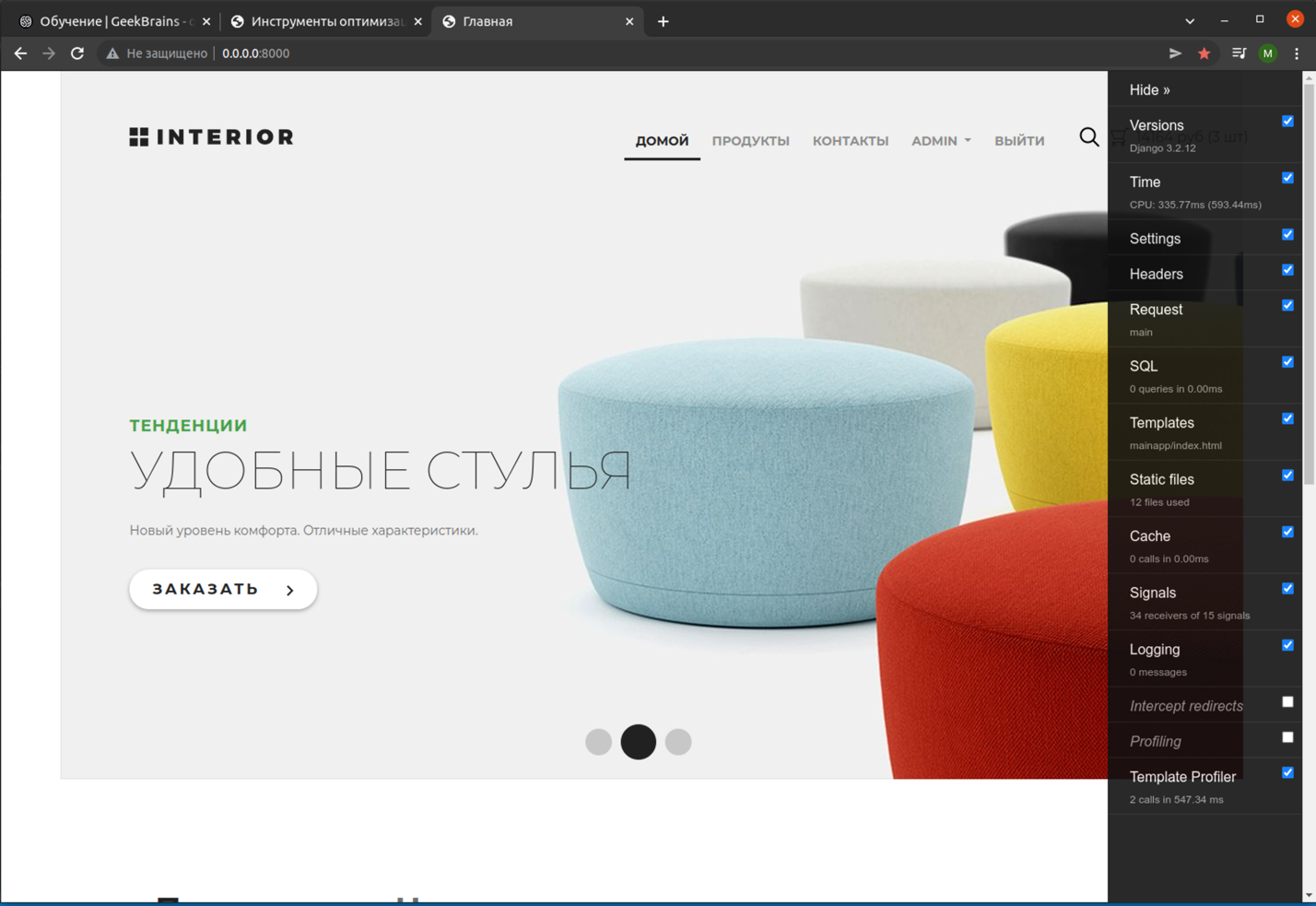
**Задание №1**

1. Установить приложение django-debug-toolbar. Оценить время загрузки страниц. Найти самые медленные контроллеры. Сделать скриншоты с учётом адресной строки браузера с замерами. *37:00*

