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 № 5

Выполнил:

студент группы ИВТАПбд-21

Гаязов Т. Р.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

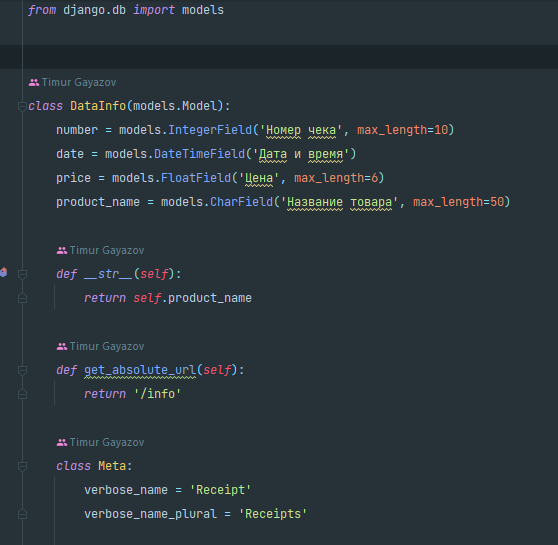
**Задание по варианту**

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 «Чека» мы используем класс DataInfo. В нем указываются поля таблицы с их атрибутами.

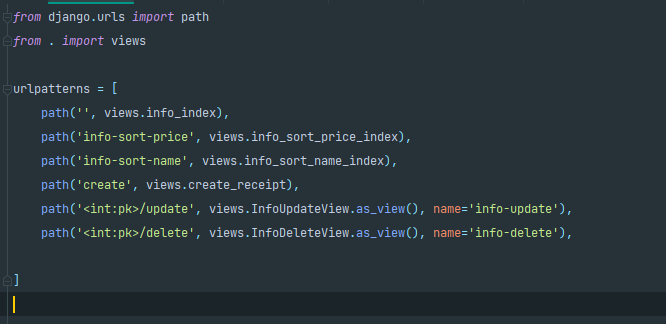
****

«forms.py»:

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



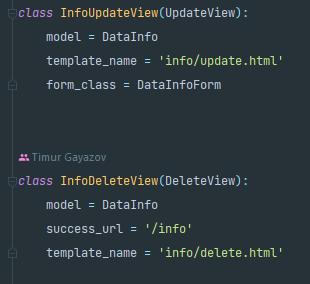
«urls.py»:  
В данном файле мы указываем по какому url будет выполнятся та или инаая функция из файла «views.py»



«views.py» функции для вывода чеков на сайте:  
сортировка происходит благодаря методу order\_by внутри которого мы указываем по какому полю требуется отсортировать данные.  


«views.py» функции для создания чека на сайте:  
в данной функции мы получаем данные из полей формы и заносим их в базу данных.  


«views.py» функции для обновления и удаления чека соответственно:



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

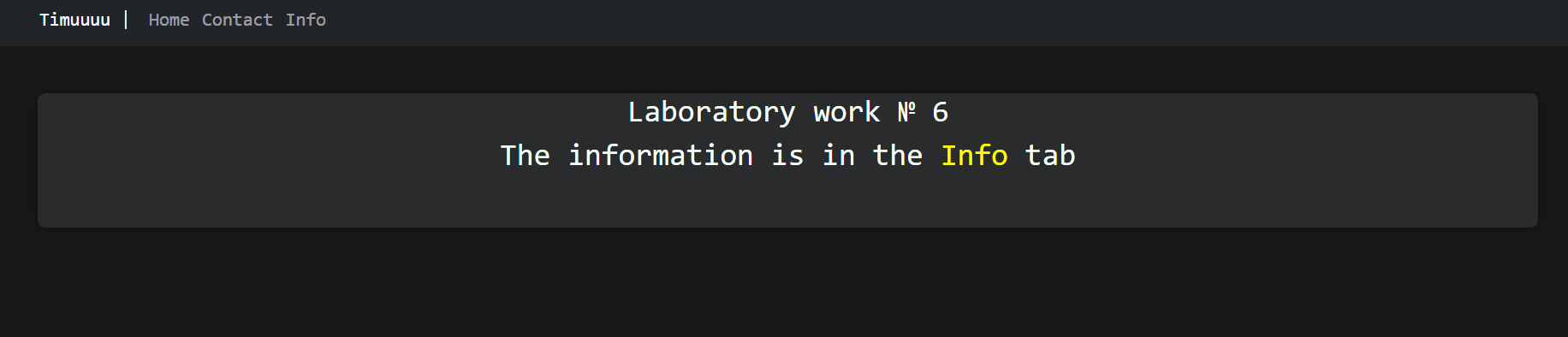
Создание суперпользователя:  
py manage.py createsuperuser

