**Редактирование и удаление записей через шаблоны**

В данной лабораторной работе мы рассмотрим создание шаблонов и вообще способ удалять или изменять записи из базы данных на локальном сервере.

Для начала нам надо создать URL-адрес для функции представления шаблона, который будет необходим для редактирования записей. Для этого переходим в файл *urls.py* приложения *news*.

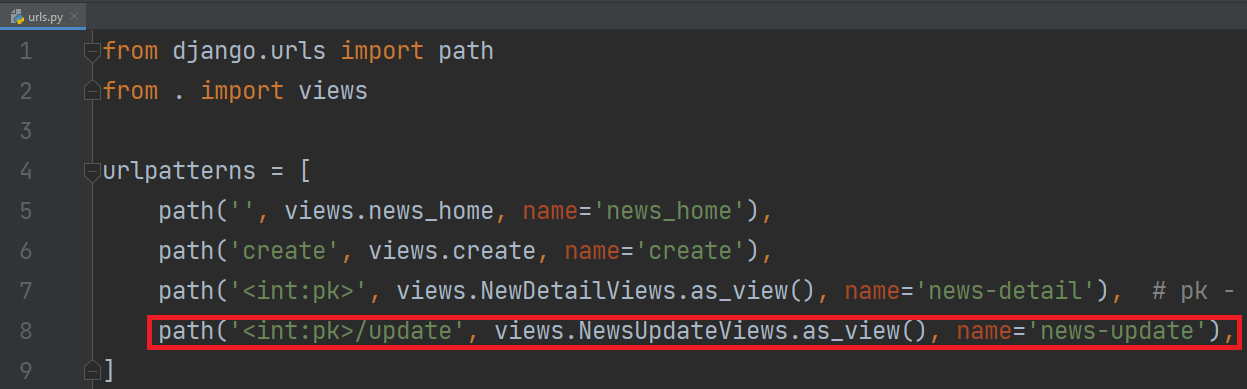


Рисунок 1 - URL-адрес страницы редактирования

Следующим шагом станет само создание функции представления шаблона в файле *views.py*. Для начала подключаем класс *UpdateView*, который позволяет редактировать записи в полях базы данных.

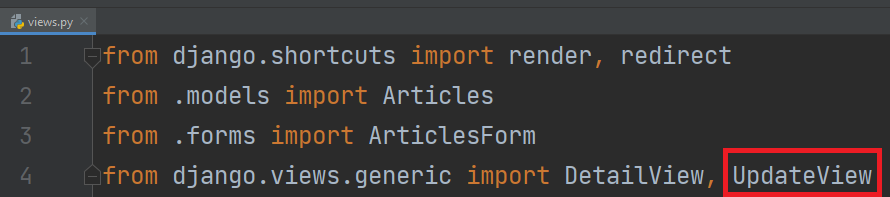


Рисунок 2 - Импортирование класса UpdateView

Опишем модель класса, путь до HTML-шаблона, а также поля для отображения.

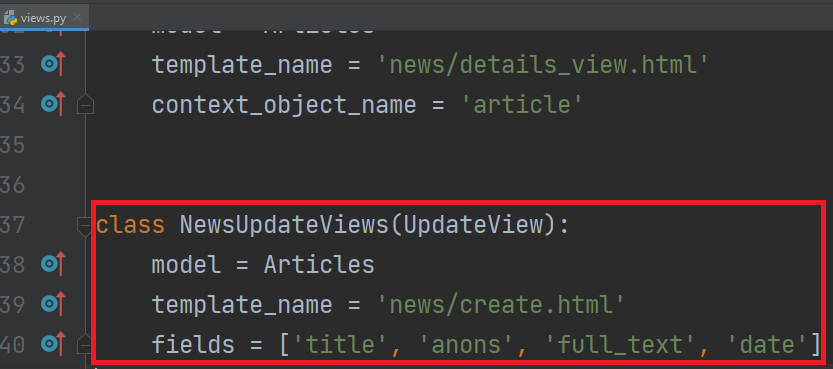


Рисунок 3 - Создание класса NewsUpdateViews

Запустим локальный сервер и перейдём на страницу редактирования.

