

УДК 004.42

РАЗРАБОТКА BACKEND ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ БРОНИРОВАНИЯ МЕДИЦИНСКИХ УСЛУГ

Кряжева Е.В.,

к. псих. н., доцент,

Калужский государственный университет им. К.Э. Циолковского,

Калуга, Россия

Салдаева А.А.,

студент,

Калужский государственный университет им. К.Э. Циолковского,

Калуга, Россия

Аннотация.

В статье рассматривается проблема разработки серверной части (backend) для системы бронирования услуг массажистов и физиотерапевтов. Итоги каждого этапа разработки представлены в сводной таблице, там же описаны применяемые инструменты и решаемые задачи. Основным технологическим решением стал фреймворк Django, который обладает набором готовых модулей для эффективной разработки веб-приложений, включая интеграцию с базами данных и систему аутентификации. Клиентская часть построена на технологиях HTML, CSS и JavaScript. В конце статьи представлены выводы по работе, обращается внимание на эффективность данного подхода.

Ключевые слова: веб-приложение, предметная область, backend, Django, PostgreSQL, HTML, CSS и JavaScript

DEVELOPMENT OF A BACKEND WEB APPLICATION FOR BOOKING MEDICAL SERVICES

Kryazheva E. V.,

Ph.D. in Psychology, Associate Professor,

Дневник науки | www.dnevniknauki.ru | СМН ЭЛ № ФС 77-68405 ISSN 2541-8327

*K. E. Tsiolkovsky Kaluga State University,
Kaluga, Russia*

Saldaeva A. A.,

Student,

*K. E. Tsiolkovsky Kaluga State University,
Kaluga, Russia*

Annotation.

The article considers the problem of developing the server part of a web application for a system for booking massage therapists and physiotherapists. The results of each stage of development are presented in a table demonstrating the tools used and the tasks to be solved. The main technological solution was the Django framework, which has a set of ready-made modules for the effective development of web applications, including integration with databases and an authentication system. The frontend solution is built on traditional HTML, CSS and JavaScript technologies, which ensures ease of development and effective separation of responsibilities between the client and the server. At the end, conclusions on the work are presented, attention is drawn to the effectiveness of this approach.

Keywords: web application, subject area, backend, Django, PostgreSQL, HTML, CSS and JavaScript

Разработка серверной части веб-приложений представляет собой многоэтапный процесс, включающий проектирование, реализацию, тестирование и развертывание системы. Каждый этап требует применения современных технологий и инструментов для обеспечения стабильности, безопасности и производительности приложения.

Разработка веб-приложения для бронирования услуг массажистов и физиотерапевтов является актуальной задачей, направленной на создание удобной и функциональной системы для автоматизации записи и управления

Дневник науки | www.dnevniknauki.ru | СМИ Эл № ФС 77-68405 ISSN 2541-8327

услугами. В данной работе описывается реализация backend для системы бронирования услуг массажистов и физиотерапевтов, где особое внимание было уделено архитектуре базы данных, обработке бронирований и управлению ролями пользователей.

Система включает в себя ряд модулей, каждый из которых выполняет свою роль. Модуль аутентификации и авторизации обеспечивает безопасность данных пользователей и доступ к функционалу системы. Модули управления услугами, расписанием и бронированиями интегрированы для предоставления пользователям простого и интуитивного интерфейса, где они могут выбрать подходящие услуги и свободное время. Модуль уведомлений автоматизирует взаимодействие между системой и пользователями, напоминая им о важных событиях. Модуль отзывов способствует улучшению качества предоставляемых услуг, позволяя клиентам оставлять обратную связь.

Основные этапы разработки backend-приложения сведены в таблицу 1, где указаны ключевые задачи и технологии, использованные в данном проекте.

Таблица 1 — Основные этапы разработки backend-части веб-приложения

№	Этап	Основные задачи	Реализация в проекте
1	Постановка задачи и анализ требований	- Определение целей и задач системы. - Анализ потребностей пользователей (клиенты и специалисты). - Формирование списка функциональных возможностей. - Выбор технологий.	- Функционал: бронирование, управление расписанием, аутентификация. - Технологии: Django 5.1.4, PostgreSQL, Django REST Framework.
2	Проектирование архитектуры	- Определение архитектурного подхода. - Проектирование базы данных. - Разработка API-эндпоинтов.	- Монолитная архитектура. - Созданы модели: CustomUser, SpecialistProfile, Booking, WorkingHours, Service, Review. - API: разработаны эндпоинты REST для взаимодействия с фронтендом.
3	Разработка функционала	- Настройка окружения. - Создание моделей и миграций. - Реализация бизнес-логики.	- Django ORM для работы с БД. - JWT-аутентификация через DRF.