Запустить проект:  
py manage.py runserver

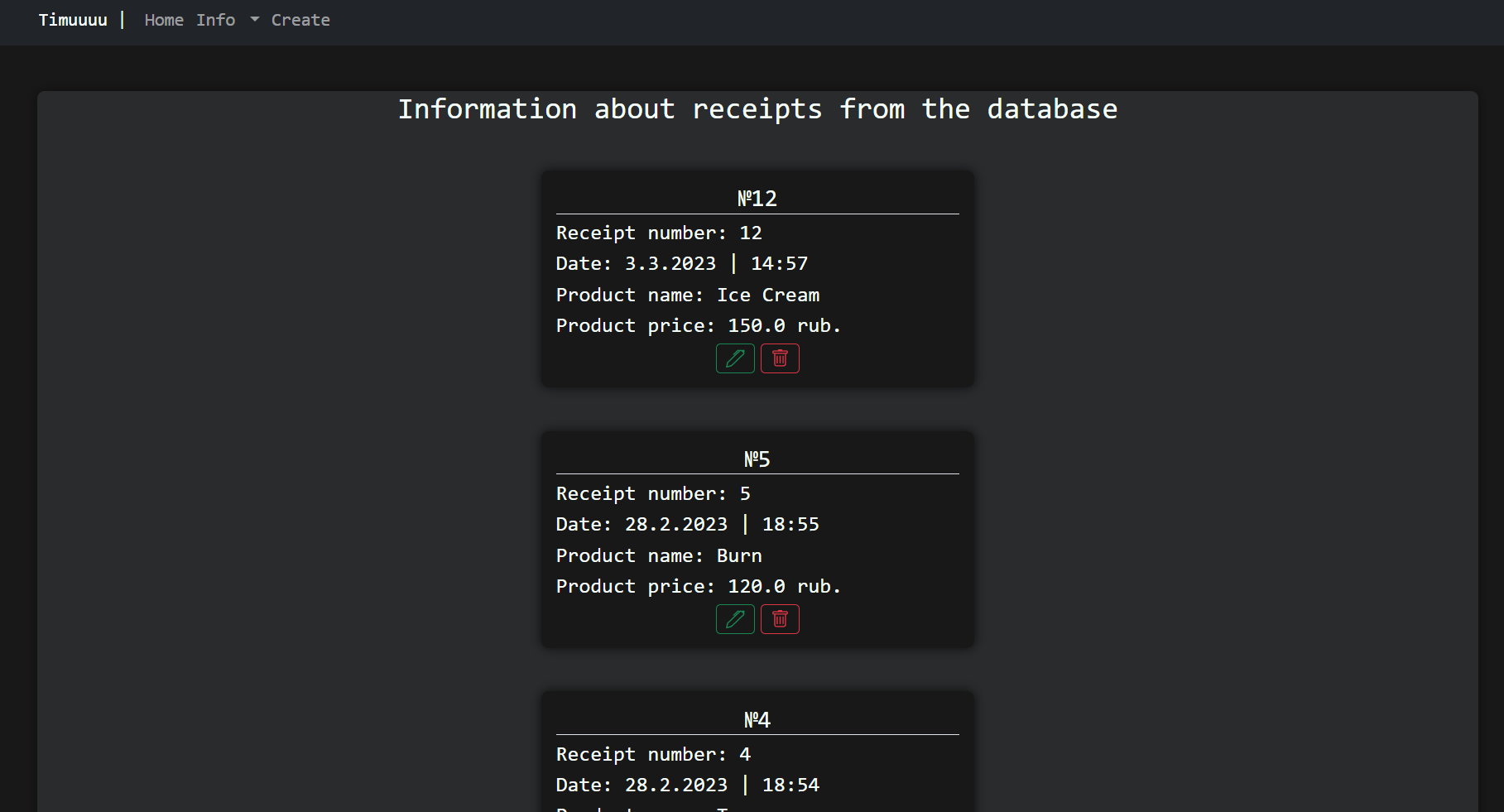
**Описание возникших затруднений**