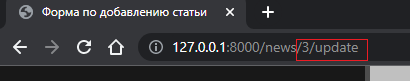


Рисунок 4 - Адрес страницы редактирования в браузере

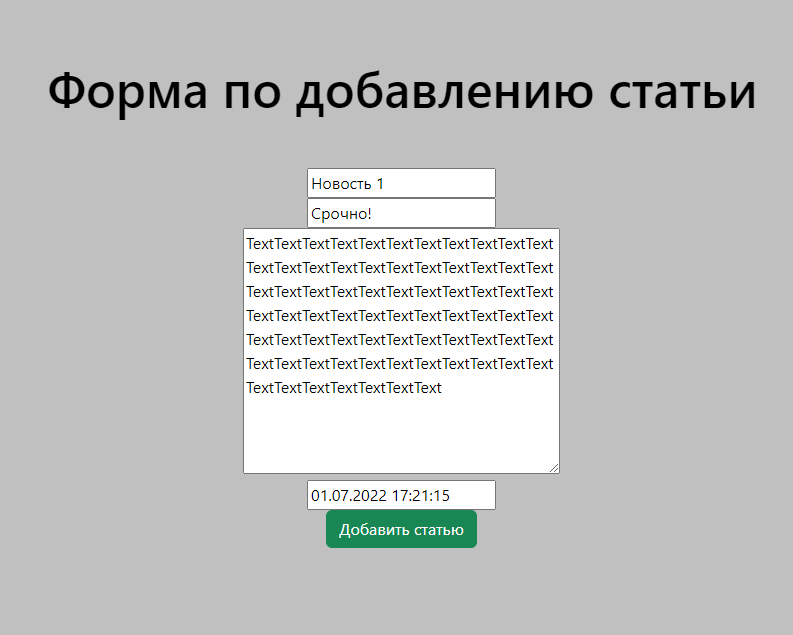


Рисунок 5 - Страница редактирования статьи

Отображаемые формы редактирования выглядят не очень подобающим образом. Исправим это, заменив в классе *NewsUpdateViews* поля отображения на класс формы.

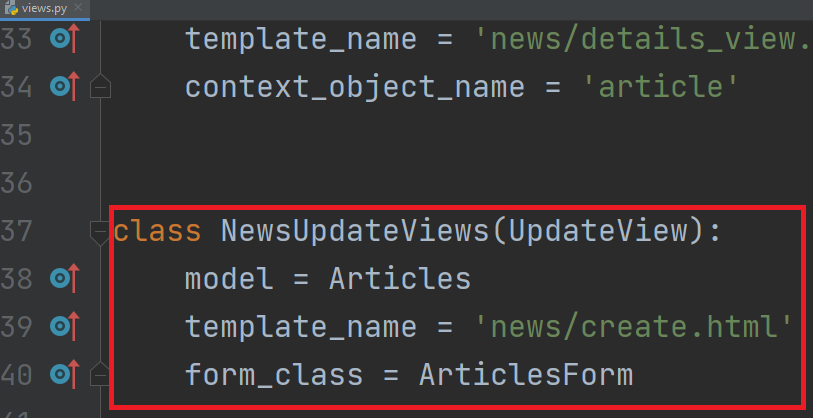


Рисунок 6 - Форма для класса NewsUpdateViews

Обновим страницу для подтверждения результата. Также попытаемся изменить любой форме данные. При нажатии на кнопку “Добавить статью” у нас произойдёт ошибка, связанная с неправильной переадресацией.

