Тестовое задание Python/Django:

Разработать приложение которое будет состоять из двух небольших страниц:

1. Страница пользователя
2. Страница со списком пользователей

Описание функционала и этапов работы

**Back-end (Python):**

* данные представляют собой список пользователей (файл users.json) и статистику пользования сайтом (файл users\_statistic.json)
* нужно создать модели User и Statistic, промигрировать и затем залить в них данные посредством management команды
* разработать функционал выборки списка пользователей и информации о них (модель User и Statistic (для выгрузки информации total\_clicks и total\_page\_views)) с постраничной навигацией (количество пользователей и текущая страница приходит в запросе с Front-end)
* разработать функционал выборки данных статистики (модель Statistic) по id пользователя с возможностью фильтрации по дате (“от” и “до” приходит в запросе)
* перед выборкой данных, должна выполняться проверка на существование пользователя
* АПИ страницы со списком пользователей сделать CORS доступным для любого домена
* АПИ страницы пользователя должно быть доступно только с текущего сайта

**Front-end:**

Страница пользователя:

* на странице пользователя должны отображаться две диаграммы со статистикой пользователя по полям page\_views и clicks
* статистика пользователя должна отображаться в виде линейной диаграммы.
* если в данных пользователя отсутствует значение за какую-то дату, это значит page\_views и clicks отсутствовали в этот день и они равны 0
* по умолчанию выводится статистика за неделю, но должна быть предусмотрена возможность выбора промежутка “от” и “до”.

Страница списка пользователей:

* создать страницу списка пользователей, на которой будет размещена таблица со списком пользователей и информацией о них, полученная с API (поля которые должны быть в таблице: id, first\_name, last\_name, email, gender, ip\_address, total\_clicks, total\_page\_views)
* на странице должна быть реализована постраничная навигация
* максимальное количество пользователей на странице 50
* при нажатии на строку в таблице, происходит переход на страницу пользователя (пример роута /users/:id)

## Требования к выполнению

Back-end: Python 3, Django 1.11, MySQL и предпочтительно DRF.  
Front-end: должно быть общение с беком через АПИ.

Проект можно задеплоить на Heroku (либо другой платформе).  
После выполнения предоставь линк на гитхаб и сайт.  
  
Подсказка: используй преимущества Django ORM; будь внимателен к деталям; предположим, это хайлоад проект.

Ссылки с тестовыми наборами:

Пользователи:

<https://drive.google.com/file/d/1QUnoXgVO-4Ku6XNMQXZguUsatOyHq78-/>

Данные по статистике:

<https://drive.google.com/file/d/1EzpuZkazaxL3DWVCDbQHQE1CCkP18T3b/>

Выполнение:

Link: <https://afternoon-tor-88988.herokuapp.com/users/>

admin: <https://afternoon-tor-88988.herokuapp.com/admin/>

login: admin

password: admin

Навигация api:

api users: <https://afternoon-tor-88988.herokuapp.com/users/api/users/> (так как пользователей 1000 вывожу первых 10)

api user: <https://afternoon-tor-88988.herokuapp.com/users/api/users/1>  (id пользователя)

api statistic: <https://afternoon-tor-88988.herokuapp.com/users/api/statistic-users/>  (так как статистик 19k вывожу первые 10)

api statistic user: <https://afternoon-tor-88988.herokuapp.com/users/api/statistic-users/1>  (id пользователя)

Результат:

* зависимости/пакеты не соответствуют тестовому заданию
* в тестовом четко написано использовать MySQL, он использует PostgreSQL
* реализован функционал который не требуется
* CORS доступ не реализован
* SQL запросы не оптимальны
* management команда не реализована
* тестов нету

Итог: результат на 85% не соответствует тестовому.