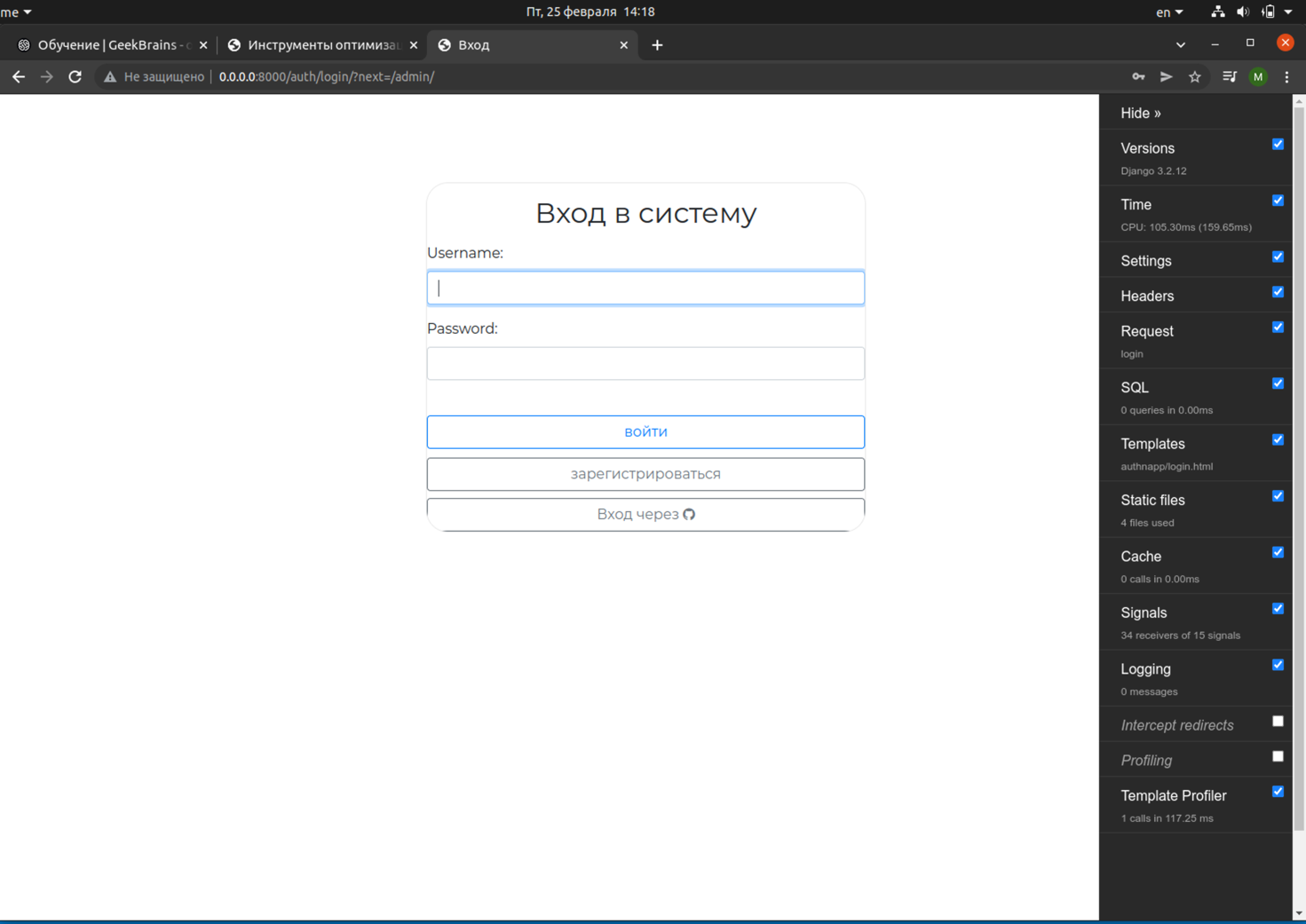
**Время загрузки mainapp/views.py (она же главная страница, как я понимаю) (CPU: 335.77ms):**