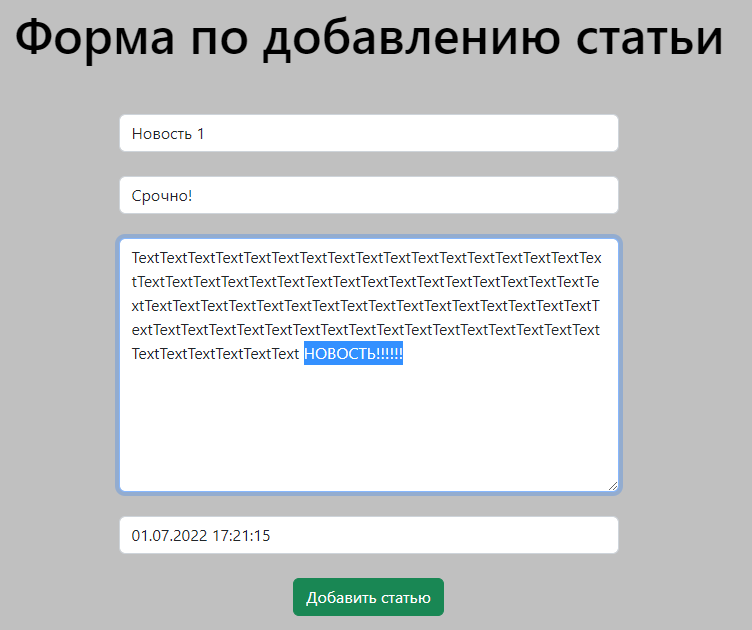


Рисунок 7 - Страница редактирования статьи

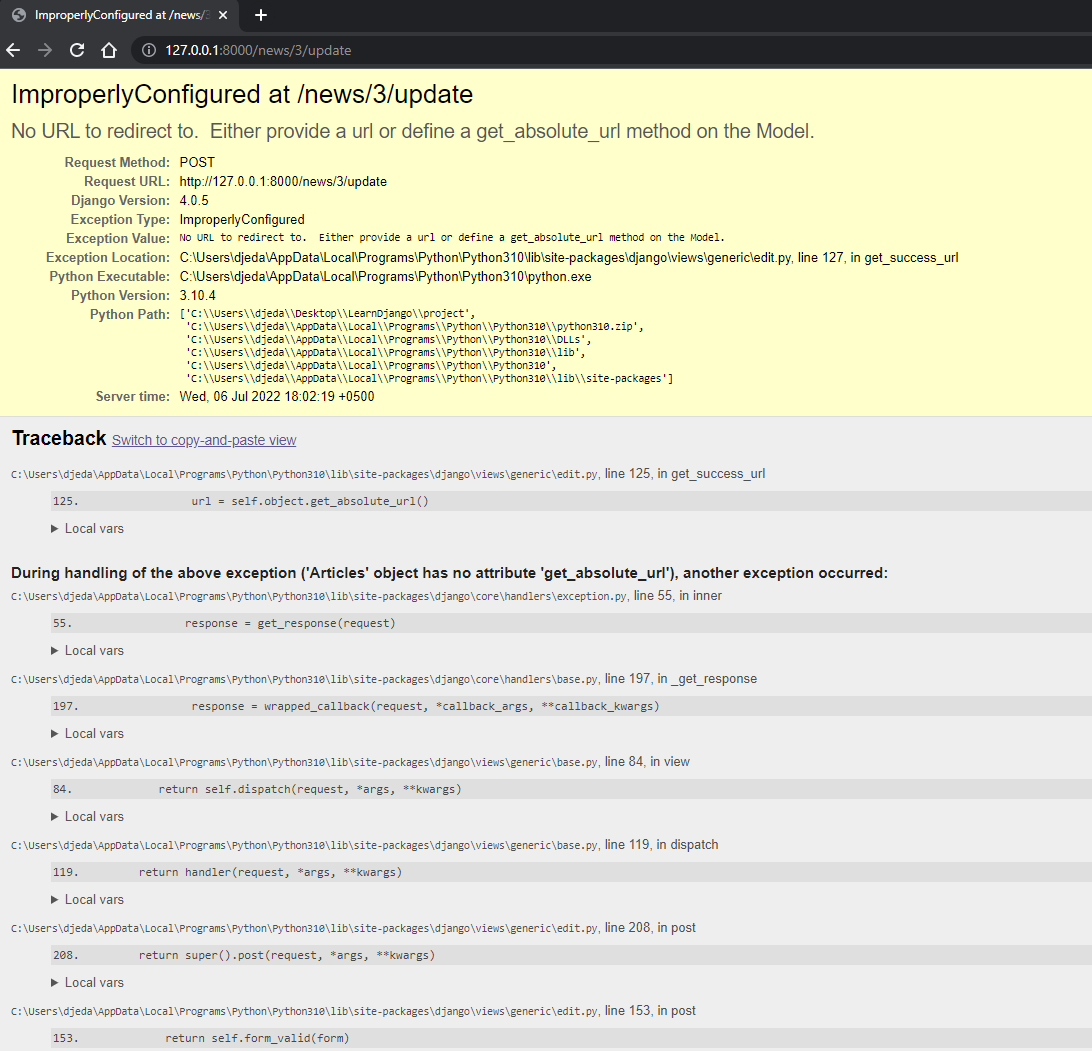


Рисунок 8 - Ошибка переадресации

Браузер выдаст нам ошибку, т. к. мы не указали, на какую страницу будем переходить, после внесения каких-либо изменений на странице редактирования. В файле *models.py* добавим метод *get\_absolute\_url* и укажем в нём адрес для страницы, на которую должен произойти процесс перехода.

