

В. А. Болтава^{1✉}, А. А. Шаранов¹

Разработка веб-платформы для публикации проектов СГУГиТ с применением языка программирования Go

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: vladimir.boltava@yandex.ru

Аннотация. В статье рассматривается процесс создания веб-платформы для публикации и взаимодействия с проектами студентов и сотрудников СГУГиТ. Основу серверной части составляет язык программирования Go. В качестве СУБД используется PostgreSQL для хранения основной информации о проектах и пользователях, а MongoDB применяется для хранения файлов (изображений) и временных данных регистрации. Описана монолитная архитектура, ролевая модель (гость, пользователь, администратор) и модули системы: регистрация, авторизация, управление проектами и профилями. Приведены особенности реализации безопасности, распределенного хранения и сроков хранения временных данных.

Ключевые слова: Go, Golang, PostgreSQL, MongoDB, веб-платформа, монолит, регистрация, авторизация, веб-интерфейс, REST API

V. A. Boltava^{1✉}, A. A. Sharapov¹

Development of web platform for publishing SSUGT projects using the Go programming language

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: vladimir.boltava@yandex.ru

Abstract. The article discusses the process of creating a web platform for publishing and interacting with projects of students and staff of SSUGT. The Go programming language forms the basis of the server side. PostgreSQL is used as a database to store basic information about projects and users, while MongoDB is used to store files (images) and temporary registration data. The monolithic architecture, role model (guest, user, administrator), and system modules are described: registration, authorization, project management, and profiles. The features of the implementation of security, distributed storage and storage time of temporary data are given.

Keywords: Go, Golang, PostgreSQL, MongoDB, web platform, monolith, registration, authorization, web interface, REST API

Введение

Цифровизация образования требует создания универсальных онлайн-платформ, способных эффективно структурировать и демонстрировать результаты научной и учебной деятельности. Для СГУГиТ подобный инструмент позволяет централизованно управлять проектами студентов и преподавателей, обеспечивая их доступность и сохранность.

Проект веб-платформы реализуется с применением языка программирования Go [1], который был выбран благодаря сочетанию высокой производительности, простоты разработки, встроенной поддержки многопоточности и строгой

типизации. Эти особенности делают Go особенно подходящим для создания надежных серверных приложений, обрабатывающих большое количество параллельных запросов. Основной целью разработки стало создание устойчивой, масштабируемой системы с удобным пользовательским интерфейсом, безопасной архитектурой и гибкой моделью разграничения прав доступа.

Методы и материалы

Для реализации серверной части платформы была выбрана монолитная архитектура [5] на языке программирования Go [1], обеспечивающем высокую скорость обработки запросов, простоту масштабирования и безопасность. В качестве основной реляционной базы данных использовалась PostgreSQL [2], что позволило эффективно управлять структурированной информацией о пользователях, проектах и их атрибутах. Дополнительно применялась MongoDB [3] – она использовалась как хранилище для файлов (изображений проектов), а также для временного хранения регистрационных данных и кода подтверждения электронной почты. Механизм хранения с ограниченным временем жизни записей (TTL) позволил реализовать автоматическую очистку таких данных через 10 минут после создания.

Аутентификация и авторизация пользователей реализованы с использованием JWT (JSON Web Token) [6], что обеспечивает безопасность доступа и возможность разграничения прав на уровне API. Все компоненты системы взаимодействуют через REST-интерфейсы [7]. Пользователи системы делятся на три роли: гость, пользователь и администратор. Гости могут регистрироваться, авторизовываться и просматривать общедоступные проекты. Зарегистрированные пользователи получают доступ к созданию, редактированию и удалению своих проектов, а также к управлению личным профилем. Администраторы, помимо указанных возможностей, могут просматривать панель администратора и обрабатывать созданные проекты и либо подтверждать их публикацию, либо отклонять с указанием причины.

Пользовательский интерфейс реализован на библиотеке React [4], что позволило создать динамичное и отзывчивое одностраничное приложение. Визуальная часть выполнена в светло-серой цветовой гамме с акцентом на простоту и доступность взаимодействия. Интерфейс интуитивно понятен, поддерживает навигацию между проектами, профилями и административными функциями (рис. 1–4).

