

ПРОЕКТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ИНТЕРАКТИВНОГО ТЕСТИРОВАНИЯ ПО ТЕОРИИ ГРАФОВ

Хакимова Лейсян Асхатовна

студент, Казанский национальный исследовательский технический университет им. А.Н.
Туполева – КАИ, РФ, г. Казань

В условиях стремительного развития цифровых образовательных технологий особую актуальность приобретают программные решения, направленные на автоматизацию процессов контроля и оценки знаний. Это особенно важно в преподавании дисциплин, требующих не только теоретического освоения материала, но и практических умений работы с абстрактными структурами. Теория графов – одна из таких дисциплин, широко применяемая в области компьютерных сетей, логистики, анализа данных и алгоритмического моделирования. Её специфика требует от студентов не только понимания формальных определений и теорем, но и уверенного владения алгоритмами, обеспечивающими решение прикладных задач.

Предлагаемая система построена на основе клиент-серверной архитектуры, обеспечивающей разделение логики между интерфейсом пользователя и обработкой данных. Такая модель позволяет повысить надёжность системы, упростить поддержку и масштабирование, а также обеспечить гибкость при интеграции с внешними сервисами.

Клиентская часть разработана с применением языка JavaScript [1] и библиотеки React [2], что позволило реализовать одностраничную структуру приложения (SPA — Single Page Application) [3]. Это обеспечивает высокую скорость отклика, динамическое обновление интерфейса и удобство взаимодействия для пользователя. Для визуализации и манипуляции графами используется библиотека Cytoscape.js [4], позволяющая реализовать интерактивные элементы: перетаскивание вершин, раскраску, отображение направленности и весов ребер. Подобный подход не только способствует лучшему восприятию учебного материала, но и предоставляет инструменты для визуальной отработки алгоритмов.

Серверная часть реализована на базе облачного решения Supabase [5], которое выступает в роли backend-as-a-service (BaaS). Оно предоставляет готовую инфраструктуру для хранения информации в реляционной базе данных PostgreSQL, управления пользователями, аутентификации и авторизации. Дополнительно платформа поддерживает Row-Level Security (RLS) — гибкий механизм разграничения доступа на уровне отдельных записей. Это особенно важно при организации индивидуальных учебных кабинетов и истории прохождения тестов. Механизмы подписки на события и потоковой передачи данных в реальном времени позволяют оперативно отслеживать изменения в системе, что даёт возможность построения адаптивной среды обучения.

Ключевой особенностью архитектурного решения является концепция «толстого клиента», при которой значительная часть логики, включая реализацию проверок, алгоритмов и визуального взаимодействия, реализована в браузере пользователя. Такой подход минимизирует количество серверных запросов, ускоряет выполнение операций и упрощает масштабирование системы без необходимости постоянного увеличения серверных ресурсов.

В основу проектирования базы данных положена реляционная модель, представленная шестью ключевыми сущностями:

- roles – содержит перечень ролей (администратор, студент);

- users – информация о пользователях: идентификатор, имя, фамилия, email, пароль и ссылка на роль;
- topics – перечень тем тестов (например, «Алгоритмы на графах», «Раскраска графов»);
- tests – параметры тестов: название, тип (теоретический, практический, смешанный), ограничение по времени;
- questions – формулировка вопроса, тип задания, правильный ответ, структура графа, количество баллов;
- user_test_results – журнал прохождения тестов: идентификатор пользователя, теста, дата, результат, затраченное время.

Все сущности связаны между собой через внешние ключи, обеспечивая целостность и логическую связанность данных.

Хранение графовых структур в приложении осуществляется в формате JSON, что обеспечивает гибкость и совместимость с библиотекой визуализации Cytoscape.js. Такой подход позволяет единообразно описывать как ориентированные, так и неориентированные графы, взвешенные и невзвешенные структуры.

В системе реализованы основные алгоритмы теории графов, необходимые для проверки интерактивных заданий. Они реализованы на клиентской стороне на языке JavaScript, что позволяет мгновенно производить вычисления и проверку ответов без обращения к серверу.

Алгоритм Дейкстры применяется в задачах на поиск кратчайшего пути в ориентированном взвешенном графе. Входными данными выступают списки вершин и ребер, а также начальная и конечная вершины. Алгоритм инициализирует расстояния до всех вершин как бесконечность, кроме начальной, и последовательно обновляет их с учетом весов ребер. В качестве результата возвращается кратчайший путь и его длина, которые сравниваются с действиями пользователя.

Алгоритм Прима используется в заданиях на построение минимального остовного дерева. На каждом шаге из множества уже включённых в остов вершин выбирается ребро минимального веса, соединяющее остов с новой вершиной. Проверка ответа пользователя осуществляется по трем критериям: корректное количество ребер ($n-1$), связность графа, отсутствие циклов и минимальная сумма весов.

Реализованы жадные алгоритмы для:

- вершинной раскраски — при которой смежные вершины не могут иметь один цвет;
- реберной раскраски — где учитывается смежность ребер по общей вершине.

Проверка заключается в определении допустимости раскраски и в оценке количества использованных цветов.

Проектирование веб-приложения для контроля знаний по теории графов включает в себя комплексное решение задач архитектурного, логического и визуального уровня. Применение современных web-технологий, таких как React и Supabase, позволило создать масштабируемую и функциональную систему, в которой сочетаются визуализация, интерактивность и алгоритмическая проверка.

Спроектированная архитектура, структура базы данных и унифицированный формат хранения графов обеспечивают расширяемость системы, а реализация алгоритмов на клиентской стороне позволяет проводить контроль знаний в реальном времени. Предложенное решение может быть интегрировано в образовательные платформы вузов, а также использовано в рамках дистанционного обучения и самоподготовки студентов технических направлений.

Список литературы:

1. Введение в JavaScript // Современный учебник JavaScript: [сайт]. – URL:

<https://learn.javascript.ru/intro> (дата обращения: 17.06.2025).

2. Что такое React и как его освоить? // Яндекс Образование: [сайт]– URL: <https://education.yandex.ru/journal/cto-takoe-react-i-kak-ego-osvoit> (дата обращения: 17.06.2025).

3. Single Page Application: что это такое и как работает // Журнал «Код»: [сайт]. – URL: <https://thecode.media/spa/> (дата обращения: 17.06.2025).

4. Cytoscape.js // Cytoscape: [сайт]. – URL: <https://js.cytoscape.org/#introduction/about> (дата обращения: 17.06.2025).

5. Supabase | The Open Source Firebase Alternative // Supabase: [сайт]. – URL: <https://supabase.com/> (дата обращения: 17.06.2025).