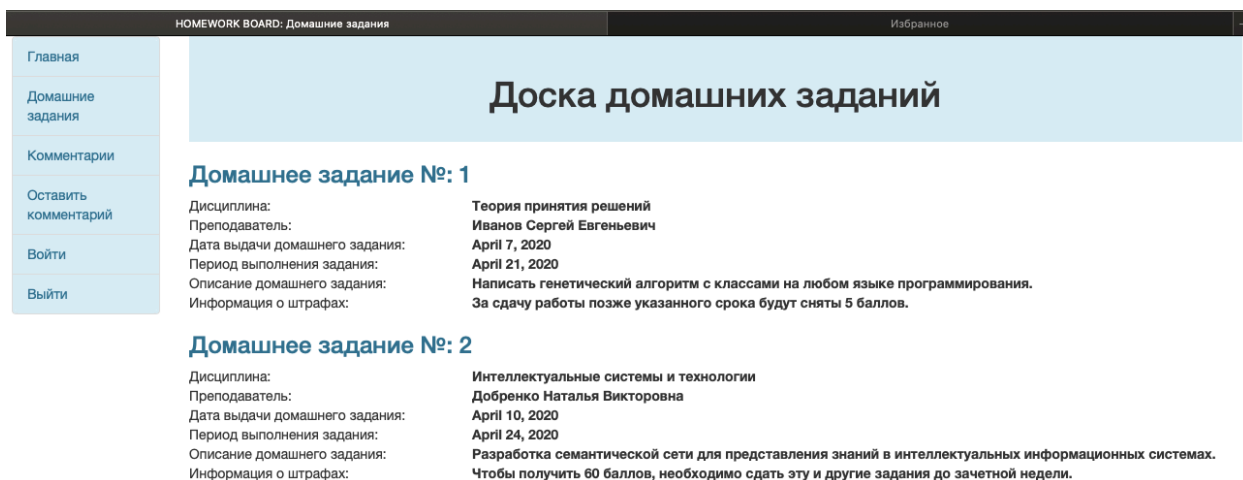


Рисунок 8 – страница сервиса «Homework Board» с домашними заданиями

3) Страница с комментариями к домашним заданиям

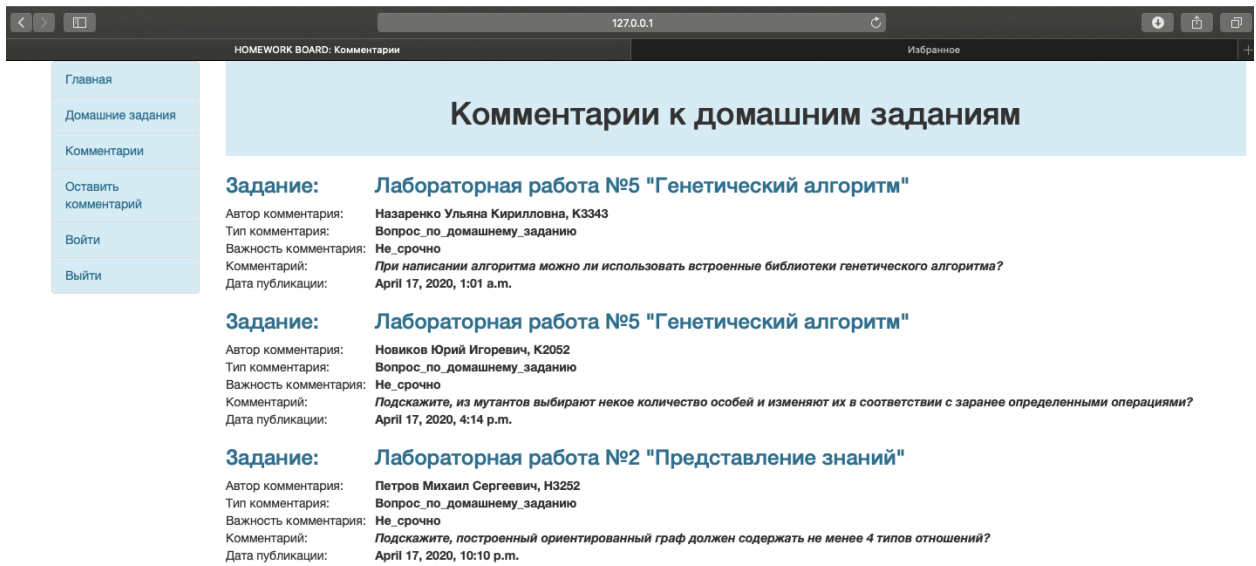


Рисунок 9 – страница сервиса «Homework Board» с комментариями к домашним заданиям

