Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ”

Лабораторная работа №1

по основам web-программирования

на тему «Реализация web-сервисов средствами Python»

Выполнила: Бакирова Ш. Д.

Группа: К3340

Преподаватель: Говоров А. И.

Санкт-Петербург

2020 г.

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

Программное обеспечение: Python 3.6, Django 2.2, PostgreSQL

Практическое задание:

Реализовать сайт используя фреймворк Django 2.2 и СУБД PostgreSQL *, в соответствии с вариантом задания лабораторной работы.

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.

Выполнение работы:

1. Модель базы данных

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

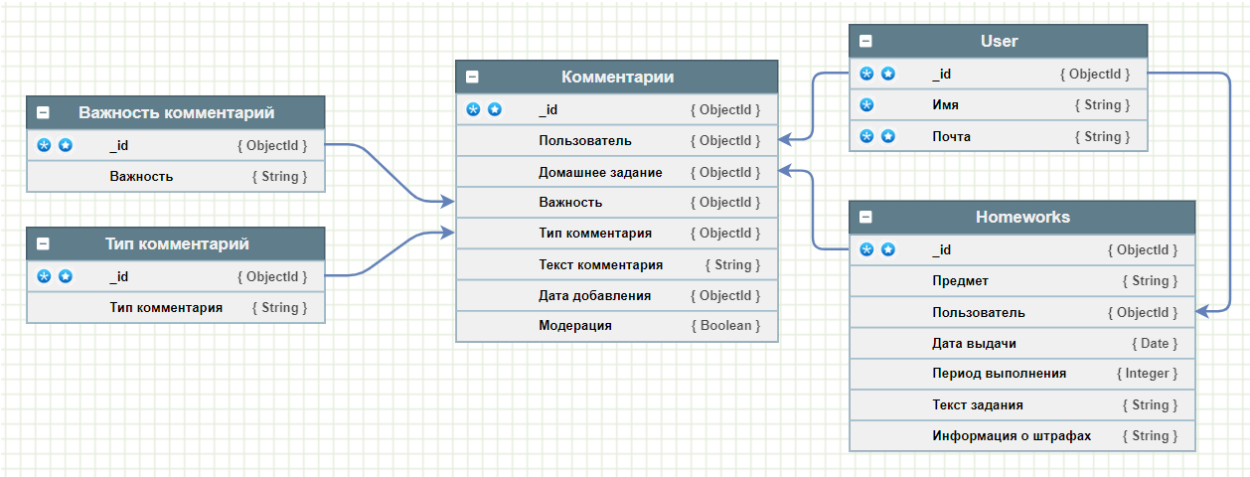

Рисунок 1. Модель базы данных.

Основные сущности и их атрибуты:

1. Домашние задания: предмет, преподаватель, дата выдачи, период выполнения, текст задания, информация о штрафах

2. Комментарии: пользователь, домашнее задание, важность, тип комментария, комментарий, дата добавления, модерация
3. Важность комментарий: важность
4. Тип комментария: тип
5. Пользователь: логин, электронная почта, пароль

2. Создание моделей в Django

ImportanceComments (важность комментария):

```python
class ImportanceComments(models.Model):
    """Класс важности комментария
    """
    title = models.CharField("Важность", max_length=50)

    class Meta:
        verbose_name = "Важность комментария"
        verbose_name_plural = "Важность комментариев"

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

TypeComments (Тип комментария):

```python
class TypeComments(models.Model):
    """Класс типа комментария
    """
    title = models.CharField("Тип", max_length=50)

    class Meta:
        verbose_name = "Тип комментария"
        verbose_name_plural = "Типы комментариев"

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

Homeworks (Домашние задания):

```python
class Homeworks(models.Model):
    """Класс домашних заданий
    """
    title = models.CharField("Предмет", max_length=100)
    user = models.ForeignKey(
        User,
        verbose_name="Пользователь",
        on_delete=models.CASCADE)
    created = models.DateTimeField("Дата выдачи", auto_now_add=True)
    duration = models.DurationField("Период выполнения")
    text = models.TextField("Текст задания")
    fines = models.CharField("Информация о штрафах", max_length=100)

    class Meta:
        verbose_name = "Домашнее задание"
        verbose_name_plural = "Домашние задания"

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

Comments (Комментарии):