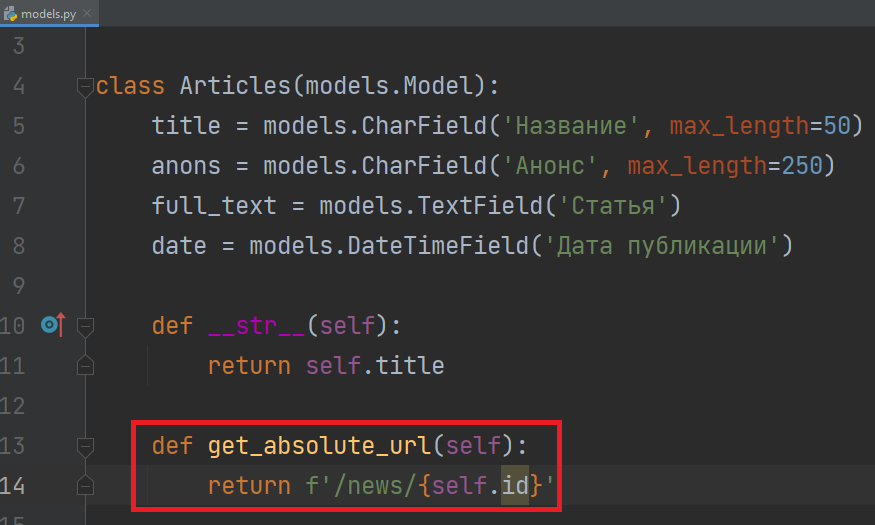


Рисунок 9 - Метод перенаправления URL-адреса после подтверждения изменений в данных базы данных

Теперь мы можем редактировать статьи на прямо на странице, а после нажатия на кнопку нас будет переносить на страницу новостной ленты.

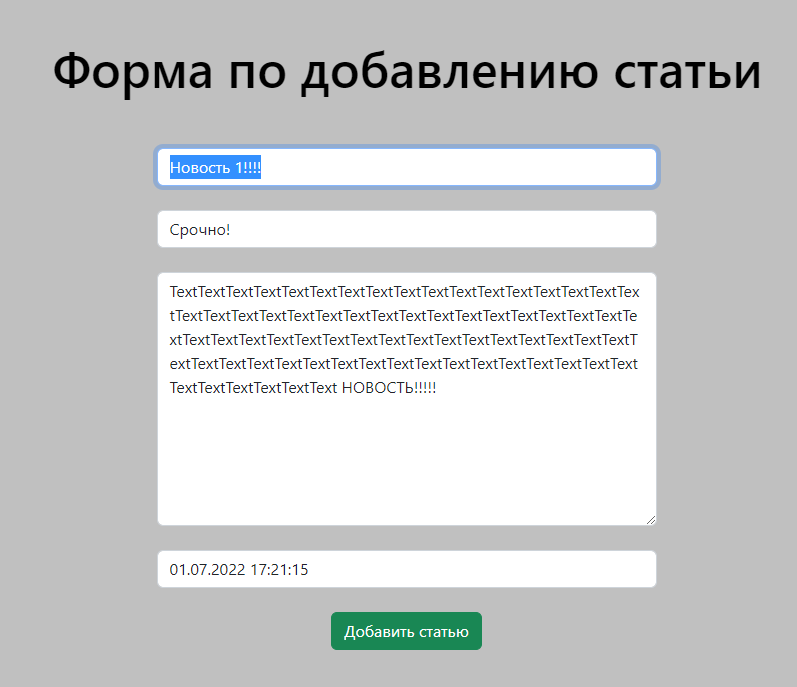


Рисунок 10 - Редактирование статьи

Теперь создадим страницу удаления статей.

