

BACKEND-АРХИТЕКТУРА ПЛАТФОРМЫ ДЛЯ ИССЛЕДОВАТЕЛЬСКОЙ ДЕЯТЕЛЬНОСТИ СТУДЕНТОВ

Поначугин А.В., к.экон.н., доцент

Добрыдень И.А., студент

ФГБОУ ВО «НГПУ им. Козьмы Минина» г. Нижний Новгород, Россия

Аннотация. В статье обсуждается развитие цифровых технологий в сфере образования, которое требует использование действенных инструментов для участия студентов в научной деятельности. Важную роль играет создание надежной backend-архитектуры, которая обеспечит стабильную работу платформы, а также хранение, передачу и безопасность данных в системе. Исследование охватывает современные методы проектирования серверной части для исследовательских платформ. В статье рассматриваются основные блоки архитектуры и выделяются ключевые решения, которые делают систему производительной и надежной.

Ключевые слова: backend-архитектура, платформа, цифровое образование, исследование, безопасность.

На настоящий момент сфера образования и научных исследований все больше переносится в цифровое пространство. Учреждения среднего и высшего образования пытаются создавать и продвигать платформы, научные проекты. Это происходит в совместной деятельности с организациями, преподавателями и студентами. Для того, чтобы данная система надежно и безопасно работала, нужно создать качественную серверную часть платформы - backend, который отвечает за обработку, хранение и передачу информации между членами данного проекта.

Backend является «двигателем» любой цифровой среды. Он руководит логикой приложения, обрабатывает запросы пользователь, а также обеспечивает взаимодействие с базой данных. Для такой платформы характерно, чтобы серверная часть была не только производительной, но и гибкой, ведь нагрузка на такие платформы с каждым годом растёт.

Основная ключевая составляющая при проектировании backend-архитектуры - это масштабируемость. Большинство научных платформ, как правило, запускают её с ограниченного количества пользователей, но постепенно расширяют свой проект в процессе распространения числом пользователей. Поэтому данная архитектура обязана обеспечивать возможность горизонтального расширения системы, разделения сервисов на более мелкие компоненты и гибкое распределение нагрузки. Для этого в настоящее время часто применяется микросерверный подход, в нем система делится на небольшие независимые компоненты, называемыми серверами, каждый из них отвечает за конкретный участок функциональности.

Например, некоторые сервисы могут обрабатывать регистрацию студентов и педагогов, управлять их проектами, хранить публикацию работ, обеспечивать общение между пользователями. Данный метод поможет обновлять и развивать отдельные компоненты без вынужденной остановки системы, этот шаг очень важен для образовательной среды, ведь доступ к этим платформам должен осуществляться ежедневный, круглосуточный.

Отдельно хочется выделить следующий аспект - обеспечение безопасности и защите данных пользователей, ведь платформа должна хранить личную информацию как студентов, так педагогов и организаций, результаты, процесс исследований, чтобы данные не могли передаваться в сторонние компании, в том числе и файлы публикаций должны иметь необходимую многоуровневую систему защиты. Здесь необходимо использовать шифрование данных, например, использование токенов для авторизаций, разграничение уровней доступа пользователей, защиту от типичных уязвимостей, таких как XSS-атаки и SQL-инъекции, ведь без надежной системы безопасности содержание и использование данной платформы будет небезопасным[1].

При проектировании backend-части проекта необходимо продумать взаимодействие с frontend-частью. Для этого часто используют GraphQL, который обеспечивает обмен данными. Данный API является главным звеном между интерфейсом платформы и логикой самого приложения, позволяя всем пользователям получать доступ к нужным функциям, таких как создание проектов, добавление участников, обмен файлами, общение, а также комментирование результатов[4,5].

Обработка больших объемов данных - это одна из особенностей любой

исследовательской платформы. Она может состоять из статей, графиков, изображений, отчетов или же отзывами о проектах. Для хранения любых данных необходимо использовать реляционные базы данных (MySQL/PostgreSQL), системы хэширования (Redis), которые нужны для ускорения доступа. Важно также соблюдать принципы нормализации данных, чтобы не создавались копии данных, а также чтобы нагрузка на сайт не превышала допустимых значений.

