Т. 12. №5 2026
https://doi.org/10.33619/2414-2948/126

УДК 004.738.5:004.43 https://doi.org/10.33619/2414-2948/126/20

РАЗРАБОТКА ВЕБ-САЙТА НА ОСНОВЕ ТЕХНОЛОГИЙ PYTHON И DJANGO

©***Кудуев А. Ж.***, *ORCID: 0000-0002-3345-1364, SPIN-код: 8519-5251, канд. техн. наук, Ошский государственный университет, г. Ош, Кыргызстан, akuduev@oshsu.kg*
©***Адилбекова Н. А.***, *ORCID: 0009-0005-1891-0322, Ошский государственный университет, г. Ош, Кыргызстан, nadilbekova@oshsu.kg*
©***Ормош уулу Э.***, *ORCID: 0009-0003-6937-9223, Ошский государственный университет, г. Ош, Кыргызстан, 619123bema@gmail.com*
©***Чжао Ямэн***, *ORCID: 0009-0002-3291-2197, Ошский государственный университет, г. Ош, Кыргызстан, zhaoym828@163.com*

DEVELOPMENT OF A WEBSITE BASED ON PYTHON AND DJANGO TECHNOLOGIES

©***Kuduev A.***, *ORCID: 0000-0002-3345-1364, SPIN-code: 8519-5251, Ph.D., Osh State University, Osh, Kyrgyzstan, akuduev@oshsu.kg*
©***Adilbekova N.***, *ORCID: 0009-0005-1891-0322, Osh State University, Osh, Kyrgyzstan, nadilbekova@oshsu.kg*
©***Ormosh uulu E.***, *ORCID: 0009-0003-6937-9223 Osh State University, Osh, Kyrgyzstan, 619123bema@gmail.com*
©***Zhao Ya Ming,*** *ORCID: 0009-0002-3291-2197, Osh State University, Osh, Kyrgyzstan, zhaoym828@163.com*

Аннотация. Рассматривается процесс разработки веб-сайта с использованием языка программирования Python и веб-фреймворка Django. Актуальность темы обусловлена растущей потребностью в создании масштабируемых и безопасных веб-приложений в условиях современного рынка. Целью работы является анализ архитектурных особенностей Django и демонстрация практической реализации веб-проекта. В качестве методологии исследования используется анализ литературных источников и экспериментальное проектирование. В статье подробно описаны ключевые компоненты Django: модели (Models), представления (Views), шаблоны (Templates) и маршрутизация (URLs). Особое внимание уделяется реализации CRUD-операций и использованию объектно-реляционного отображения (ORM) для взаимодействия с базами данных на примере PostgreSQL. Результатом работы является создание прототипа веб-сайта — электронной доски объявлений. В заключении сформулированы выводы о преимуществах использования Django для веб-разработки и даны рекомендации по оптимизации производительности готовых решений.

Abstract. Discusses the process of developing a website using the Python programming language and the Django web framework. The relevance of the topic is due to the growing need for creating scalable and secure web applications in the modern market. The aim of the work is to analyze the architectural features of Django and demonstrate the practical implementation of a web project. The research methodology includes the analysis of literary sources and experimental design. The article describes in detail the key components of Django: Models, Views, Templates, and URLs. Special attention is paid to the implementation of CRUD operations and the use of Object-Relational Mapping (ORM) for interacting with databases using PostgreSQL as an example. The result of the work is the creation of a prototype website—an online bulletin board. The conclusion formulates

Т. 12. №5 2026
https://doi.org/10.33619/2414-2948/126

findings on the advantages of using Django for web development and offers recommendations for optimizing the performance of finished solutions.

Ключевые слова: Python, Django, веб-разработка, фреймворк, ORM, базы данных, PostgreSQL, CRUD, HTML, шаблонизация.

Keywords: Python, Django, web development, framework, ORM, databases, PostgreSQL, CRUD, HTML, templating.

В современном мире веб-технологии проникли во все сферы человеческой деятельности. От малого бизнеса до крупных корпораций — все стремятся иметь представительство в сети интернет. Спектр веб-сайтов огромен: от простых статичных страниц-визиток до сложных высоконагруженных порталов и интернет-магазинов. Выбор правильного инструмента для разработки является критическим фактором, влияющим на успех проекта, его стоимость, скорость вывода на рынок и простоту дальнейшей поддержки [3].