Сначала укажем в файле *urls.py* URL-адрес и функцию представления для шаблона удаления записей из базы данных.



Рисунок 11 - URL-адрес страницы удаления статьи

Далее в файле *views.py* импортируем класс *DeleteView*, который позволяет удалять записи из базы данных.

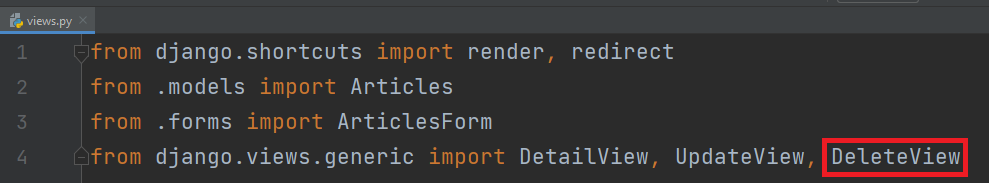


Рисунок 12 - Импортирование класса DeleteView

Создадим класс *NewsDeleteViews*. В нём нам понадобится описать только модель и путь до нового HTML-шаблона.

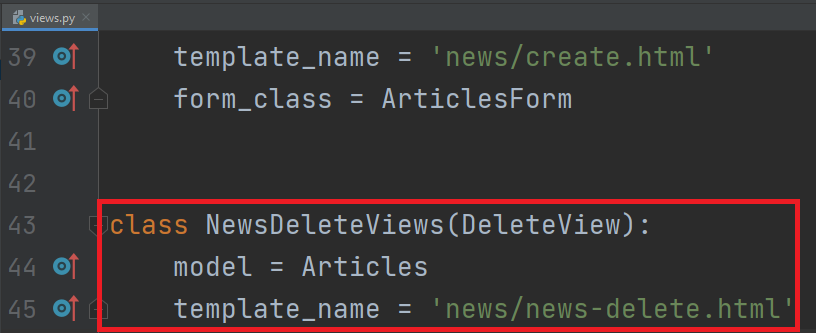


Рисунок 13 - Класс NewsDeleteViews

Создадим указанный ранее HTML-шаблон. В нём нам понадобится обязательно *CSRF-токен* и кнопка удаления статьи.

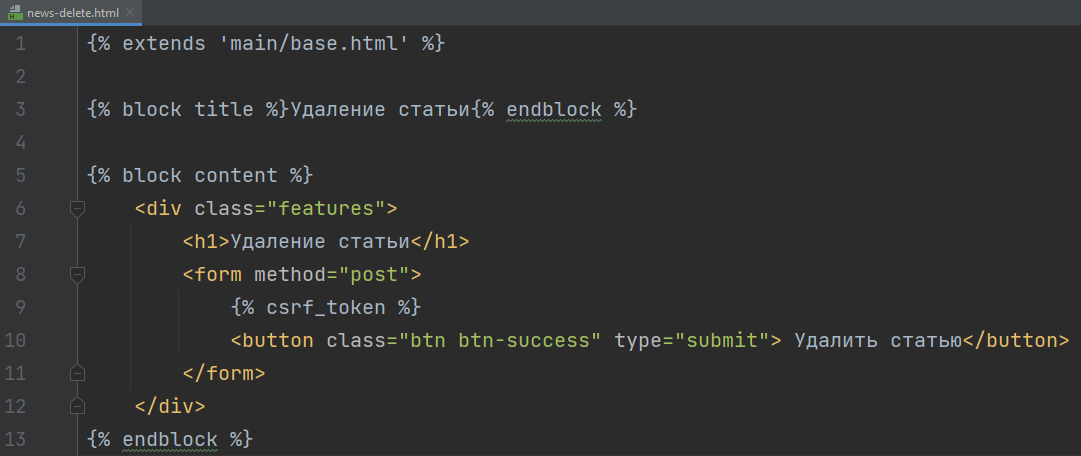


Рисунок 14 - Параметры страницы удаления статьи

Перейдём на страницу удаления статьи.