		- Настройка аутентификации и авторизации.	- Разработаны API: бронирования, отзывы, услуги.
4	Тестирование системы	- Юнит-тестирование. - Интеграционное тестирование API. - Проверка производительности.	- Написаны тесты для моделей и API. - Инструменты: Django TestCase, pytest, Postman.
5	Развёртывание	- Настройка сервера и базы данных. - Интеграция с внешними сервисами. - Обеспечение безопасности.	- Размещение на Gunicorn + Nginx. - PostgreSQL развернут на сервере. - Настроен HTTPS и ограничение доступа.
6	Сопровождение и обновление	- Мониторинг производительности. - Регулярные обновления. - Реализация новых функций.	- Логирование через Django Logging. - Автоматизация развертывания (CI/CD). - Поддержка API версионирования.

Реализация системы бронирования услуг массажистов и физиотерапевтов потребовала детального анализа доступных технологий, направленного на обеспечение функциональности, производительности и безопасности. Первостепенной задачей стало формирование технологического стека, способного поддерживать как серверную логику, так и взаимодействие с пользователем, что отражено в работах по проектированию веб-приложений [1].

В качестве основы серверной части был выбран фреймворк Django, предоставляющий готовые решения для разработки сложных веб-систем. Его встроенные модули, такие как ORM для взаимодействия с базой данных и система аутентификации, значительно ускорили процесс создания бизнес-логики. Для организации взаимодействия между клиентской и серверной частями было интегрировано расширение Django REST Framework, упрощающее разработку REST API. Данный выбор обусловлен необходимостью обеспечения стабильности и соответствия современным стандартам, что подтверждается исследованиями в области веб-разработки [3].

Важным аспектом при внедрении Django стала его встроенная система безопасности. Фреймворк автоматически предотвращает распространённые

угрозы, включая SQL-инъекции и XSS-атаки, а также поддерживает HTTPS и механизмы хэширования паролей. Эти возможности исключили необходимость использования сторонних инструментов для защиты данных, что особенно критично для сервисов, где конфиденциальность пользователей является приоритетом [4].

Для хранения данных была внедрена реляционная СУБД PostgreSQL, выбранная благодаря поддержке транзакций и сложных запросов [5]. Её интеграция с Django через встроенный адаптер позволила эффективно управлять расписаниями специалистов и бронированиями, обеспечивая целостность данных даже при высокой нагрузке.

Клиентская часть системы реализована на базе стандартных веб-технологий — HTML, CSS и JavaScript. Такой подход минимизировал сложность разработки интерфейса, сосредоточив усилия на серверной логике. Шаблоны Django использовались для динамического формирования страниц, а статические файлы — для подключения стилей и скриптов. Это решение соответствует принципу разделения ответственности между фронтендом и бэкендом, упрощая дальнейшее масштабирование.

Язык программирования Python, лежащий в основе Django, был выбран благодаря лаконичному синтаксису и широкому набору библиотек. Его преимущества, такие как читаемость кода и активное сообщество разработчиков, упростили реализацию как базового функционала, так и дополнительных задач, включая обработку данных и интеграцию со сторонними сервисами [1].

Совокупность выбранных технологий — Django, PostgreSQL и стандартных веб-инструментов — обеспечила баланс между надёжностью и скоростью разработки. Использование проверенных решений с открытым исходным кодом снизило риски, связанные с безопасностью и поддержкой, а их совместимость подтвердила эффективность в условиях реальных нагрузок [2].

Таким образом, проведённый анализ и обоснованный выбор технологий позволили создать структуру, соответствующую требованиям проекта по Дневник науки | www.dnevniknauki.ru | СМИ Эл № ФС 77-68405 ISSN 2541-8327

разработке backend для системы бронирования услуг массажистов и физиотерапевтов. Это обеспечило не только корректную работу системы бронирования, но и заложило основу для её дальнейшего развития, включая добавление новых функций и оптимизацию производительности.

Библиографический список:

1. Вин Ч. Как спроектировать современный сайт / Ч. Вин. - Пер. с англ. - СПб.: Питер, 2011. - 192 с.
2. Меле, Антонио. Django 4 в примерах. Разрабатывайте мощные и надежные веб-приложения на Python / Антонио Меле. — М.: ДМК-Пресс, 2023. — 800 с.
3. Рыбникова, С. В. Разработка информационного ресурса салона красоты / С. В. Рыбникова, В. А. Помазан // Евразийский научный журнал. – 2016. – № 8. – С. 265-267.
4. Слюсарь, Е. Ю. Безопасность веб-приложений - атаки с использованием SQL-инъекций / Е. Ю. Слюсарь, Р. С. Каграманян, А. М. Кумратова // Цифровизация экономики: направления, методы, инструменты : Сборник материалов I всероссийской студенческой научно-практической конференции, Краснодар, 21–25 января 2019 года. – Краснодар: Кубанский государственный аграрный университет имени И.Т. Трубилина, 2019. – С. 302-304.
5. Форсье, Джефф; Биссекс, Пол; Чан, Уэсли Дж. Django. Разработка веб-приложений на Python / Джефф Форсье, Пол Биссекс, Уэсли Дж. Чан. — СПб.: Символ-Плюс, 2017. — 442 с.

Оригинальность 77%