Язык программирования Python зарекомендовал себя как мощный и гибкий инструмент, который благодаря своей простоте и читаемости кода позволяет быстро создавать работающие прототипы и сложные программные комплексы. Одним из главных преимуществ Python в области веба является наличие зрелых и активно развивающихся фреймворков. Среди них особое место занимает Django — высокоуровневый фреймворк, который следует принципу "Don't Repeat Yourself" (DRY) и предоставляет разработчику практически всё необходимое «из коробки» (https://bhv.ru/).

Понимание архитектуры и принципов работы Django необходимо современному веб-разработчику для создания эффективных и безопасных решений.

Материал и методика

Теоретической базой исследования послужили научные статьи, посвященные веб-разработке, техническая документация, а также специализированная литература по фреймворку Django (https://bhv.ru/) [2, 3, 6].

Для достижения поставленной цели — демонстрации процесса разработки веб-сайта — был использован метод экспериментального проектирования. В качестве инструментария выступили: язык Python версии 3.11, фреймворк Django 5.1, система управления базами данных PostgreSQL и среда разработки PyCharm.

В ходе работы был спроектирован и реализован прототип веб-сайта — доска объявлений. Данный выбор обусловлен тем, что подобный проект позволяет охватить типовые для большинства веб-приложений задачи: работа с пользователями, создание, отображение и редактирование контента (CRUD), а также взаимодействие с базой данных.

Результаты и их обсуждение

Разработка на Django базируется на архитектурном паттерне Model-View-Template (MVT), который является вариацией классического MVC (Model-View-Controller). Рассмотрим каждый компонент в контексте разрабатываемого приложения «Доска объявлений».

Модели (Models) — это единственный, авторитетный источник информации о ваших данных. Она содержит все поля и поведение данных, которые вы храните. Django следует философии "DRY", позволяя определить модель данных в одном месте и автоматически генерировать на её основе миграции для базы данных. Для нашего проекта были созданы две основные модели: Category (категория) и Ad (объявление).

Т. 12. №5 2026
https://doi.org/10.33619/2414-2948/126

Модель Ad содержит поля для заголовка, текста объявления, цены, изображения, а также внешние ключи для связи с категорией и пользователем.

Листинг 1. Пример моделей для приложения "Доска объявлений"

```
python
from django.db import models
from django.contrib.auth.models import User

class Category(models.Model):
    name = models.CharField('Название категории', max_length=100)
    slug = models.SlugField(unique=True)

    def __str__(self):
        return self.name

    class Meta:
        verbose_name = 'Категория'
        verbose_name_plural = 'Категории'

class Ad(models.Model):
    title = models.CharField('Заголовок', max_length=200)
    description = models.TextField('Описание')
    price = models.DecimalField('Цена', max_digits=10, decimal_places=2)
    image = models.ImageField('Фото', upload_to='ads/%Y/%m/%d', blank=True)
    category = models.ForeignKey(Category, on_delete=models.CASCADE,
verbose_name='Категория')
    author = models.ForeignKey(User, on_delete=models.CASCADE, verbose_name='Автор')
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.title

    class Meta:
        verbose_name = 'Объявление'
        verbose_name_plural = 'Объявления'
        ordering = ['-created_at']
```

Использование ORM Django позволяет абстрагироваться от конкретной СУБД. В проекте использовали PostgreSQL, для чего в настройках settings.py необходимо было указать соответствующий движок и параметры подключения [5].

ORM преобразует код Python в SQL-запросы, что значительно упрощает взаимодействие с базой данных. Например, для получения последних 10 объявлений в определенной категории достаточно написать Ad.objects.filter(category__slug='some-slug').order_by('-created_at') [8].

Представления содержат логику обработки пользовательского запроса. Они взаимодействуют с моделями для получения данных и передают их в шаблоны для отображения. Django поддерживает как функциональные, так и классовые представления. Для проекта использование классового представления ListView для отображения списка объявлений позволило сократить количество шаблонного кода.