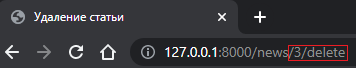


Рисунок 15 - Адрес страницы удаления статьи в браузере

На странице удаления статьи у нас виден только заголовок страницы и кнопка, предлагающая статью удалить.

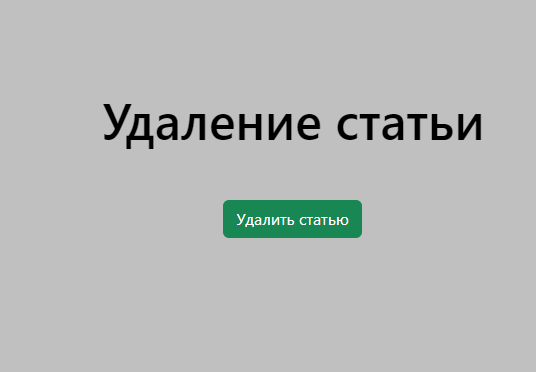


Рисунок 16 - Страница удаления статьи

При нажатии на кнопку “Удалить статью” у нас происходит немного другая ошибка, но также связанная с URL-маршрутизацией. Отсутствует перенаправление на другую страницу.



Рисунок 17 - Ошибка переадресации

Для того, чтобы исправить данную ошибку, необходимо в файле views.py в классе *NewsDeleteViews* добавить строку с URL-адресом, на который будет перенаправлен пользователь при нажатии на кнопку “Удалить статью”.

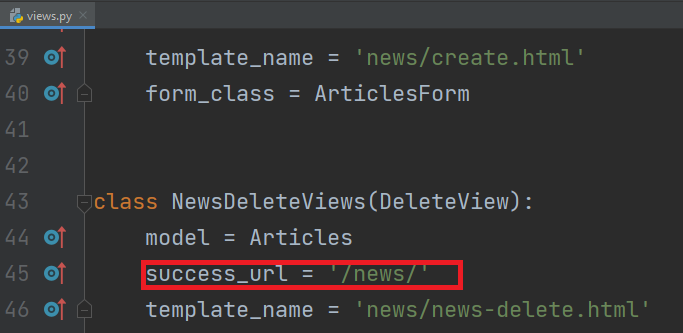


Рисунок 18 - Путь переадресации

Обновив страницу, можно увидеть, что произошел переход на страницу со списком новостей и той новости, которую мы удалили, уже нет.