4) Страница входа в аккаунт

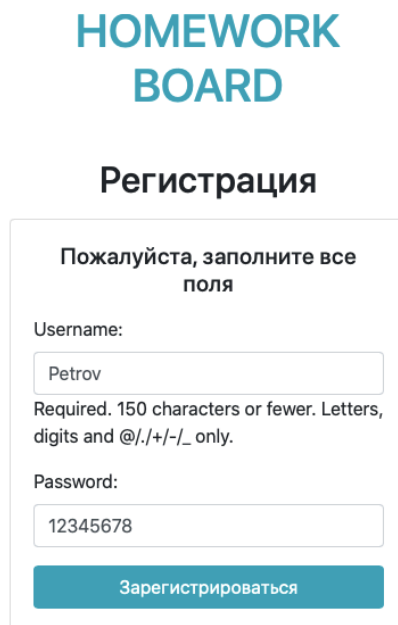
После входа в аккаунт, пользователь будет перенаправлен на главную страницу.

The screenshot shows the login page of the 'Homework Board' service. The page has a light blue header with the text 'HOMEWORK BOARD'. The main content area is titled 'Вход в рабочий кабинет'. It contains two input fields: 'Username*' and 'Password*'. Below the input fields is a blue button labeled 'Войти'. At the bottom, there is a link that says 'Нет рабочего кабинета? Зарегистрироваться!'.

Рисунок 10 – страница входа в аккаунт сервиса «Homework Board»

5) Страница регистрации нового пользователя

После корректного заполнения полей, пользователю необходимо ввести личные данные.



**HOMEWORK
BOARD**

Регистрация

Пожалуйста, заполните все поля

Username:

Petrov

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

Password:

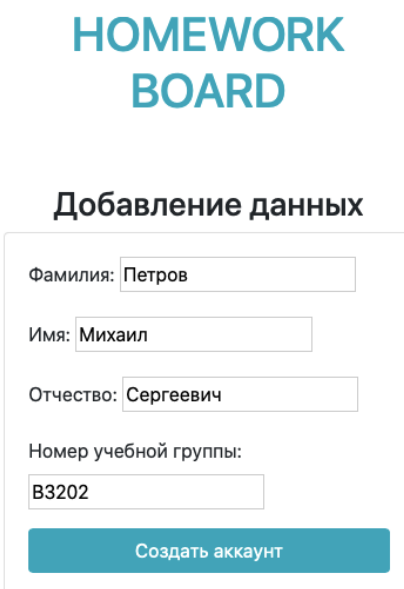
12345678

Зарегистрироваться

Рисунок 11 – страница регистрации аккаунта сервиса «Homework Board»

6) Страница добавления личных данных студента

После корректного ввода всех данных, пользователь будет успешно зарегистрирован и перенаправлен на главную страницу.



**HOMEWORK
BOARD**

Добавление данных

Фамилия: Петров

Имя: Михаил

Отчество: Сергеевич

Номер учебной группы:

В3202

Создать аккаунт

Рисунок 12 – страница добавления данных студента на «Homework Board»

7) Страница выхода из аккаунта

После выхода из аккаунта у пользователя есть возможность заново войти в аккунт.

HOMEWORK BOARD

До скорой встречи!

[Войти в свой аккаунт](#)

Рисунок 13 – страница добавления данных студента на «Homework Board»

8) Страница с созданием комментария к домашним заданиям

Зарегистрированные пользователи могут написать комментарий к домашнему заданию в следующей форме:

Оставить комментарий

Укажите домашнее задание, к которому хотите оставить комментарий:

Лабораторная работа №2 "Представление знаний" ▾

Выберите тип комментария: Вопрос По Домашнему Заданию ▾

Укажите важность комментария: Не Срочно ▾

Напишите свой комментарий:

Подскажите, построенный ориентированный граф должен содержать не менее 4 типов отношений?

Опубликовать

Рисунок 14 – страница сервиса «Homework Board» с оставлением комментария

10. Описание работы с веб-сервисом «Homework Board»

Данным сервисом могут пользоваться как зарегистрированные пользователи, так и люди без созданных аккаунтов. Все пользователи могут просматривать домашние задания и соответствующую информацию к ней, а также видеть комментарии, оставленные авторизованными пользователями. Пользователи без аккаунта могут легко зарегистрироваться. После регистрации и входа в аккаунт, пользователю предоставляется возможность писать комментарии к домашним заданиям.

Выводы:

В ходе данной лабораторной работы были получены практические навыки и умения реализации web-сервисов средствами Django 2.2. Был реализован сайт, по средствам использования фреймворка Django 2.2 для варианта 12 («Доска домашних заданий») практического задания, была построена модель базы данных «Доска домашних заданий» в MySQLWorkbench, а также был использован Bootstrap4 для удобства оптимизации работы над созданием сайта.