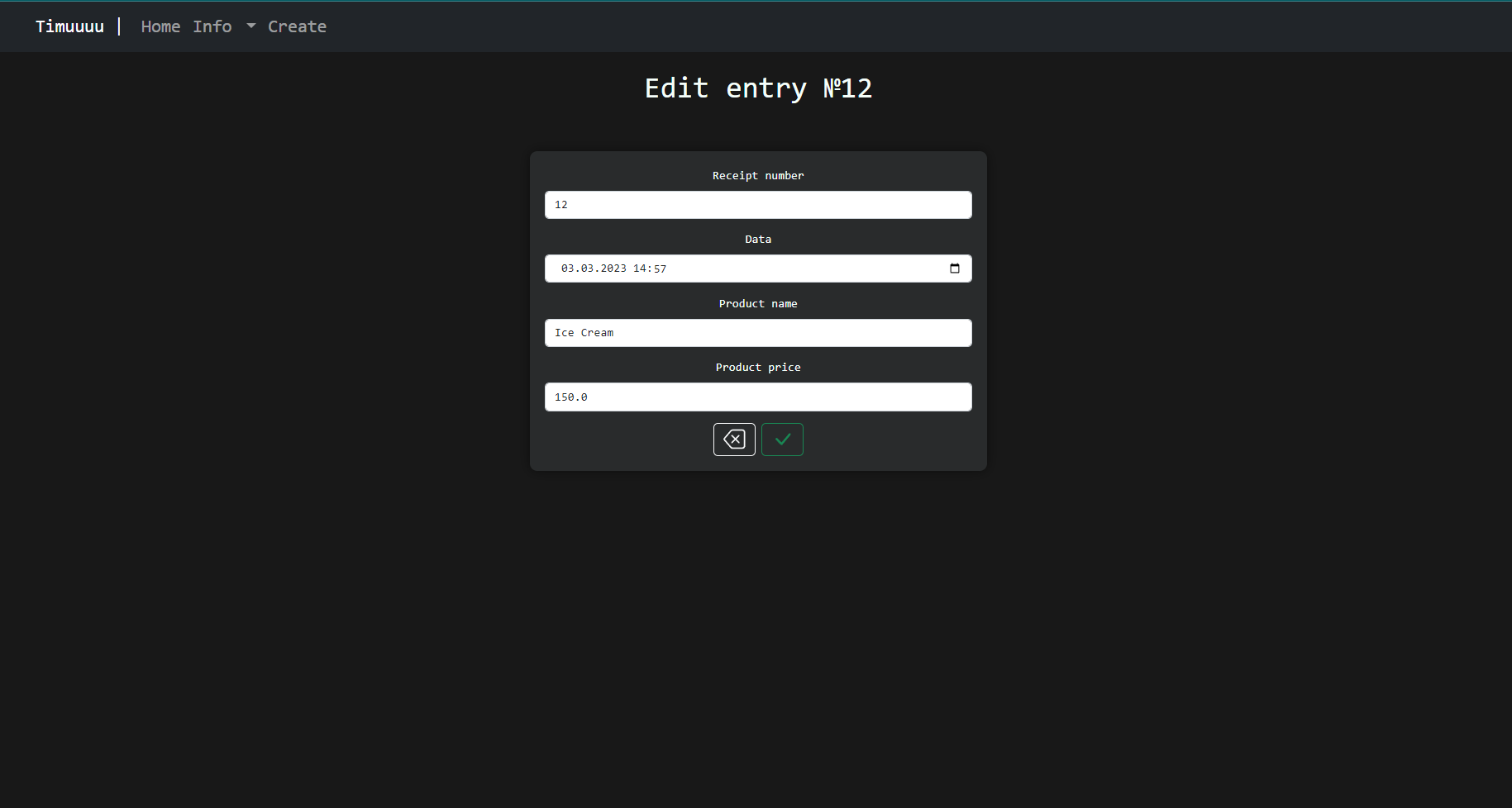
В ходе выполнения данной лабораторной работы затруднений не возникло