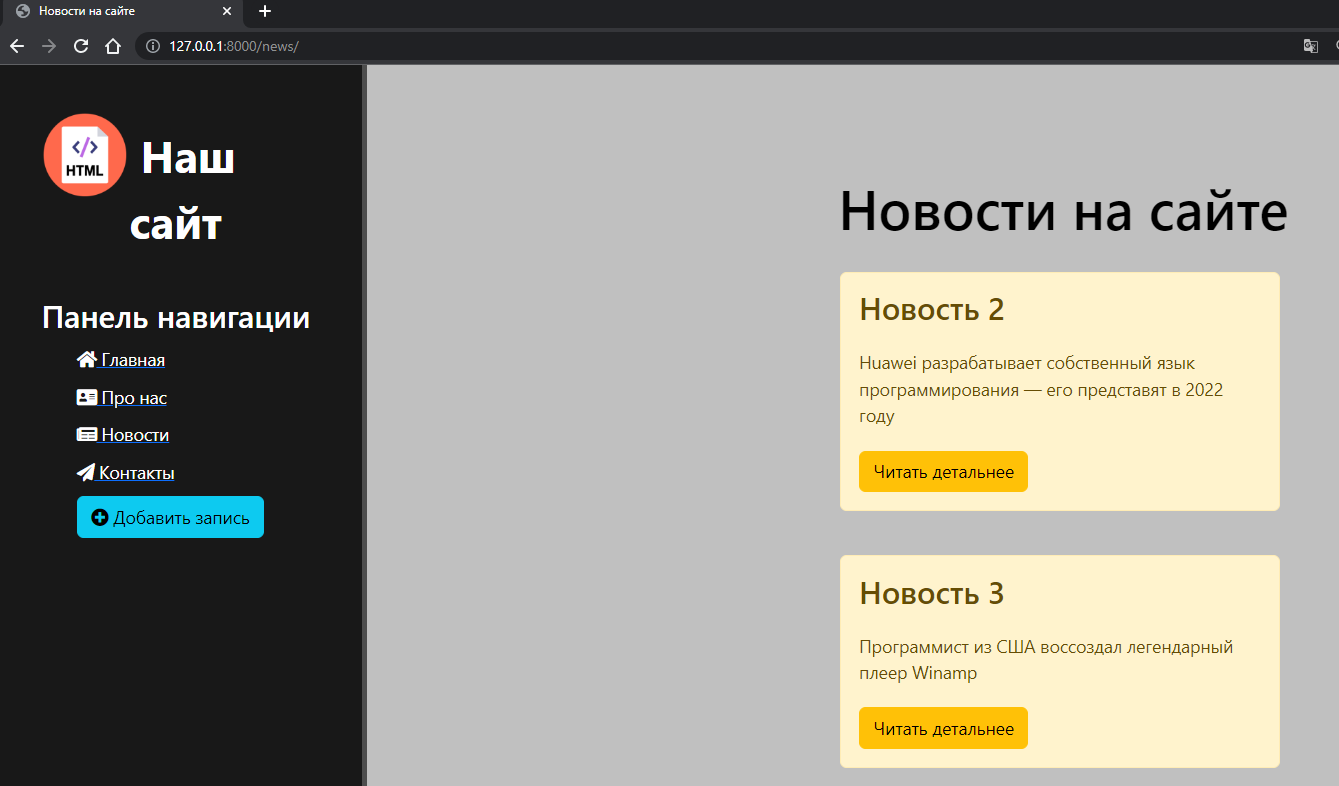


Рисунок 19 - Новостная лента после удаления статьи

Чтобы нам постоянно не вводить в адресной строке браузера адреса созданных нами страниц, добавим на страницу просмотра статей две кнопки. В файле *details\_view*.html добавим теги *<a href>* для создания двух кнопок. Внешний вид кнопок возьмём из подключённой библиотеки Bootstrap. В них укажем ключи для адресов страниц редактирования и удаления, а также идентификаторы удаляемых/редактируемых записей.

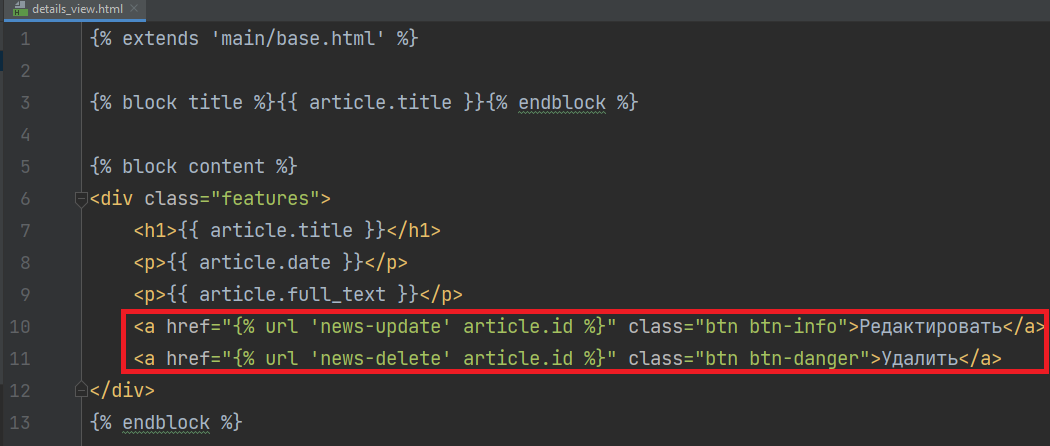


Рисунок 20 - Кнопки “Редактировать” и “Удалить” на странице просмотра статьи

Проверим результат, обновив страницу.

