

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

Колледж экономики и информатики им. А.Н. Афанасьева

Отчёт по выполненному заданию на самостоятельную работу

по учебной практике.

Выполнил студент

Группы: ИСдо-41

Проконов Кирилл Сергеевич

Ульяновск

2023

**Анализ и обработка исключений для веб-приложения “РентАвто”.**

Для анализа необходимо выполнить действия.

**Обработчик ошибки о переходе к несуществующей странице.**

Первым делом узнаем, что будет при выполнении перехода на несуществующую страницу. Вписываем в URL следующие значения: http://127.0.0.1:8000/Owner/Delete

Веб-браузер выдаёт запись из Debug лога фреймворка Django. Для дальнейшей работы дебаггер необходимо выключить.

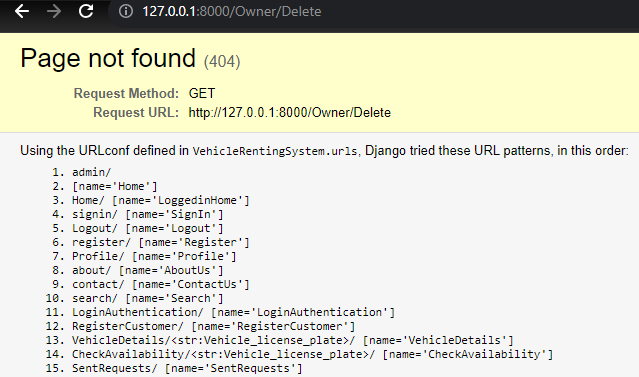


Рисунок 1 – Debug лог информирующий о несуществующей страницы

Для отключения Debug лога нужно зайти в файл settings.py, и вписать хостинг в параметр разрешённые хосты.

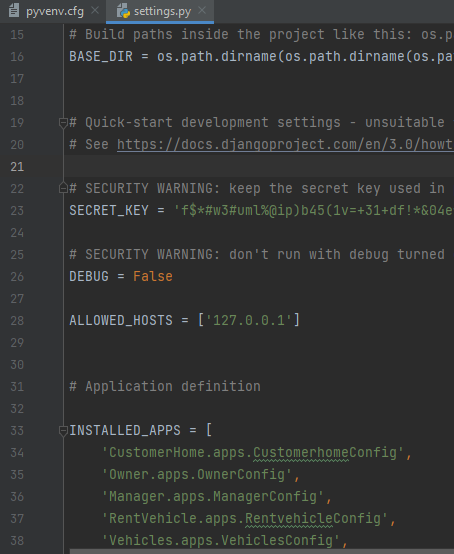


Рисунок 1 – Настройка файла Settings.py, отключения дебаггера

Заново перейдём на недействующий URL: http://127.0.0.1:8000/Owner/Delete

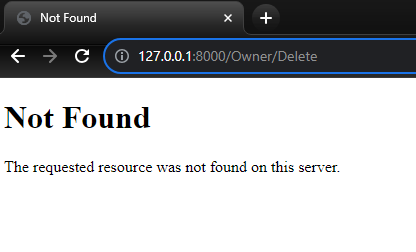


Рисунок 1 – Сообщение веб-браузера о несуществующей странице на сервере

Как можем заметить веб-браузер выдаёт ошибку “Not Found” означает, что браузер не может найти страницу по заданному адресу. Создадим обработчик исключения для данной ситуации.

Напишим обработчик с помощью вызова функции, которая при вызове будет выводить рендер HTML страницы. Название у функции соответствующее, «page\_not\_found\_view». Прописывать небоходимо в файл views.py, так как он отвечает за представления-функции веб-приложения.

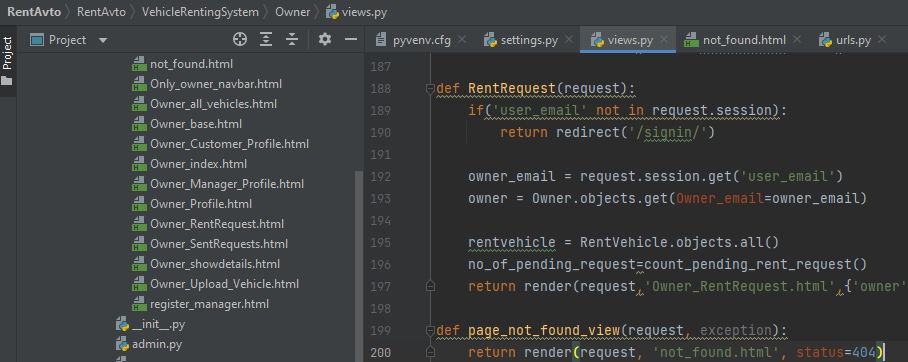


Рисунок 1 – Обработчик на основе вызова функции

В файле urls.py определяем значение переменной handler404:

handler404 = "Owner.views.page\_not\_found\_view"

