

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**СОЗДАНИЕ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ КУРСА ТЕОРИИ ГРАФОВ
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студента 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Бровко Романа Александровича

Научный руководитель
доцент, к. ф.-м. н.

Ю. Н. Кондратова

Заведующий кафедрой
доцент, к. ф.-м. н.

С. В. Миронов

ВВЕДЕНИЕ

Актуальность темы. Эффективность образовательного процесса сильно зависит от удобства инструментов, которые в нем используются. Сейчас в Саратовском государственном университете им. Н. Г. Чернышевского (СГУ) для большинства дисциплин, связанных с программированием, используется сайт course.sgu.ru, основанный на обучающей платформе Moodle. Для разных дисциплин на этой платформе можно разместить лекции, тесты и задания.

С одной стороны, это удобно тем, что за всеми заданиями можно следить в одном месте.

С другой стороны, такое единообразие не дает возможности приспособить обучающую платформу под специфику конкретного предмета. Из-за этого студентам может быть сложнее понимать материал. Поэтому для некоторых предметов логичнее было бы использовать специализированные сайты, на которых будут учтены особенности конкретной дисциплины.

В качестве примера можно взять курс по теории графов. Чтобы студенты лучше понимали этот предмет, было бы полезно создать приложение, визуализирующее графовые алгоритмы, позволяющее создавать свои графы, проводить эксперименты, видеть наглядные примеры выполнения теорем и придумывать свои алгоритмы. Для удобства такое приложение должно быть объединено с платформой для сдачи заданий по программированию на проверку преподавателю. Таким образом, в одном приложении должны быть реализованы и функции платформы СГУ для обучения (course.sgu.ru), и новые возможности, связанные со спецификой конкретного предмета.

Цель бакалаврской работы — разработка веб-приложения со следующими функциями:

1. Студент и преподаватель могут авторизоваться в системе.
2. После авторизации студенту доступны материалы курса: лекции, тесты, задания по программированию.
3. Студент может отправить задание на проверку преподавателю.
4. Любому пользователю доступен редактор графов, где он может создавать свои графы, редактировать их, а также запускать алгоритмы, изучаемые в семестре.
5. Должны быть доступны разные типы графов: ориентированные и неориентированные, взвешенные и невзвешенные.

6. Преподаватель может добавлять в систему других преподавателей, проверять задания, выставять баллы, редактировать лекции, тесты, задания.

Поставленная цель определила **следующие задачи**:

1. Изучить технологии, необходимые для создания веб-приложений: Java Spring Boot, HTML, CSS, JavaScript, React, SQL, протокол HTTP.
2. Описать функции, которые будет выполнять приложение, понять, как должен выглядеть интерфейс.
3. Реализовать приложение, сравнить его с текущими решениями.

Методологические основы веб-разработки представлены в работах Д. Р. Чиганова, В. И. Суханова, Д. С. Цыганковой, Т. В. Зудиловой и М. Л. Бурковой, Д. Макфарланда, А. А. Горбачева и Е. С. Горбачевой.

Практическая значимость бакалаврской работы. Разработанный курс может быть полезен для студентов, изучающих теорию графов в университете. Специализированный курс теории графов с визуализацией алгоритмов может помочь студентам быстрее освоить данную дисциплину.

Структура и объём работы. Бакалаврская работа состоит из введения, 3 разделов, заключения, списка использованных источников и 2 приложений. Общий объём работы — 59 страниц, из них 42 страницы — основное содержание, включая 10 рисунков и 0 таблиц, цифровой носитель в качестве приложения, список использованных источников информации — 28 наименований.

1 Краткое содержание работы

Первый раздел «Обзор существующих образовательных ресурсов» посвящен исследованию аналогов разрабатываемого приложения и альтернативных способов изучения теории графов. Рассмотрены следующие образовательные ресурсы:

1. Книги и сайты с теоретическими материалами. Преимущества: могут содержать много теоретического материала, большое разнообразие. Недостатки: отсутствие закрепления знаний на практике, потому что для проверки практических заданий необходимо либо участие преподавателя, либо автоматическая проверка кода.
2. Платформы для решения задач по программированию (Codeforces, Leetcode). Преимущества: много заданий с автоматической проверкой. Недостатки: меньше теории, нет тестов для проверки теоретических знаний, нет визуализации алгоритмов.
3. Платформы, основанные на Moodle (например, course.sgu.ru). Преимущества: есть теория, задания и тесты. Недостатки: не учтена специфика конкретной дисциплины (нет визуализации алгоритмов).
4. Специализированные сайты с визуализацией графов (Graph Editor, Graph Online). Преимущества: эти инструменты хорошо помогают в наглядном изучении теории графов. Недостатки: меньшая гибкость по сравнению с Moodle (нельзя редактировать материалы курса), не хватает некоторых других возможностей, например, тестов.

В результате обзора сделан вывод, что было бы полезно объединить преимущества всех рассмотренных вариантов (лекции, задачи, тесты, визуализация алгоритмов, возможность редактировать материалы курса).

