Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Отчет по лабораторной работе №6**

Дисциплина: «Разработка профессиональных приложений»

WEB - разработка

Вариант № 14

Выполнил:

студент группы ИВТАПбд-21

Мухаметзянов Т. А.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

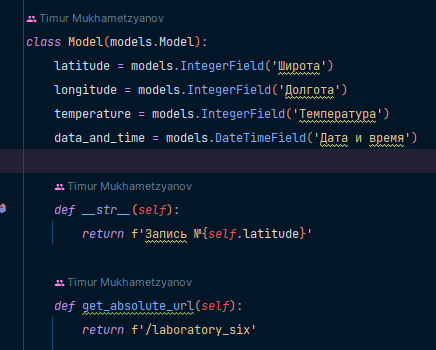
**Задание по варианту**

Необходимо реализовать web-интерфейс для просмотра всех объектов, хранящихся в базе данных. Предметная область - из лабораторной работы 3 (4). Для доступа к данным желательно использовать ORM peewee (возможны альтернативные библиотеки). Для реализации web-части - cerrypy. Ajax использовать необязательно, можно сделать все на статических объектах.

Добавление / изменение необходимо реализовать минимум для 1 таблицы БД с внешними ключами.

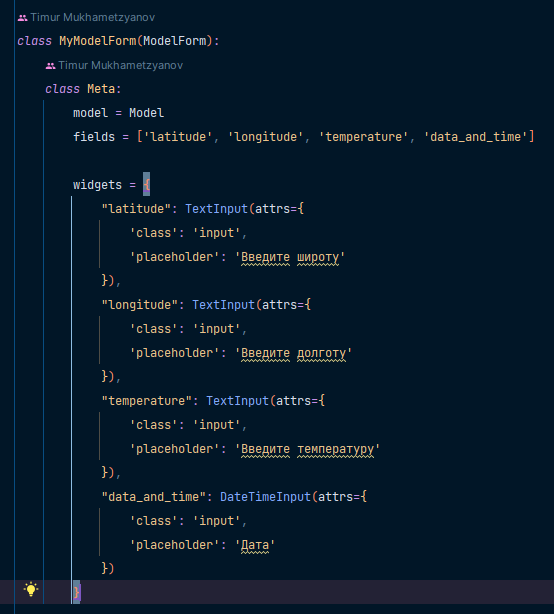
**Описание реализации**

«models.py»:   
Для создания таблицы «Температурные осадки» используется класс Model. В нем указываются поля таблицы с их атрибутами.

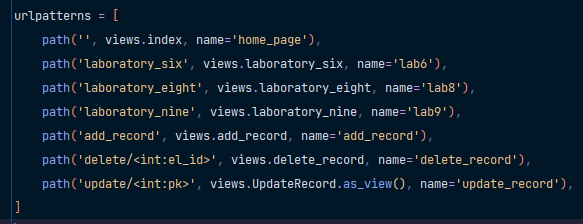


«forms.py»:

В данном файле мы описываем как будет выглядеть форма для создания таблицы.

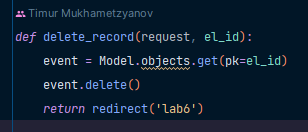


«urls.py»:  
В данном файле мы указываем по какому url будет выполнятся та или иная функция из файла «views.py»

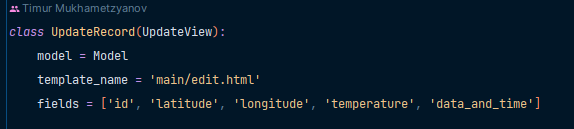


«views.py» функции для добавления записи в бд, сначала мы находим форму для добавления, и если она валидна, то сохраняем ее и редеректим пользователя на страницу с таблицей, если нет то происходит вывод ошибки

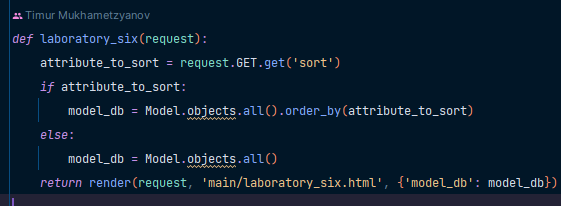


«views.py» функции для удаления записи из бд, получаем id объекта и удаляем его  


«views.py» класс для редактирования записей, получаем модель и поля, дальше используем это в форме для редактирования