```python
class Comments(models.Model):
    """Ксласс комментариев к новостям
    """
    user = models.ForeignKey(
        User,
        verbose_name="Пользователь",
        on_delete=models.CASCADE)
    homework = models.ForeignKey(
        Homeworks,
        verbose_name="Домашнее задание",
        on_delete=models.CASCADE)
    importance = models.ForeignKey(
        ImportanceComments,
        verbose_name="Важность комментария",
        on_delete=models.CASCADE)
    type = models.ForeignKey(
        TypeComments,
        verbose_name="Тип комментария",
        on_delete=models.CASCADE)
    text = models.TextField("Комментарий")
    created = models.DateTimeField("Дата добавления", auto_now_add=True, null=True)
    moderation = models.BooleanField("Модерация", default=False)
```

```python
class Meta:
    verbose_name = "Комментарий"
    verbose_name_plural = "Комментарии"

def __str__(self):
    return "{}".format(self.user)
```

Модель User взята из стандартной библиотеки «django.contrib.auth.get_user_model»

3. Создание админ панели

Содержимое файла admin.py:

```python
class HomeworkAdmin(SummernoteModelAdmin):
    """ Доска домашних заданий
    """
    list_display = ("title", "user", "created", "duration")
    list_editable = ("user",)
    search_fields = ["title", "user__username"]
    list_filter = ("user", "created")
    summer_note_fields = 'text'


class CommentAdmin(admin.ModelAdmin):
    """ Комментарии
    """
    list_display = ('user', 'homework', 'importance', 'created', 'type', 'moderation')


admin.site.register(Homeworks, HomeworkAdmin)
admin.site.register(Comments, CommentAdmin)
admin.site.register(ImportanceComments)
admin.site.register(TypeComments)
```

4. Создание форм

Форма для создания комментариев к статьям. Хранится в forms.py.

```python
class CommentForm(ModelForm):
    """Форма комментариев к статьям
    """

    class Meta:
        model = Comments
        fields = ('importance', 'type', 'text')
```

5. Создание контролеров для обработки данных

Представления размещены в файле view.py.

homework_list – представление для вывода всех домашних заданий

```python
def homework_list(request):
    """Вывод всех домашних заданий
    """
    homeworks = Homeworks.objects.all()
    return render(request, "homeworks/homework_list.html", {"homeworks": homeworks})
```

homework_single – представление для вывода и добавления комментариев к данному домашнему заданию.

```python
def homework_single(request, pk):
    """Вывод полного дз
    """
    homework: object = get_object_or_404(Homeworks, id=pk)
    comments = Comments.objects.filter(homework=pk, moderation=True)
    if request.method == "POST":
        form = CommentForm(request.POST)
        if form.is_valid():
            form = form.save(commit=False)
            form.user = request.user
            form.homework = homework
            form.save()
            return redirect(homework_single, pk)
    else:
        form = CommentForm()
    return render(request, "homeworks/homework_single.html",
                  {"homework": homework,
                   "comments": comments,
                   "form": form})
```

6. Внешний вид приложения

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

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

Содержание templates:

- base.html – базовая страница для отображения кнопок входа/выхода и перехода на регистрацию установленного модуля allauth

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>{% block title %}{% endblock title %}</title>
</head>
<body>
            {% if user.is_active %}
                <i class="mr-3">Вы вошли как <b>{{ request.user }}</b>,</i>
                <a href="/accounts/logout/" class="btn btn-danger">Выйти</a>
            {% else %}
                <a href="/accounts/login/" class="btn btn-primary">Войти</a>
            {% endif %}
    <div class="page">
        <div class="main">
            {% block content %}
            {% endblock content %}
        </div>
    </div>
</body>
</html>
```

- homeworks
    - base.html – базовая страница которая ссылается на предыдущую и выполняет роль структурирования страниц

```
{% extends 'base.html' %}

{% block title %}

{% endblock title %}

{% block search %}

{% endblock search %}

{% block content %}

{% endblock content %}
```

    - homework_list.html – страница для отображения всех домашних заданий

```
{% extends 'homeworks/base.html' %}

{% block content %}
    {% for hm in homeworks %}
        <h2> {{ hm.title }}</h2>
        <p>Описание задания: {{ hm.text|safe }}</p>
        <p>Дата выдачи: {{ hm.created }}</p>
        <p>Срок выполнения: {{ hm.duration }}</p>
        <p>Информация о штрафах: {{ hm.fines }}</p>
        <a href="{% url 'homework_single' pk=hm.id %}">комментарии</a>
    {% endfor %}
{% endblock content %}
```