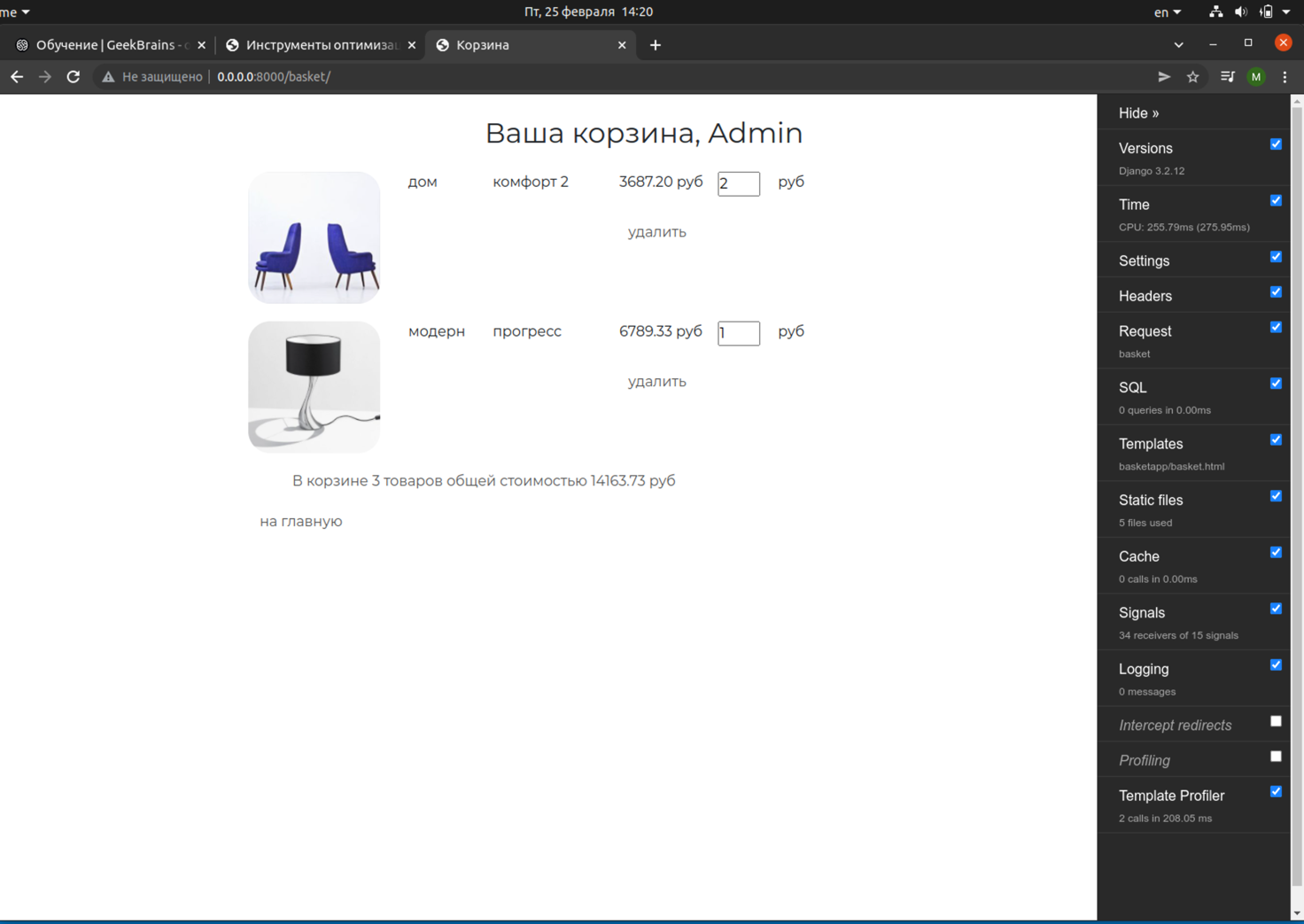
**Время загрузки adminapp/views.py (CPU: ? ms):**

