

Итоговый отчет по проекту «Сервис коллективных блогов».

Горшков Илья, группа 143.

Начну описание работы с момента, когда были готовы шаблоны всех страниц и навигация между ними (описание работы над этим этапом содержится в промежуточном отчете). После этого большая часть работы была проделана над функционалом.

Первым делом, для того, чтобы пользователи могли идентифицировать друг друга, нужно было добавить возможность регистрации и авторизации. Этот функционал я реализовал без особых проблем, создав необходимые для этого формы и настроив их валидацию таким образом, чтобы при регистрации осуществлялась проверка совпадения паролей и т.д.

Далее следовало дать пользователям возможность делать публикации самостоятельно (пока это можно было делать исключительно из панели администратора). Для этого была создана отдельная страница. Так как каждый пост принадлежит некоторому блогу, я сделал ссылку на добавление нового поста только на страницах блогов, а на главной странице эта опция отсутствует. Разумеется, не авторизовавшись на сайте, пользователь не может делать публикации.

В этот момент я понял, что посты, сделанные в стандартной форме, состоящие из обычного однородного текста, выглядят некрасиво и непрактично, поэтому встал вопрос о расширении поля создания поста. С этим заданием прекрасно справляются WYSIWYG - редакторы. Добавить их в свой проект позволяет модуль `django_wysiwyg`, поддерживающий работу с большинством WYSIWYG — редакторов. Несмотря на то, что наиболее распространенным и функциональным является SCEditor, мой выбор пал на TinyMCE. Я посчитал, что многие функции, представленные в SCEditor, излишни, а выбранный мной редактор содержал только самые необходимые функции — добавление изображений, выравнивание текста, применение к тексту стилей курсив/жирный/перечеркнутый. Прочитав описание модуля `django_wysiwyg`, я без особых проблем подключил к своему проекту улучшенный редактор текста.

Теперь гости моего сервиса могли делиться друг с другом публикациями, однако им сложно было отличить хорошую публикацию от плохой. В этом им должны были помочь рейтинги постов. Внизу страницы поста были добавлены кнопки повышения/понижения рейтинга поста. Если пользователь не авторизовался, то нажав на кнопку, он увидит предупреждение «Пожалуйста, авторизуйтесь». Если же пользователь, например, уже повышал рейтинг некоторой публикации, то он получит сообщение «Вы уже лайкали эту запись». Функционал изменения рейтинга выполнен при помощи технологии AJAX и не требует перезагрузки страницы. Также следует отметить, что числа в скобках рядом с именами блогов в панели «Лучшие блоги» являются не количеством публикаций в этом блоге, а рейтингом этого блога, который является рейтингом суммы всех постов, принадлежащих этому блогу.

Чтобы пользователи могли делиться впечатлениями от постов, были добавлены комментарии и кнопка, позволяющая поделиться ссылкой на новость в сети ВКонтакте.

Несмотря на то, что для администратора изменение и удаление постов доступно в стандартной панели администратора Django, на сайт были добавлены кнопки, позволяющие удалять нежелательные посты и комментарии. Это сделано для того, чтобы администратор не

тратил время на вход в панель, когда необходимо как можно быстрее удалить вредоносную публикацию.

Также для борьбы с недоброжелателями (на самом деле просто для ознакомления) была добавлена капча при регистрации и авторизации. Использовалась капча из модуля Django captcha. Её интеграция требует добавления пары строк и тоже не вызвала существенных затруднений.

На данный момент вся защита от доступа людей, не имеющих на это полномочий, к особым функциям заключалась в скрытии от них ссылки. То есть, хотя гость и не видит кнопки «Удалить пост», предназначенной для администратора, он может ввести соответствующий url, определенный в urls.py, и удалить пост. В том числе для борьбы с этим была создана кастомная страница 404.html и определено перенаправление на нее в случаях, когда пользователь пытается получить доступ к запрещенным ему функциям по url. В том числе, теперь неавторизованный пользователь не может перейти на страницу добавления поста, хотя раньше у него была такая возможность. Помимо описанных функций, кастомная страница 404.html открывается в стандартных функциях, когда в urls.py не нашлось ни одного обработчика запроса пользователя.

В конце, когда основной функционал был наконец реализован, была проделана работа по незначительному улучшению сайта — добавление уникальных title'ов на каждую страницу и кнопки поделиться в twitter.