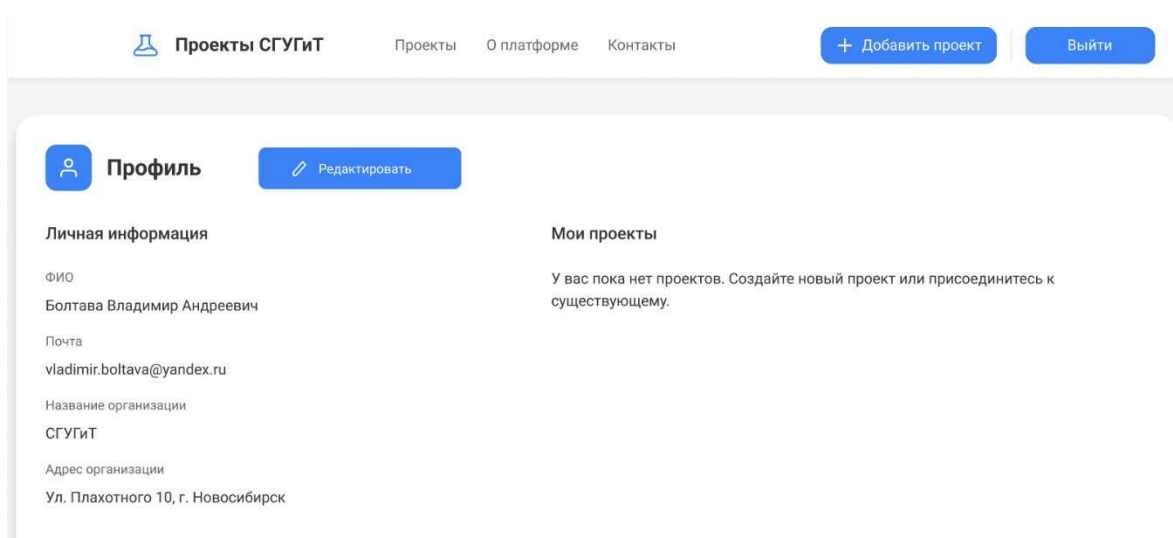


Рис. 1. Страница профиля пользователя

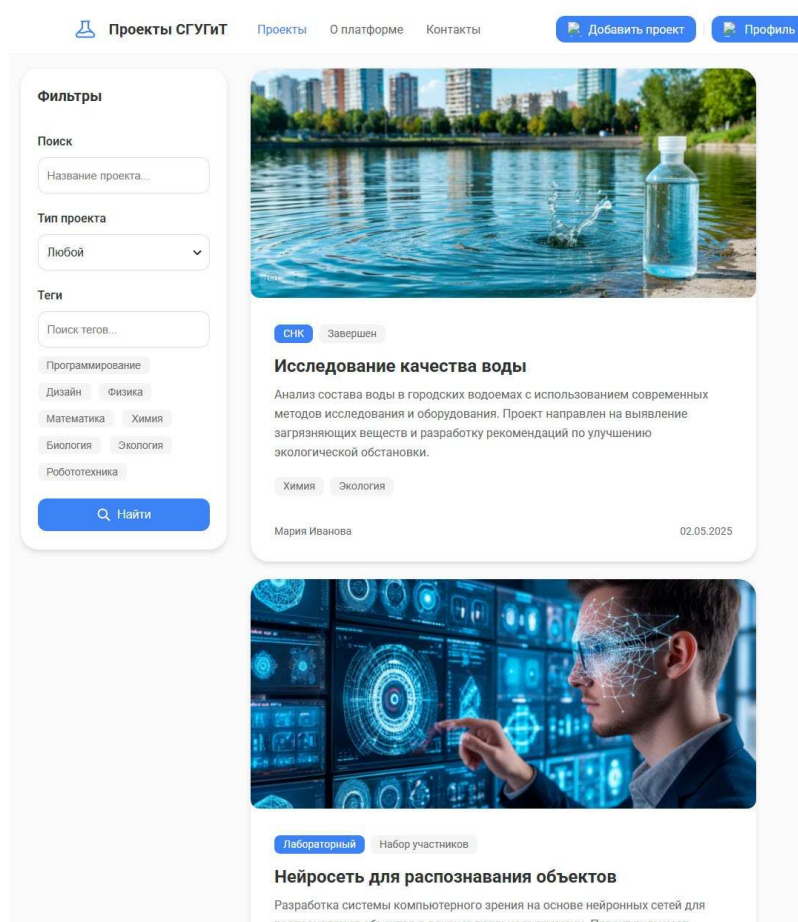


Рис. 2. Страница поиска проектов

Рис. 3. Страница создания проекта

Рис. 4. Страница модерирования проектов

Результаты

В ходе разработки создано программное обеспечение, обеспечивающее:

1. Регистрацию и авторизацию пользователей с подтверждением электронной почты.
2. Разграничение прав доступа на основе ролей (гость, пользователь, администратор).
3. Создание, редактирование, удаление и публикацию проектов.

4. Управление пользовательским профилем и просмотр профилей других участников.

5. Хранение структурированной информации в PostgreSQL и файлов в MongoDB.

6. Безопасную работу через REST API с использованием JWT.

7. Интуитивно понятный веб-интерфейс на базе React.

Разработка платформы активно продолжается: серверная и клиентская части интегрированы и проходят функциональное тестирование. Платформа уже демонстрирует устойчивую работу, и в ближайшее время планируется ее внедрение в учебный процесс.

Обсуждение

Выбор языка Go и монолитной архитектуры был обусловлен необходимостью быстрой и надежной реализации проекта с минимальными накладными расходами на инфраструктуру. Использование PostgreSQL обеспечило стабильную работу с ключевыми сущностями платформы, а MongoDB – удобную и производительную обработку файлов и временных данных. Ролевое разграничение пользователей и безопасная авторизация на основе JWT создают прочную основу для дальнейшего масштабирования системы.