«views.py» функция для сортировки наших данных по всем параметрам, получаем атрибут сортировки, который записывается в нашей ссылки <a> в шаблоне, дальше сортируем по нему все объекты с помощью order\_by



Для отображения данных на сайте используется встроенный шаблонизатор jinja. Создается файл шаблон (то что будет отображаться на каждой странице сайта, к примеру шапка сайта) layout.html. Затем создаются html файлы других страниц сайта и с помощью extends наследуется шаблон из файла «layout.html»  


Миграции создаются командой в терминале:  
py manage.py migrations  
py manage.py migrate

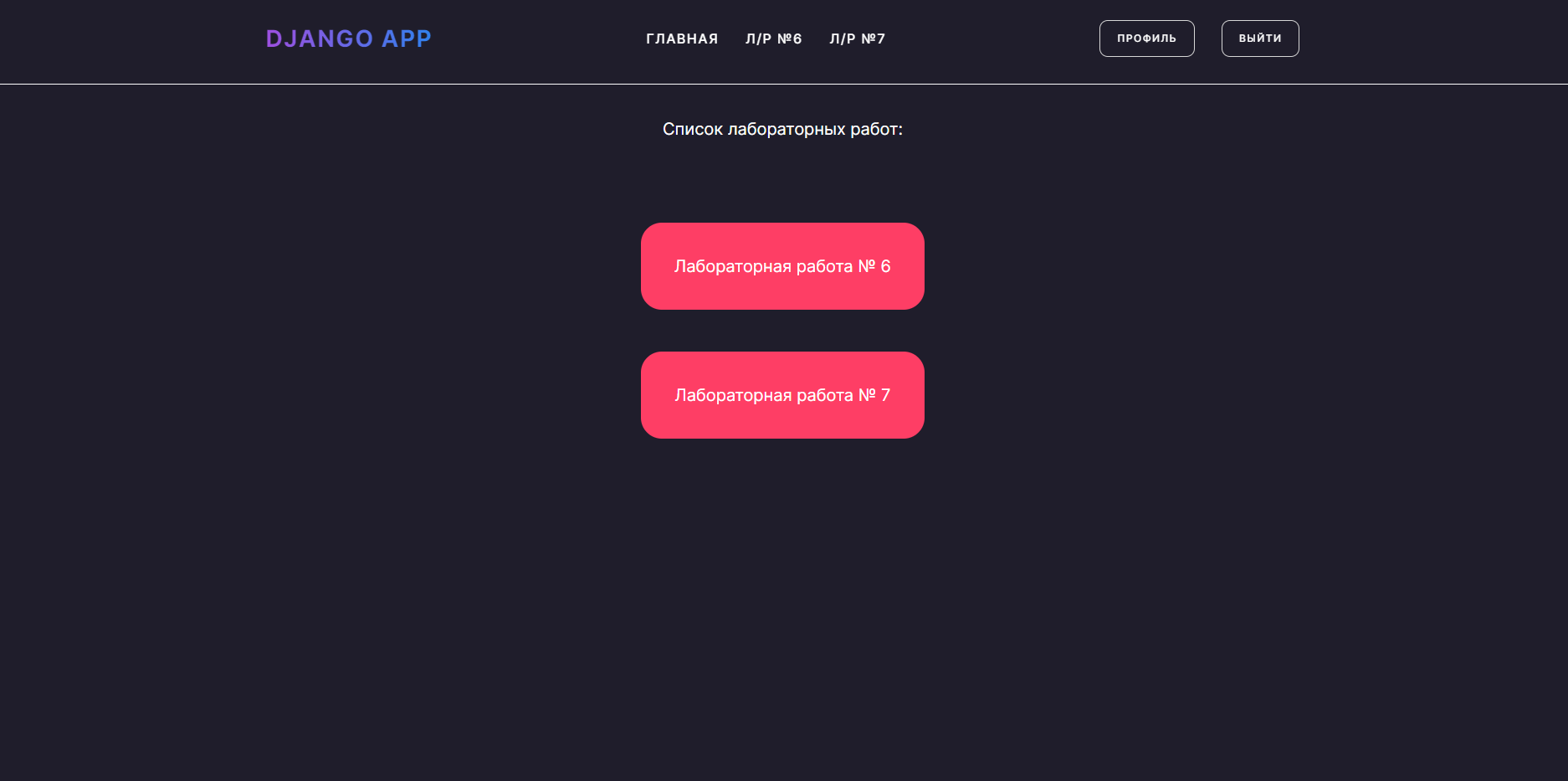
Создание суперпользователя:  
py manage.py createsuperuser

