

**Государственное бюджетное общеобразовательное учреждение города
Москвы «Школа № 2033»**

САЙТ ДЛЯ ОБЩЕНИЯ И ПУБЛИКАЦИИ СТАТЕЙ

Участники:

Ученик 10 «Т» класса ГБОУ «Школа №2033»

Кравчук Тимофей Сергеевич

Ученик 10 «Т» класса ГБОУ «Школа №2033»

Алиев Руслан Андреевич

Научный руководитель:

Гришина Арина Александровна

Москва, 2024

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ЦЕЛЬ И ЗАДАЧИ ПРОЕКТА.....	4
МЕТОДИКА ВЫПОЛНЕНИЯ РАБОТЫ.....	5
ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.....	5
САЙТ.....	6
ТЕСТИРОВАНИЕ.....	11
ВЫВОД.....	12
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	13

ВВЕДЕНИЕ

В настоящее время создание и поддержка веб-сайтов становится важной задачей для многих разработчиков. Одной из популярных технологий для разработки веб-приложений является фреймворк Django, который предоставляет множество инструментов для быстрого создания надежных и масштабируемых веб-сервисов. Данный проект представляет собой веб-сайт, основанный на Django, с функциональностью публикации постов, написания комментариев к ним и регистрации пользователей.

Целью разработки проекта было создание простого, но функционального веб-приложения, которое позволяет пользователям взаимодействовать с контентом и друг с другом, оставляя комментарии под постами. Также предусмотрена система регистрации и авторизации пользователей для ограничения доступа к определенным функциям.

ЦЕЛИ И ЗАДАЧИ ПРОЕКТА

Цель проекта: создать сайт, на котором пользователи смогут публиковать свои статьи, просматривать чужие и их комментировать.

Задачи проекта:

1. Изучить теорию, необходимую для WEB разработки и создания сайта: изучить технологии, необходимые для создания сайта и WEB приложений в т.ч. язык программирования Python, библиотеку Django, язык гипертекстовой разметки HTML.
2. Продумать функционал сайта.
3. Продумать и реализовать BACKEND часть сайта.
4. Продумать и реализовать FRONTEND часть сайта.
5. Провести тесты сайта, выявить ошибки и исправить их.

МЕТОДИКА ВЫПОЛНЕНИЯ РАБОТЫ

Теоретическая часть

Django — это высокоуровневый Python-фреймворк для веб-разработки, который позволяет быстро разрабатывать сложные веб-приложения. В основе Django лежат принципы повторного использования кода, принцип DRY (Don't Repeat Yourself) и "административный интерфейс", который позволяет эффективно управлять данными через браузер.

Основные компоненты Django:

1. Модели (Models) — представляют данные, которые хранятся в базе данных.
2. Представления (Views) — содержат логику отображения данных пользователю.
3. Шаблоны (Templates) — отвечают за отображение HTML-страниц.
4. URLs — маршруты, связывающие запросы с соответствующими представлениями.

Сайт

Дизайн сайта выполнен в сочетании белого, фиолетового и тёмно-серого цветов. Страницы написаны на языке гипертекстовой разметки HTML с использованием CSS. Всего сайт состоит из восьми обычных страниц и одного общего шаблона.

Общий шаблон содержит в себе панель навигации в верхней части сайта и CSS-код для создания общей стилистики сайта.

Первая страница является главной, для не авторизированных пользователей здесь содержится приглашение зарегистрироваться или войти и соответствующие кнопки. При попытке авторизованного пользователя зайти на первую страницу будет происходить переадресация на вторую страницу.

На второй странице находится список со всеми статьями расположенными по дате создания, чем позже опубликована статья, тем выше она находится.

На третьей странице находятся результаты поиска по названиям статей.

На четвёртой странице содержится выбранная конкретная статья, комментарии к ней. Также если на страницу зашёл автор статьи ему дополнительно отображается кнопки удаления статьи и её изменения.

На пятой странице содержится уведомление о необратимости удаления статьи и кнопка её удаления. Если страницу открыл не автор статьи, то пользователя переадресует на главную.

На шестой странице находится панель для написания/редактуры статьи, где можно указать заголовок статьи, её содержание и прикрепить картинку. Если на страницу зашёл не автор статьи, то пользователя переадресует на главную.

На седьмой странице содержится более подробная информация о профиле выбранного пользователя, его аватарка, описание, подписки и подписчики, опубликованные статьи.