Листинг 2. Представление для отображения списка объявлений

```
python
from django.views.generic import ListView
```

Т. 12. №5 2026
https://doi.org/10.33619/2414-2948/126

```
from .models import Ad

class AdListView(ListView):
    model = Ad
    template_name = 'ads/ad_list.html'
    context_object_name = 'ads'
    paginate_by = 10

    def get_queryset(self):
        """Возвращает объявления, отфильтрованные по категории, если параметр slug передан в URL."""
        queryset = super().get_queryset()
        category_slug = self.kwargs.get('slug')
        if category_slug:
            queryset = queryset.filter(category__slug=category_slug)
        return queryset
```

Шаблоны в Django отвечают за представление данных. Они пишутся на HTML и содержат специальные теги шаблонизатора Django для вставки динамического контента [1, 8].

В проекте для создания интерфейса использовался Bootstrap 5 для обеспечения адаптивности и современного внешнего вида. Шаблоны наследуются от базового файла base.html, что позволяет избежать дублирования кода (принцип DRY). Маршрутизация (URLs) URL-адресов Django позволяет создавать чистые, понятные адреса без расширений файлов. Сопоставление URL с представлениями происходит с помощью регулярных выражений или специальных конвертеров путей. Для приложения были определены маршруты для главной страницы, страницы со списком объявлений по категориям и страницы детального просмотра объявления.

Листинг 3. Пример маршрутизации

```
python
from django.urls import path
from . import views

urlpatterns = [
    path('', views.AdListView.as_view(), name='home'),
    path('category/<slug:slug>/', views.AdListView.as_view(), name='category_detail'),
    path('ad/<int:pk>/', views.ad_detail, name='ad_detail'),
]
```

Для реализации возможности добавления и редактирования объявлений были использованы формы Django, связанные с моделью (ModelForm). Это автоматически создает HTML-формы на основе модели, генерирует поля и обеспечивает базовую валидацию данных. В сочетании с классовыми представлениями CreateView и UpdateView процесс создания таких страниц занимает минимум времени и кода, что является одним из ключевых преимуществ Django (https://clc.li/LfmUy).

После реализации базового функционала была проведена первичная оценка производительности с помощью встроенных инструментов Django и стандартных метрик, таких как время генерации страницы и количество запросов к базе данных. Для главной страницы, выводящей 10 объявлений, количество запросов к БД было минимальным. Однако, на странице детального просмотра объявления возникла проблема "N+1 запроса" при отображении данных автора, которая была решена с помощью метода select_related() в

Т. 12. №5 2026
https://doi.org/10.33619/2414-2948/126

представлении, что позволило объединить запросы и значительно ускорить загрузку страницы [2, 4].

В ходе выполнения работы была достигнута поставленная цель: разработан прототип веб-сайта «Доска объявлений» на базе Python и Django, проанализированы ключевые архитектурные компоненты фреймворка. Полученные результаты позволяют сделать следующие выводы:

Django предоставляет мощный и структурированный подход к веб-разработке, который благодаря встроенной ORM, админ-панели и системе шаблонов позволяет сосредоточиться на бизнес-логике приложения, а не на рутинных операциях.

Архитектура MVT способствует созданию чистого, поддерживаемого и масштабируемого кода.

Правильное использование инструментов Django, таких как классовые представления и select_related, критически важно для обеспечения высокой производительности готового продукта.

Выбор Django в качестве основы для проекта оправдан для создания широкого класса веб-приложений — от сайтов-визиток до сложных корпоративных порталов и интернет-магазинов.

Дальнейшее развитие работы может быть направлено на добавление асинхронных компонентов с помощью Django Channels для реализации реального времени (например, чатов между пользователями), а также на проведение более глубокого нагрузочного тестирования.

Список литературы:

1. Яценко Р. М. Web-технології. Методичні рекомендації до виконання курсових проєктів. Харків: ХНЕУ ім. С. Кузнеця, 2023. 15 с.

2. Жуков С. В., Ковалева О. А., Ковалев С. В. Концептуальная модель загрузки веб-страницы // Информационные технологии и вычислительные системы. 2024. № 2. С. 26-36.