REST API упростил интеграцию клиентской части с сервером и повысил читаемость архитектурных решений. В процессе разработки особое внимание уделялось гибкости интерфейса, чтобы обеспечить доступность функционала для разных групп пользователей.

Заключение

Разработанная веб-платформа обеспечивает централизованное хранение, управление и представление студенческих и исследовательских проектов. В дальнейшем планируется внедрение системы комментариев и обратной связи, добавление рейтинговой системы, которая поможет выделять наиболее значимые и успешные работы, а также разработка мобильной версии или полноценного приложения для повышения доступности платформы на различных устройствах.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Go Documentation // Документация. URL: <https://go.dev/doc/> (дата обращения: 20.04.2025).
2. PostgreSQL Documentation // Документация. URL: <https://www.postgresql.org/docs/> (дата обращения: 20.04.2025).
3. MongoDB Documentation // Документация. URL: <https://www.mongodb.com/docs/> (дата обращения: 20.04.2025).
4. React Documentation // Документация. URL: <https://react.dev/> (дата обращения: 20.04.2025).
5. Monolithic Architecture - System Design // Документация. URL: <https://www.geeksforgeeks.org/monolithic-architecture-system-design/> (дата обращения: 20.04.2025).
6. JSON Web Tokens (JWT): Introduction // Документация. URL: <https://jwt.io/introduction/> (дата обращения: 20.04.2025).
7. RESTful API Design Guide // Документация. URL: <https://restfulapi.net/> (дата обращения: 20.04.2025).

8. Шеховцов В.Ю. Применение языка Go для разработки веб-приложений // КиберЛенинка. URL: <https://cyberleninka.ru/article/n/primenenie-yazyka-go-dlya-razrabotki-veb-prilozheniy> (дата обращения: 20.04.2025).
9. Коротков А.А. Построение высоконагруженных систем на PostgreSQL // КиберЛенинка. URL: <https://cyberleninka.ru/article/n/postroenie-vysokonagruzhennyh-sistem-na-postgresql> (дата обращения: 20.04.2025).
10. Савинов И.А. Современные подходы к хранению данных: сравнение реляционных и нереляционных СУБД // КиберЛенинка. URL: <https://cyberleninka.ru/article/n/sravnenie-relyatsionnyh-i-nerelelyatsionnyh-subd> (дата обращения: 20.04.2025).
11. Левин С.Ю. Использование MongoDB для хранения мультимедийной информации в веб-приложениях // КиберЛенинка. URL: <https://cyberleninka.ru/article/n/ispolzovanie-mongodb-dlya-hraniyeniya-multimediyuoy-informatsii> (дата обращения: 20.04.2025).
12. Чумаков Д.С. Разработка клиентской части веб-приложений на React // Молодой ученый. URL: <https://moluch.ru/archive/388/86138/> (дата обращения: 20.04.2025).
13. Иванов П.В. Роль фронтенд-разработки в построении пользовательских интерфейсов // КиберЛенинка. URL: <https://cyberleninka.ru/article/n/rol-front-end-razrabotki-v-postroenii-interfeysov> (дата обращения: 20.04.2025).
14. Дьяконов А.А. Архитектура монолитных и микросервисных веб-приложений: сравнение подходов // КиберЛенинка. URL: <https://cyberleninka.ru/article/n/monolitnaya-i-mikroservisnaya-arhitektura-veb-prilozheniy> (дата обращения: 20.04.2025).
15. Razon J. Go Web Development Cookbook // Документация. URL: <https://www.packtpub.com/product/go-web-development-cookbook/9781838554491> (дата обращения: 20.04.2025).
16. Lungu I. Mastering PostgreSQL in Application Development // Документация. URL: <https://masteringpostgresql.com/> (дата обращения: 20.04.2025).
17. Mehta H. MongoDB Basics and Real-World Projects // Документация. URL: <https://link.springer.com/book/10.1007/978-1-4842-7060-2> (дата обращения: 20.04.2025).
18. Accomazzo A. Fullstack React: The Complete Guide to ReactJS and Friends // Документация. URL: <https://www.fullstackreact.com/> (дата обращения: 20.04.2025).
19. Власов Д.А. Основы проектирования пользовательских интерфейсов // КиберЛенинка. URL: <https://cyberleninka.ru/article/n/osnovy-proektirovaniya-polzovatelskih-interfeysov> (дата обращения: 20.04.2025).
20. Малышев А.В. Разработка информационных систем в образовательной среде вуза // КиберЛенинка. URL: <https://cyberleninka.ru/article/n/razrabotka-informatsionnyh-sistem-v-obrazovatelnoy-srede-vuza> (дата обращения: 20.04.2025).

© В. А. Болтава, А. А. Шаранов, 2025