Важным элементом нынешней backend-разработки будет интеграция каких-либо DevOps-практик, таких как контейнеризация (Docker) и интеграция или развертывание (CI/CD). Они помогают автоматизировать обновление любой системы, используются для минимизирования ошибок при использовании новых версий и обеспечения стабильностью платформы при активной разработке. Данный подход делает возможным внедрение новых функций, ведь это очень важно при улучшении университетских проектов, где требования имеют возможность меняться в течение учебного года[2].

Backend-архитектура требует не только технические аспекты проектирования, но также и понимание логики взаимодействия пользователей с системой. В ней должны быть выделены следующие роли: студент, преподаватель (научный руководитель), а также представитель компании. Каждый из них должен иметь свой набор прав и функций начиная от создания проектов, заканчивая проверкой отчетов и дальнейшей публикацией результатов. Архитектура в свою очередь должна обеспечивать бесперебойную работу всех пользователей по ролям без утечек данных и конфликтов[3].

Таким образом, создание backend-архитектуры для научных исследований руководителей организаций, преподавателей и студенты становится значимой задачей, которая включает архитектурный дизайн, безопасность и хранение данных и интеграции различных сервисов, ведь четко сформулированный backend помогает обеспечить устойчивость всей информационной среды, а также возможность дальнейшего развития и удобство работы всех пользователей.

Подводя итоги, нужно не забывать, что успех различных информационных систем зависит не только от простого программирования, но и от их грамотного сочетания. При создании микро сервисной архитектуры нужно учитывать современные инструменты, например, контейнеризация, кодирование информации и т.п., а также систем безопасности, ведь благодаря им можно создавать надежные и гибкие платформы, способствующие поддерживать научную деятельность пользователь. Данные решения становятся будущем цифрового пространства современного образования, способствуя созданию новой исследовательской среды в нынешних университетах.

Литература

1. Волкова, О. Н. Сущность понятия и классификация угроз информационной безопасности / О. Н. Волкова, А. А. Садакова // Образование в цифровую эпоху: опыт, проблемы и перспективы, Нижний Новгород, 21–22 декабря 2023 года. – Нижний Новгород: Нижегородский государственный педагогический университет им. К. Минина, 2024. – С. 90-94. – EDN JYVRNE.
2. Золотухина, Д. Ю. Эффективность распределённых кэширующих платформ в современных backend-архитектурах: сравнительный анализ Redis и Hazelcast // Программные системы и вычислительные методы. – 2024. – №4 . URL: <https://cyberleninka.ru/article/n/effektivnost-raspredelyonnyh-keshiruyuschih-platform-v-sovrem>

ennyh-backend-arhitekturah-sravnitelnyy-analiz-redis-i-hazelcast (дата обращения: 10.10.2025).

3. Лисай, В. Э. Безопасность в backend-разработке: современные методы защиты и шифрования данных // Вестник науки. – 2024. – №4 (73). URL: <https://cyberleninka.ru/article/n/bezopasnost-v-backend-razrabotke-sovremennye-metody-zaschity-i-shifrovaniya-dannyh> (дата обращения: 10.10.2025).

4. Сафонов, А. А. Цифровая трансформация образования : учебник и практикум для вузов / А. А. Сафонов, М. А. Сафонова. – Москва : Издательство Юрайт, 2025.

–
100 с.

–
URL: <https://urait.ru/bcode/569813> (дата обращения: 10.10.2025).

5. Темников, Д. О., Дубинин Р. С. Проектирование отказоустойчивых и резервированных backend-систем // Universum: технические науки.

–
2025.

–
№6 (135). URL: <https://cyberleninka.ru/article/n/proektirovanie-otkazoustoychivyh-i-rezervirovannyh-backend-sistem> (дата обращения: 10.10.2025).