3. Дронов В. А. Django 5. Практика создания веб-сайтов на Python. СПб.: БХВ-Петербург, 2025. 864 с.

4. Олейник П. П. Применение шаблона проектирования Read/Write Lock для управления доступом к функциональности сервера приложений // Journal of Instrument Engineering. 2007. №2. С. 51-54.

5. Графкин А. В., Мыльников Е. Н., Понамаренко Д. И. WEB – конструктор масштабируемых робототехнических систем моделирования сложных поверхностей для динамических испытаний // Вестник молодежной науки. 2022. № 2(21). С. 78-86.

6. Романов М. А., Бадалин А. О., Рыбалка Д. В., Баженов А. Э. Анализ и сравнительное исследование архитектурных подходов к разработке одностраничных приложений (SPA, SSR, SSG) // Программные системы и вычислительные методы. 2025. №4. С. 108-117.

7. Шворин А.Б. Параллельное сложение вещественных чисел в системах счисления с перекрытием // Программные системы: теория и приложения. 2015. Т. 6, № 2(25). С. 101-117.

8. Чернецкий И. И. Создание инновационного метода адаптивной веб-разработки для повышения производительности и удобства веб-приложений // Молодой ученый. 2024. №29 (528). С. 25-33.

References:

1. Yatsenko, R. M. (2023). Web-tekhnologiï. Metodichni rekomendatsiï do vikonannya kursovikh proєktiv. Kharkiv. (in Russian).

2. Zhukov, S. V., Kovaleva, O. A., Kovalev, S. V. (2024). Kontseptual′naya model′ zagruzki veb-stranitsy. *Informatsionnye tekhnologii i vychislitel'nye sistemy,* (2), 26-36. (in Russian).

3. Dronov, V. A. (2025). Django 5. Praktika sozdaniya veb-sajtov na Python. St. Petersburg. (in Russian).

4. Olejnik, P. P. (2007). Primenenie shablona proektirovaniya Read/Write Lock dlya upravleniya dostupom k funktsional′nosti servera prilozhenij. *Journal of Instrument Engineering,* (2), 51-54. (in Russian).

5. Grafkin, A. V., Myl′nikov, E. N., & Ponamarenko, D. I. (2022). WEB – konstruktor masshtabiruemykh robototekhnicheskikh sistem modelirovaniya slozhnykh poverkhnostej dlya dinamicheskikh ispytanij. *Vestnik molodezhnoj nauki,* (2(21)), 78-86. (in Russian).

6. Romanov, M. A., Badalin, A. O., Rybalka, D. V., & Bazhenov, A. E. (2025). Analiz i sravnitel′noe issledovanie arkhitekturnykh podkhodov k razrabotke odnostranichnykh prilozhenij (SPA, SSR, SSG). *Programmnye sistemy i vychislitel'nye metody,* (4), 108-117. (in Russian).

7. Shvorin, A. B. (2015). Parallel′noe slozhenie veshchestvennykh chisel v sistemakh schisleniya s perekrytiem. *Programmnye sistemy: teoriya i prilozheniya, 6*(2(25)), 101-117. (in Russian).

8. Chernetskij, I. I. (2024). Sozdanie innovatsionnogo metoda adaptivnoj veb-razrabotki dlya povysheniya proizvoditel′nosti i udobstva veb-prilozhenij. *Molodoj uchenyj,* (29 (528)), 25-33. (in Russian).

Поступила в редакцию
12.03.2026 г.

Принята к публикации
20.03.2026 г.

Ссылка для цитирования:

Кудуев А. Ж., Адилбекова Н. А., Ормош уулу Э., Чжао Ямэн Разработка веб-сайта на основе технологий Python и Django // Бюллетень науки и практики. 2026. Т. 12. №5. С. 173-178. https://doi.org/10.33619/2414-2948/126/20

Cite as (APA):

Kuduev, A., Adilbekova, N., Ormosh uulu, E., & Zhao, Ya Ming (2026). Development of a Website Based on Python and Django Technologies. *Bulletin of Science and Practice, 12*(5), 173-178. (in Russian). https://doi.org/10.33619/2414-2948/126/20