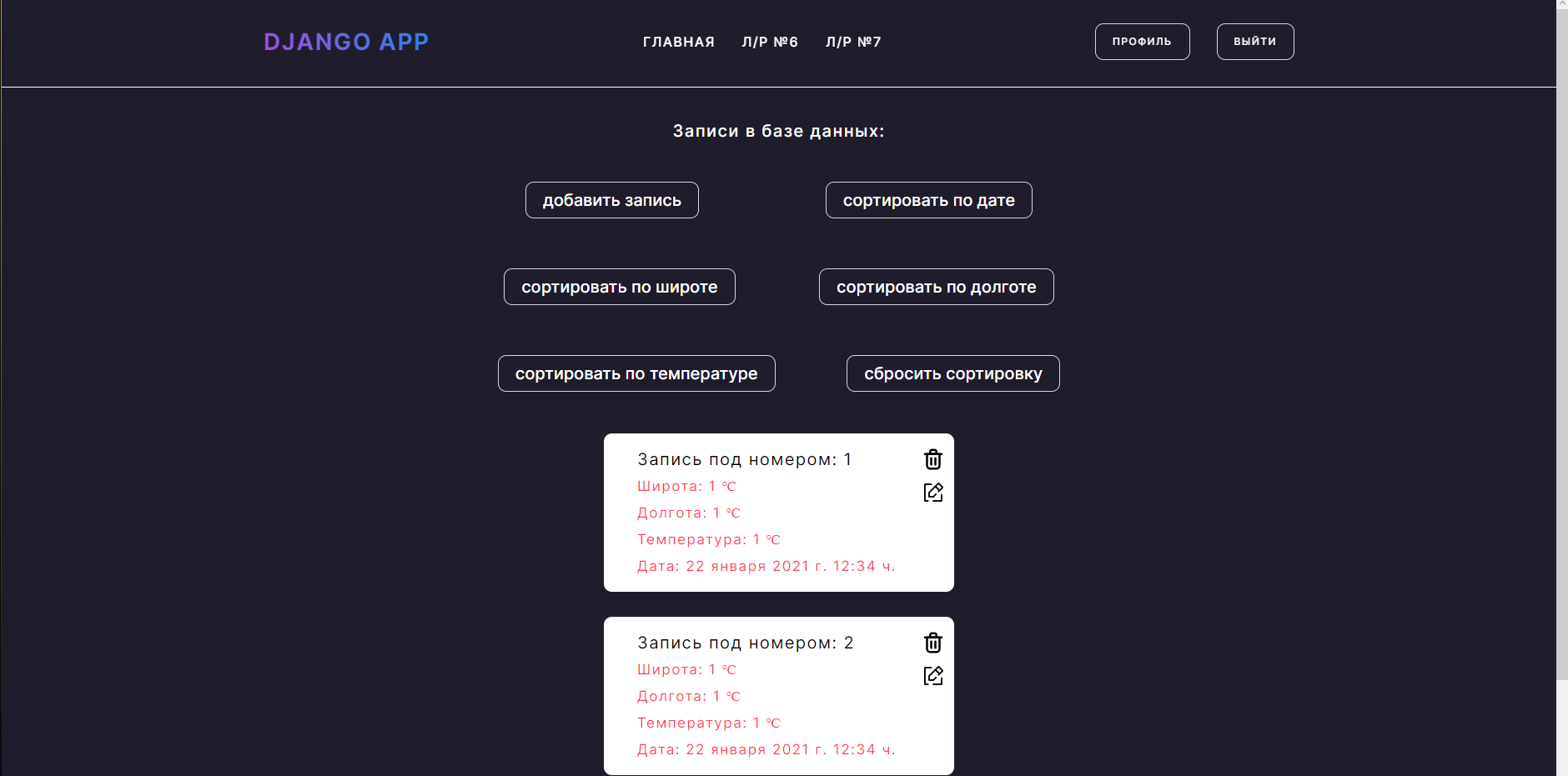
Запустить проект:  
py manage.py runserver