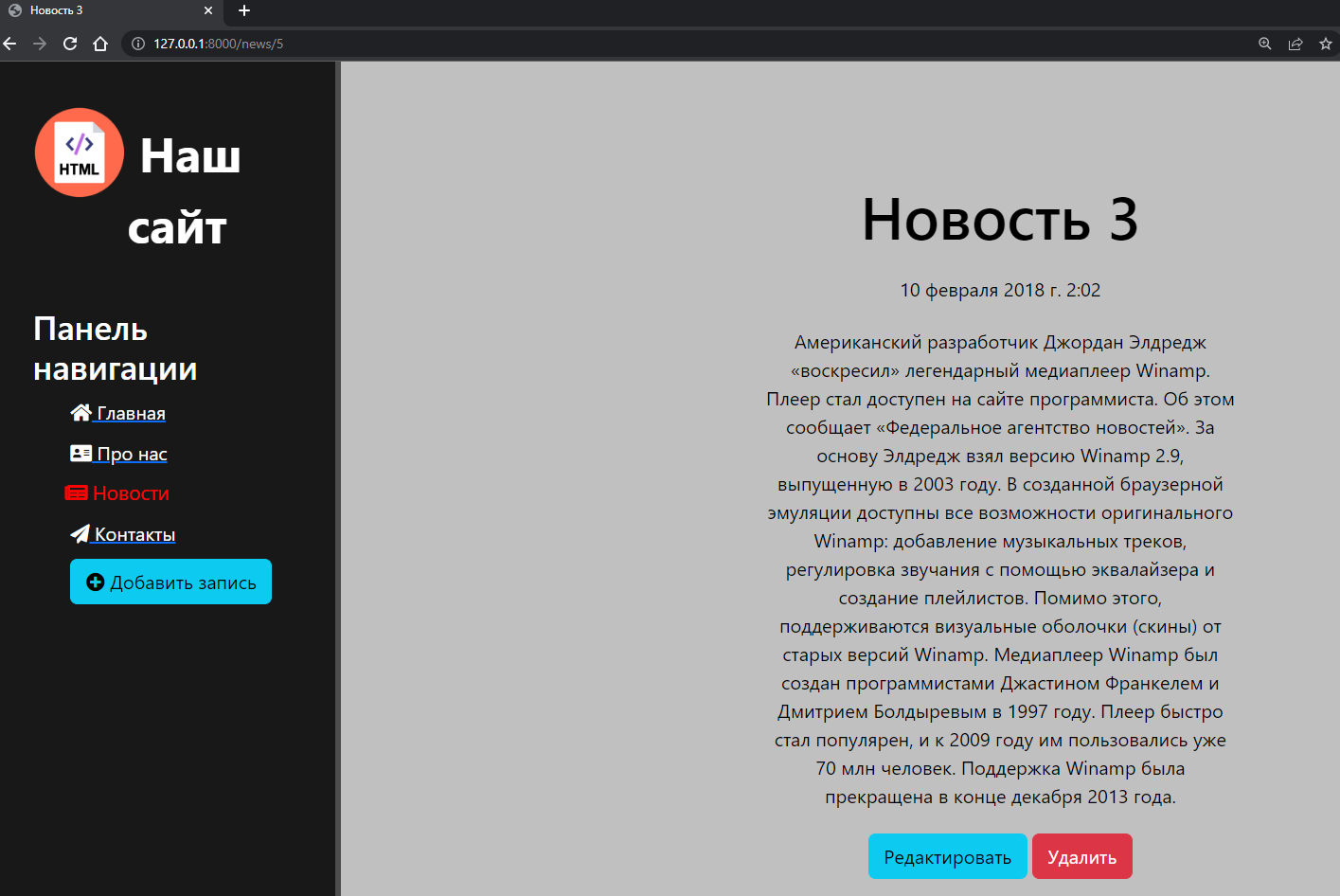


Рисунок 21 - Страница просмотра статьи