На восьмой странице содержится панель для изменения информации в профиле пользователя. Если на страницу зашёл не владелец профиля, то пользователя переадресует на главную.

На девятой странице находится панель с полями для ввода юзернейма и пароля для входа в аккаунт.

На десятой странице находится панель с полями для ввода юзернейма, пароля и повтора пароля для регистрации аккаунта.

На таблице на рисунке 1 представлена схема перехода по страницам сайта.

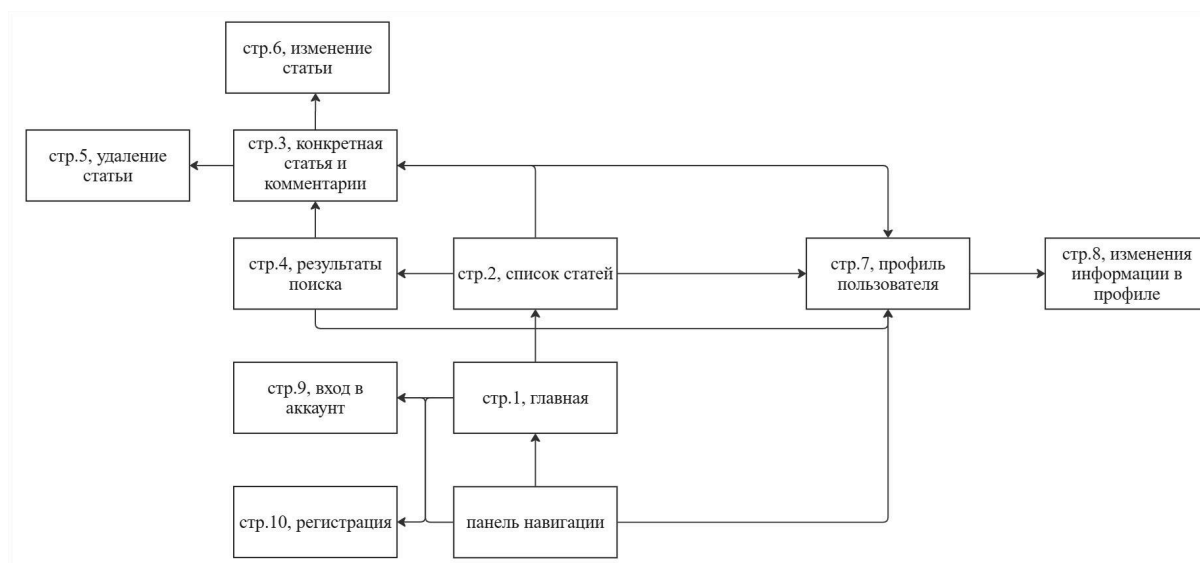


Рисунок 1 — Схема перехода между страницами сайта

Кнопки и гиперссылки сайта реагируют на наведение и нажатие с помощью CSS. Комментарии и посты на странице 2 и странице 3 при наведении увеличиваются в размере.

Для реализации функции регистрации/входа, публикации/редактирования/удаления статей, написания комментариев к статьям, поиска по статьям использовались Django и СУБД Sqlite3.

При регистрации пользователя информация о нём автоматически передаётся в базу данных с помощью моделей User и Profile, для профиля устанавливается стандартный аватар default.png.

При написании комментария/статьи информация также автоматически передаётся в базу данных.

Для поиска по статьям на странице со всеми статьями существует поле для поиска. При нажатии кнопки “найти” введённая информация передаётся в представление, которое с помощью модели Post получает из базы данных список всех статей, соответствующих требованиям. Эти статьи передаются на сайт и выводятся с помощью template tags.

Отображение комментариев и статей работает схожим образом, однако с помощью моделей Post и PostComment мы получаем все статьи и комментарии, которые далее сортируются по дате публикации и выводятся с помощью template tags.

Система отображения подписок и подписчиков в профиле пользователя работает также.

Информация о профиле пользователя получается с помощью модели Profile, которая получает из url адреса имя пользователя, чей профиль нас интересует.

Возможности взаимодействия пользователя с сайтом представлены на UML-диаграмме на рисунке 2.