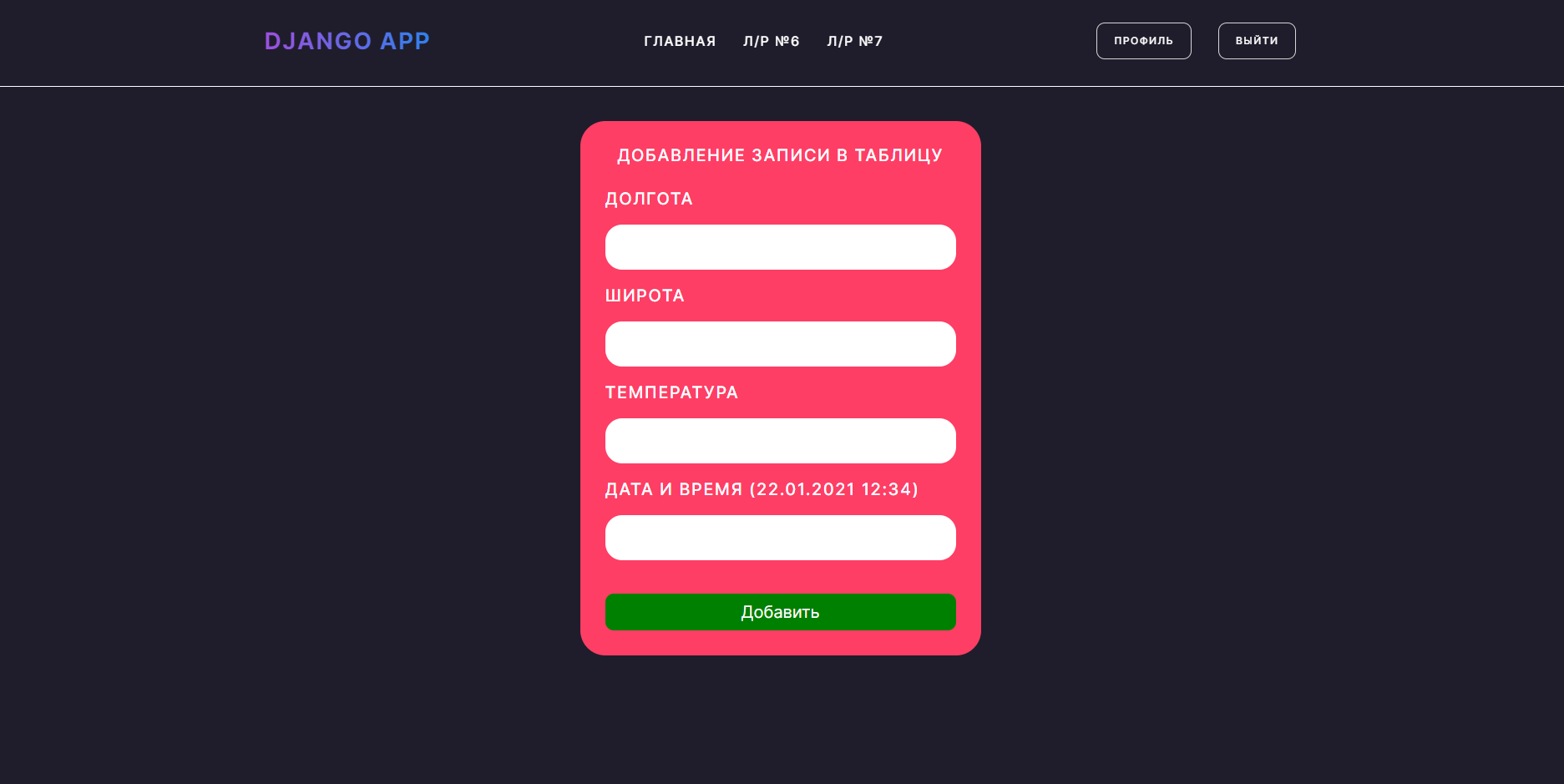
**Описание возникших затруднений**