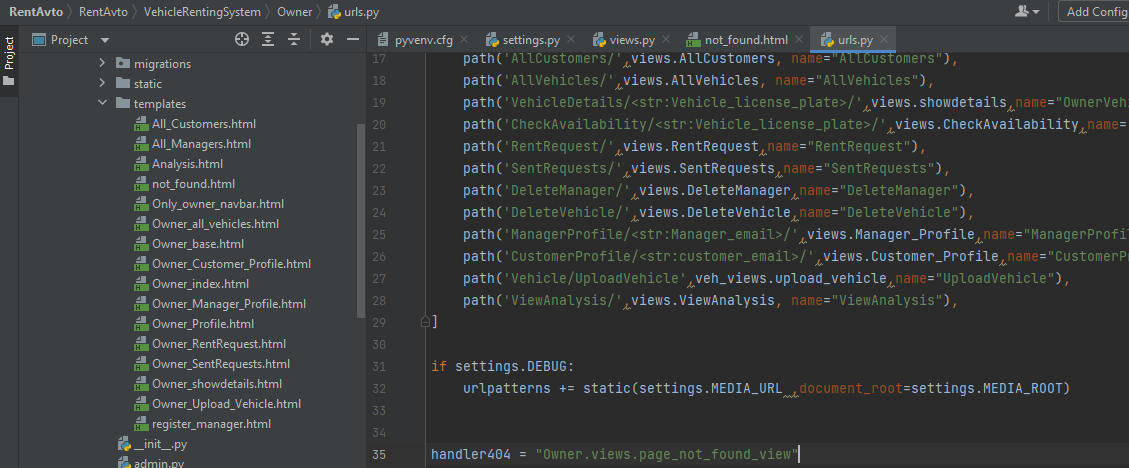


Рисунок 1 – Определение функции в файле urls.py

После проделанных действий нужно создать шаблон HTML, в котором будут находиться теги для отображения сообщающего текста об ошибке.

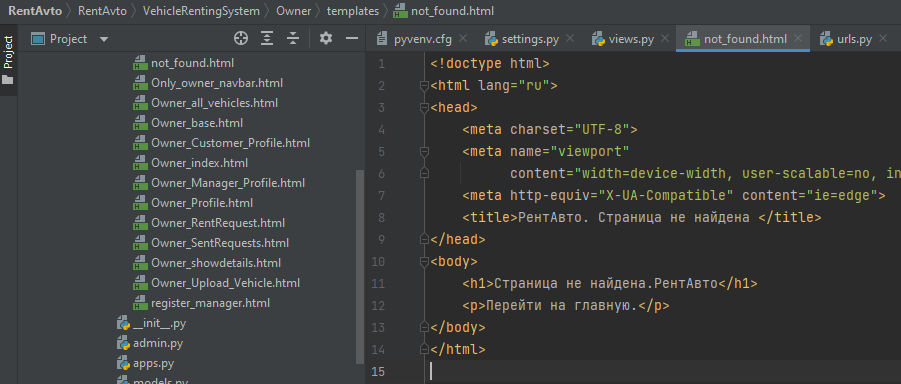


Рисунок 1 – Страница, которая сообщает пользователю об ошибке

**Обработчик ошибки сервера.**

Ошибка 500 — это внутренняя проблема сервера. Она возникает, когда браузер или другой клиент отправляет серверу запрос, а тот не может его обработать. Например, в веб-приложении появился не оптимизированый скрипт, который заполняет оперативную память до её максимума.

Моё приложение правильно настроено и таких ошибок не наблюдалось, но в любом случае необходимо сделать данный обработчик с целью дальнейшей разработки веб-приложения.

Проделаем действия для обработки данного статуса ошибки.

Прописываем функцию в файл views.py:

def handler500(request):

return render(request, ’500.html’, status=500)

В файле urls.py определяем значение переменной handler500:

Handler500 = "Owner.views.page\_not\_found\_view"

from django.conf.urls import handler505

handler500 = “your\_app.views.handler\_other\_errors”

После проделанных действий нужно создать шаблон HTML, в котором будут находиться теги для отображения сообщающего текста об ошибке.



Рисунок 1 – Страница, которая сообщает пользователю об ошибке