- homeworl_single.html – страница для комментарий и формы для добавления комментарий

```
{% extends 'homeworks/base.html' %}
{% block title %}{{ new.title }}{% endblock title %}

{% block content %}

    <h4>Комментарии</h4>
     {% for comment in comments %}
        Пользователь - {{ comment.user }}<br>
        {{ comment.text }} <br>
        Важность - {{ comment.importance }}<br>
        Тип - {{ comment.type }} <br>
        Добавлен - {{ comment.created }}<br><br>
     {% endfor %}

    {% if user.is_active %}
        <form action="" method="post">
            {% csrf_token %}
            {{ form.as_p }}
            <button type="submit">Отправить</button>
        </form>
    {% else %}
        <h4>Что бы оставить комментарий авторизуйтесь</h4>
    {% endif %}
{% endblock content %}
```

7. Адресация

Пути для доступа к страницам содержаться в urls.py

admin/ – страница для редактирования данных с правами администратора

summernote/ – для удобного редактирования текста

‘’ – домашняя страница

accounts/ - для регистрации и входа

```
urlpatterns = [
                  path('admin/', admin.site.urls),
                  path('summernote/', include('django_summernote.urls')),
                  path('', include("homework.urls")),
                  path('accounts/', include('allauth.urls')),
              ] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

8. Полученные интерфейсы

Вход в приложение:

Messages:

- Вы вышли.

Menu:

- Sign In
- Sign Up

Войти

Если у вас ещё нет учётной записи, пожалуйста, сначала зарегистрируйтесь.

Имя пользователя: bakirova

Пароль: ••••••••

Запомнить меня: ☐

Забыли пароль? Войти

Регистрация в приложении:

Menu:

- Sign In
- Sign Up

Регистрация

Уже зарегистрированы? Войдите.

Имя пользователя: Имя пользователя

E-mail (опционально): E-mail адрес

Пароль: Пароль

Пароль (еще раз): Пароль (еще раз)

Регистрация »

Главная страница:

*Вы вошли как **bakirova**,* Выйти

Абаеведение

Описание задания:

Написать оду Абаю

Дата выдачи: 29 сентября 2020 г. 17:19

Срок выполнения: 3 days, 0:00:00

Информация о штрафах: За невыполнение в срок все баллы обнуляются

комментарии

Английский язык

Описание задания:

Перевести 10 страниц Гарри Поттера. Философский камень.

Дата выдачи: 29 сентября 2020 г. 17:22

Срок выполнения: 1 day, 0:00:00

Информация о штрафах: нет

комментарии

Комментарии к домашнему заданию для авторизованных пользователей добавление комментариев:

Комментарии

Пользователь - admin
А можно переводить другую книгу?
Важность - Не срочно. Важно.
Тип - Вопрос по заданию
Добавлен - 29 сентября 2020 г. 17:28

Пользователь - bakirova
Ничего не работает!
Важность - Не срочно. Не важно.
Тип - Найденная ошибка
Добавлен - 29 сентября 2020 г. 19:46

Пользователь - bakirova
Так себе задание...
Важность - Не срочно. Не важно.
Тип - Другое
Добавлен - 30 сентября 2020 г. 11:54

Важность комментария: ---------

Тип комментария: ---------

Комментарий:

Отправить

Для неавторизованных пользователей нет возможности добавления комментариев

Войти

Комментарии

Пользователь - admin
А можно переводить другую книгу?
Важность - Не срочно. Важно.
Тип - Вопрос по заданию
Добавлен - 29 сентября 2020 г. 17:28

Пользователь - bakirova
Ничего не работает!
Важность - Не срочно. Не важно.
Тип - Найденная ошибка
Добавлен - 29 сентября 2020 г. 19:46

Пользователь - bakirova
Так себе задание...
Важность - Не срочно. Не важно.
Тип - Другое
Добавлен - 30 сентября 2020 г. 11:54

Что бы оставить комментарий авторизуйтесь

Выводы:

В ходе 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 2 («Доска домашних заданий») практического задания, была построена модель базы данных в PostgreSQL.