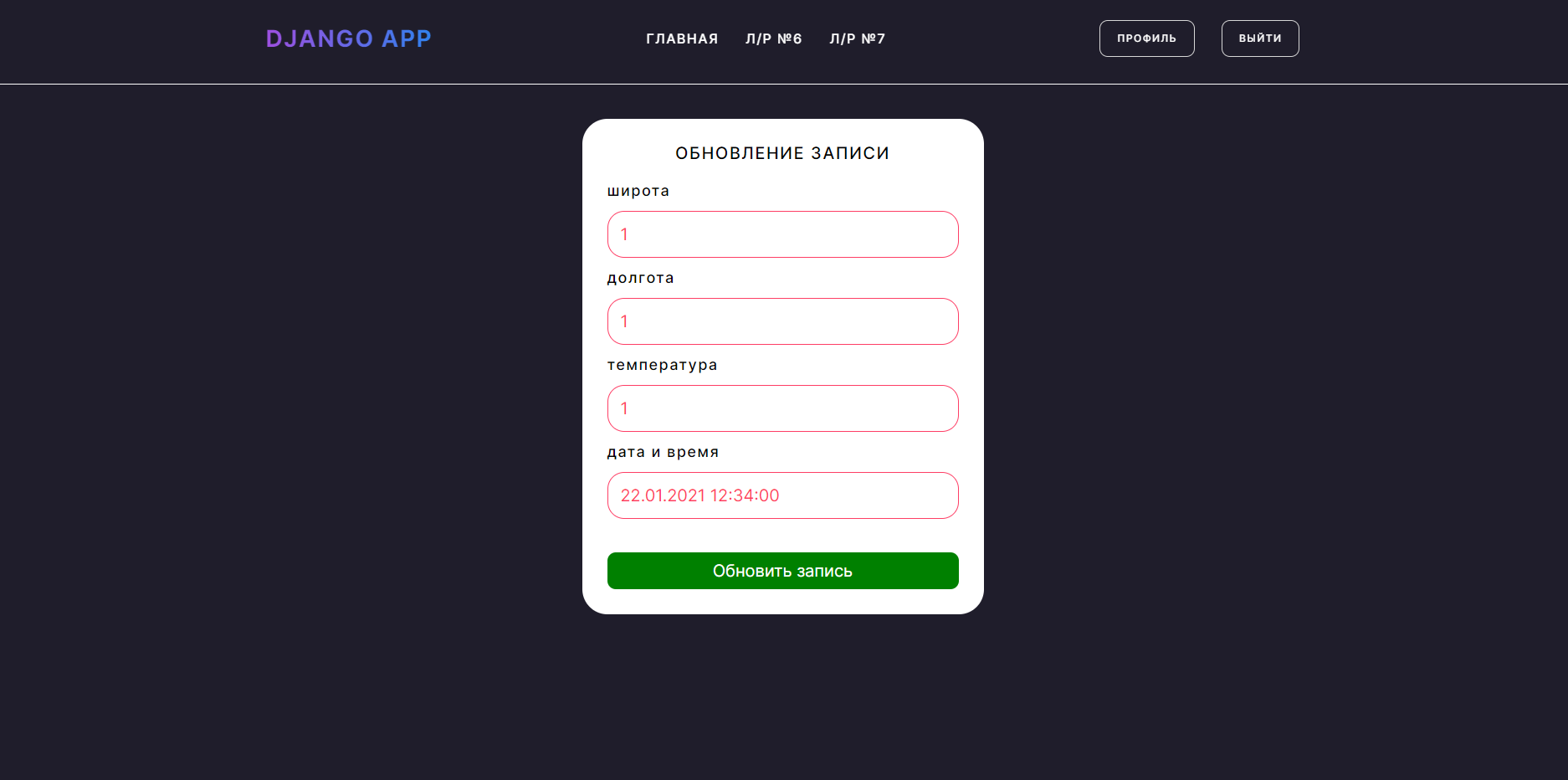
В ходе выполнения данной лабораторной работы затруднений не возникло