Не могу почему-то зайти в админку. Раньше получалось – сейчас нет :(

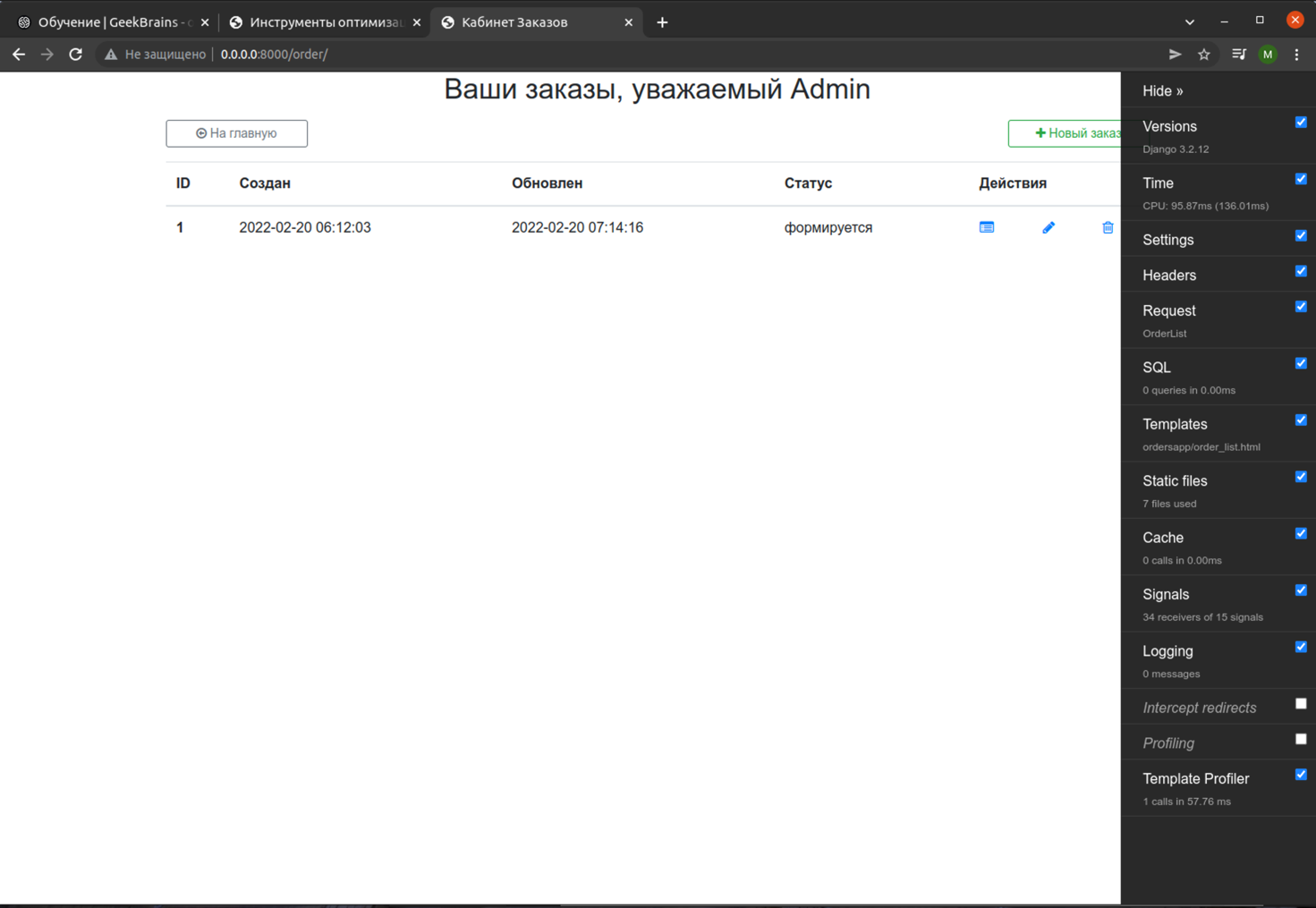
**Время загрузки authnapp/views.py (CPU: 105.30ms):**



**Время загрузки basketapp/views.py (CPU: 255.79ms):**



**Время загрузки ordersapp/views.py (CPU: 95.87ms):**



**ВЫВОД**: mainapp/views.py – самый медленный контроллер.

**Задание №2**

1. Установить утилиту siege и провести нагрузочное тестирование. *1:20* Зафиксировать результаты в скриншотах, а так же выявить какие контроллеры работали с ошибками. На скриншотах зафиксировать:
   * Адреса, с которыми проводилась осада
   * Результаты до оптимизации (в том числе с фиксацией ошибок перехода на недоступные адреса)
   * Результат после оптимизации

Файлы в отдельной папке.