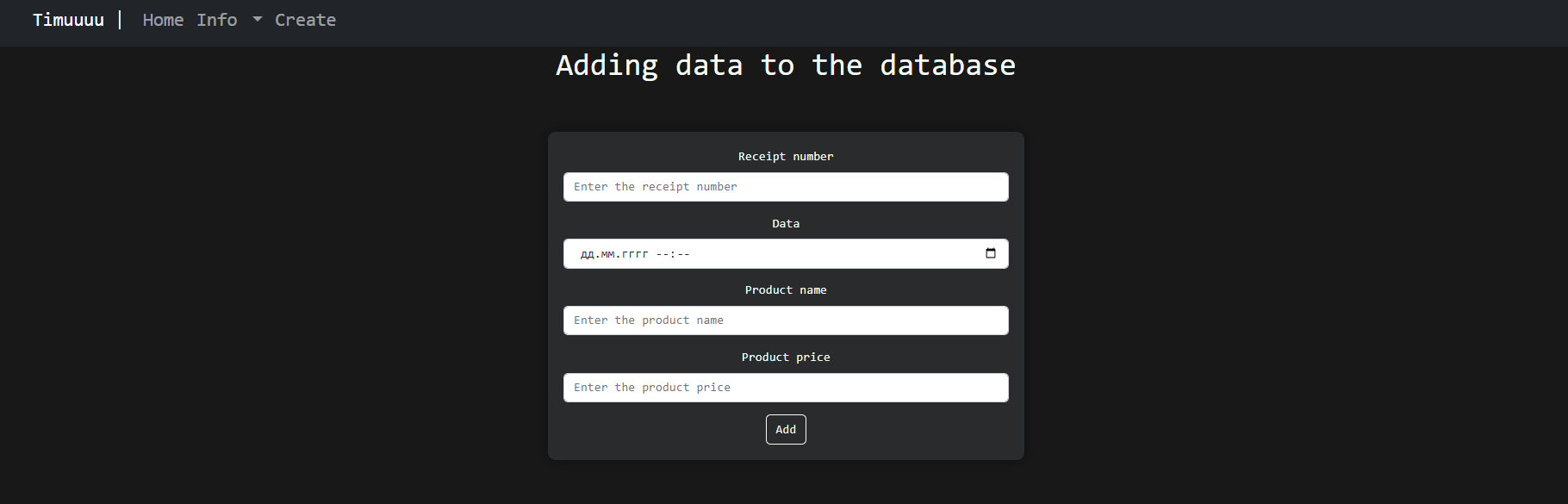
**Тестирование приложения**

****

**Рис. 1 «Главная страница»**

**  
Рис. 2 «Страница с чеками»**

**  
Рис. 3 «Редактирование чека»**

**  
Рис 4. «Создание чека»**

**Исходный код**

**https://github.com/TimurGayazov/Simple-Django-App**