**Тестирование приложения**

**Рис. 1 «Главная страница»**

 **Рис. 2 «Страница с записями из 6 лабы, также кнопки для сортировки, удаления и добавления»**

 **Рис. 3 «Форма добавления записи»**

 **Рис 4. «Редактирование записи»**

**Приложение (html файл)**

{% extends 'main/layout.html' %}  
  
{% block title %}  
 Лабораторная работа №6  
{% endblock %}  
  
{% block content %}  
 <h2 *class*="title\_model">Записи в базе данных:</h2>  
 <div *class*="edit\_block">  
 <a *class*="edit\_btn add" *href*="{% url 'add\_record' %}">Добавить запись</a>  
 <a *class*="edit\_btn edit" *href*="{% url 'lab6' %}?sort=data\_and\_time">Сортировать по Дате</a>  
 <a *class*="edit\_btn edit" *href*="{% url 'lab6' %}?sort=latitude">Сортировать по Широте</a>  
 <a *class*="edit\_btn edit" *href*="{% url 'lab6' %}?sort=longitude">Сортировать по Долготе</a>  
 <a *class*="edit\_btn edit" *href*="{% url 'lab6' %}?sort=temperature">Сортировать по Температуре</a>  
 <a *class*="edit\_btn edit" *href*="{% url 'lab6' %}?sort=id">Сбросить сортировку</a>  
 </div>  
 <div *class*="model\_content">  
 {% if model\_db %}  
 {% for el in model\_db %}  
 <div *class*="model">  
 <a *class*="model\_btn delete" *href*="{% url 'delete\_record' el.id %}">  
 <svg *viewBox*="0 0 24 24" *fill*="none" *xmlns*="http://www.w3.org/2000/svg">  
 <path *d*="M10 11V17" *stroke*="#000000" *stroke-width*="2" *stroke-linecap*="round"  
 *stroke-linejoin*="round"/>  
 <path *d*="M14 11V17" *stroke*="#000000" *stroke-width*="2" *stroke-linecap*="round"  
 *stroke-linejoin*="round"/>  
 <path *d*="M4 7H20" *stroke*="#000000" *stroke-width*="2" *stroke-linecap*="round"  
 *stroke-linejoin*="round"/>  
 <path *d*="M6 7H12H18V18C18 19.6569 16.6569 21 15 21H9C7.34315 21 6 19.6569 6 18V7Z" *stroke*="#000000"  
 *stroke-width*="2" *stroke-linecap*="round" *stroke-linejoin*="round"/>  
 <path *d*="M9 5C9 3.89543 9.89543 3 11 3H13C14.1046 3 15 3.89543 15 5V7H9V5Z" *stroke*="#000000"  
 *stroke-width*="2" *stroke-linecap*="round" *stroke-linejoin*="round"/>  
 </svg>  
 </a>  
 <a *class*="model\_btn edit" *href*="{% url 'update\_record1' el.id %}">  
 <svg *viewBox*="0 0 24 24" *fill*="none" *xmlns*="http://www.w3.org/2000/svg">  
 <path *d*="M18 10L14 6M18 10L21 7L17 3L14 6M18 10L17 11M14 6L8 12V16H12L14.5 13.5M20 14V20H12M10 4L4 4L4 20H7" *stroke*="#000000" *stroke-width*="1.5" *stroke-linecap*="round" *stroke-linejoin*="round"/>  
 </svg>  
 </a>  
 <h3 *class*="id\_number">Запись под номером: {{el.id}}</h3>  
 <p *class*="latitude">Широта: {{el.latitude}} ℃</p>  
 <p *class*="longitude">Долгота: {{el.longitude}} ℃</p>  
 <p *class*="temperature">Температура: {{el.temperature}} ℃</p>  
 <p *class*="data\_and\_time">Дата: {{el.data\_and\_time}} ч.</p>  
 </div>  
 {% endfor %}  
 {% else %}  
 <p *class*="empty\_model">У вас нет записей</p>  
 {% endif %}  
 {# Комментарий #}  
 </div>  
  
{% endblock %}