# Почему-то authapp после оптимизации стал работать хуже – не понимаю в чём причина.

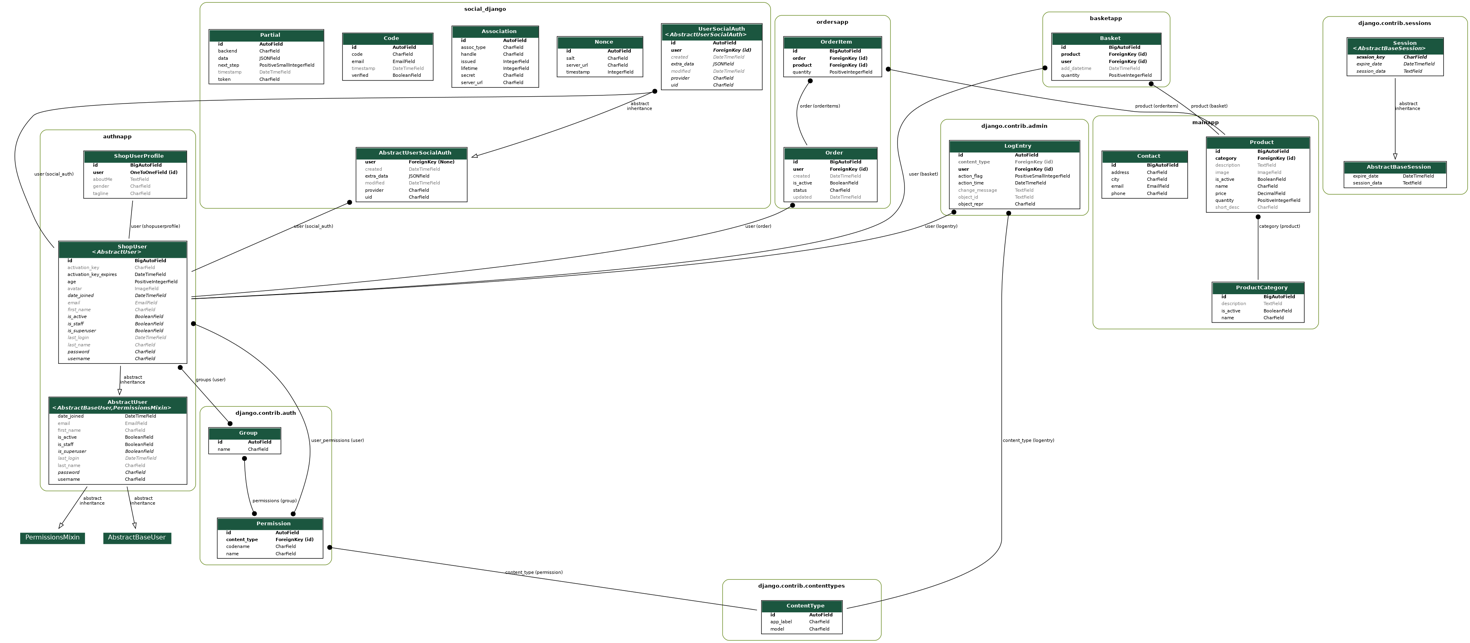
**Задание №3**

1. Провести оптимизацию работы с БД в проекте. Зафиксировать результат в скриншотах: до оптимизации, после оптимизации.

?

**Задание №4**

1. (\*) Визуализировать структуру моделей проекта при помощи django\_extensions. *53:30*



Все скриншоты размещайте в проекте в папке screenshots с обязательным указанием счётчика вначале. Например, на первом шагу Вы проверяете установку gunicorn - делаете скриншот и называете его 001\_gunicornCheck.png. Чтобы оценить результат используйте ключевые слова before и after. Например, до оптимизации базы данных скриншот после осады может называться 003\_before\_db\_optimize.png и после 004\_after\_db\_optimize.png.

Визуализированную структуру так же размещайте в папке screenshots