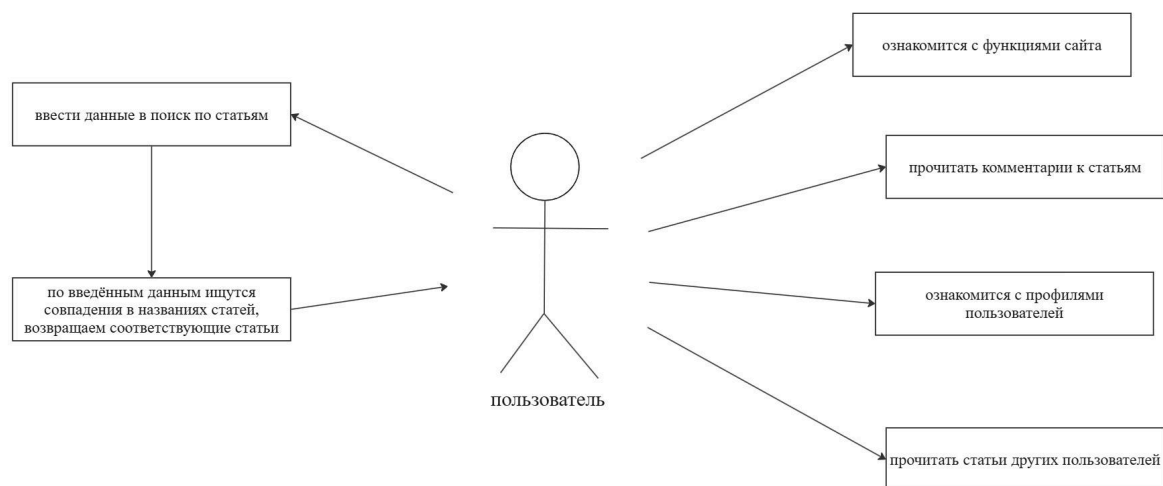


Рисунок 2 — UML-диаграмма взаимодействия пользователя с сайтом

База данных делится на 5 основных таблиц: `users_usr` для хранения информации о пользователе, `users_profile` для хранения информации о профиле пользователя, `user_profile_follows` для хранения информации о подписках пользователя, `posts_post` для хранения статей, `posts_postcomment` для хранения комментариев под статьями.

`users_profile` связана с `users_usr`, при удалении информации о пользователе из второй, удаляется информация о профиле пользователя из первой. `posts_post` связана с `users_profile` по схожему принципу, также как `posts_postcomment` с `posts_post`.

Схематично архитектура базы данных представлена на рисунке 3.

Рисунок 3 — Архитектура базы данных

Тестирование

Тестирование проекта включало в себя несколько ключевых этапов:

1. Юнит-тесты:

- Были написаны юнит-тесты для проверки корректности работы моделей и представлений. Например, проверка валидации данных для постов и комментариев.
- Проверка правильности работы форм регистрации и авторизации.

2. Функциональное тестирование:

- Проверка всех ключевых функций: создание постов, добавление комментариев, регистрация/авторизация пользователей.
- Проверка взаимодействия между компонентами: например, корректность отображения комментариев под соответствующими постами.

3. Интерфейсное тестирование:

- Тестирование взаимодействия пользователя с веб-интерфейсом: корректность отображения страниц, формы, кнопок и сообщений об ошибках.

ВЫВОД

В процессе разработки был изучен процесс создания веб-приложений и технологии с ними связанные, в т.ч. работа баз данных, Django, HTML, CSS.

Проект продемонстрировал основные возможности фреймворка Django для создания веб-приложений с функционалом публикации контента и взаимодействия с ним. Создание системы регистрации и авторизации пользователей добавляет важную безопасность и персонализацию для взаимодействия с сайтом. Тестирование показало, что проект стабилен, все функции работают корректно.

Данный проект может служить основой для более сложных веб-приложений и быть расширен новыми функциями, такими как рейтинги комментариев, возможность редактирования постов и комментариев и многое другое.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Django Documentation. Официальная документация Django [Электронный ресурс]. URL: <https://docs.djangoproject.com/en/stable/> (дата обращения: 18.12.2024).
2. Python Documentation. Официальная документация Python [Электронный ресурс]. URL: <https://docs.python.org/3/> (дата обращения: 20.12.2024).
3. Two Scoops of Django. О. М. Стерн, Д. В. Янг. Two Scoops of Django 3.0: Best Practices for the Django Web Framework. 3-е изд. США: Two Scoops Press, 2020. 504 с.
4. Django for Beginners. У. П. Холл. Django for Beginners [Электронный ресурс]. URL: <https://djangoforbeginners.com/> (дата обращения: 10.12.2024).
5. Django 3 By Example. А. М. Вакес. Django 3 By Example. 2-е изд. Великобритания: Packt Publishing, 2020. 554 с.