Второй раздел «Используемые технологии» посвящен описанию необходимых технологий для создания веб-приложений, а также конкретных технологий, выбранных в данной работе. Раздел состоит из 3 частей:

1. Архитектура и принцип работы веб-приложений. Этот подраздел описывает основные особенности и применения веб-приложений, а также основные компоненты их архитектуры: презентационный слой (отвечающий за взаимодействие с пользователем и отображение данных), логический слой (обрабатывающий запросы, полученные от презентационного слоя, а также выполняющий бизнес-правила и взаимодействующий с базой данных)

и слой доступа к данным (может содержать реляционные и нереляционные базы данных; они необходимы для хранения информации, которая должна сохраняться на сайте при повторном посещении).

2. Общее описание технологий, необходимых для создания веб-приложения. В этом подразделе описываются основные языки программирования и фреймворки, которые нужны для создания разных слоев приложения. Также указаны технологии, используемые в данной работе, перечислены их некоторые преимущества.
3. Подробное описание технологий для создания веб-приложений, используемых в данной работе. В этом подразделе описывается каждая технология (язык программирования, фреймворк, протокол), используемые в созданном веб-приложении. Подробно описаны преимущества технологий, их назначение и основные синтаксические конструкции. В работе использованы следующие технологии: язык Java с фреймворком Spring Boot (для создания серверной стороны приложения), языки HTML, CSS и JavaScript с библиотекой React для создания клиентской стороны приложения, протокол HTTP для обмена сообщениями между клиентской и серверной частями приложения, SQL для взаимодействия с базой данных.

В результате данного раздела был описан набор используемых технологий и обосновано их использование.

Третий раздел «Реализация веб-приложения» посвящен описанию реализованного приложения для курса теории графов, его структуры и интерфейса, а также подготовленных для курса материалов: лекций, задач и тестов.

Раздел состоит из 3 частей:

1. Структура проекта. Этот подраздел описывает основные реализованные классы и функции приложения. Структура проекта разделена на серверную часть (backend), написанную на Java Spring Boot, и клиентскую часть (frontend), написанную на React.

На стороне сервера реализованы функции, связанные с обращением к базе данных, работой с JWT-токенами для аутентификации и вычислением баллов за тесты. Серверная часть проекта включает в себя следующие файлы:

- SecurityConfig — файл с конфигурационным классом для Spring Security. Нужен для настройки таких параметров аутентификации, как способ

хэширования пароля.

- `JwtTokenUtil` — класс с функциями для работы с JWT-токенами, необходимыми для аутентификации пользователя. Когда пользователь регистрируется или входит в аккаунт, ему отправляют с сервера JWT-токен, который содержит полезную информацию (например, идентификатор пользователя и его роль) и подпись, гарантирующую, что пользователь авторизован. Когда пользователь вносит изменения на сайт, сервер проверяет эту подпись, чтобы никто другой не мог выдать себя за данного пользователя.
- Контроллеры. Это классы для взаимодействия с клиентом, ответа на его запросы по протоколу HTTP. В проекте есть контроллеры для обработки запросов, связанных с лекциями, просмотром баллов за курс, тематическими разделами курса, задачами, тестами, пользователями.
- `CorsFilter` — класс, необходимый для корректной работы запросов с одного порта на другой (cross-origin resource sharing).
- DTO (Data Transfer Objects) — объекты, которые передаются по протоколу HTTP от клиента на сервер или наоборот. В проекте используются DTO для передачи информации о файле с решением задачи, о варианте ответа на вопрос теста, о добавлении вопроса в тест, о выставлении баллов за задачу, об общем количестве баллов у студента, об ответе студента на вопрос теста и на весь тест, о баллах студента за тест и т.д.
- Сущности (Entity) — объекты, описывающие таблицы в базе данных. В проекте используются сущности для описания лекций, текстовых блоков и изображений в лекциях, тематических разделов курса, тестов, вопросов теста, ответов студента и его баллов за тест, задач, вариантов задачи и их присвоений студентам, попыток решения задачи студентом и файлов в них, а также пользователей курса.
- Репозитории — классы, описывающие взаимодействие с базой данных (SQL-запросы). В проекте есть репозитории, соответствующие каждой сущности.
- Модели — классы, которые с которыми работает серверная часть проекта, но которые не связаны с хранением в базе данных или от-

правкой по протоколу HTTP. В проекте используются модели для работы с пользователями и с баллами студента по задаче.

- Сервисы — классы, в которых реализована основная логика приложения, не связанная напрямую с обращением к базе данных или взаимодействием с клиентом. В проекте есть один сервис — `UserService`, в котором обрабатываются операции кодирования пароля перед добавлением в базу данных и получения пользователя по его имени с обработкой случая, когда такого пользователя нет.
- `GraphsApplication` — класс с функцией для запуска серверной части приложения.

Клиентская часть проекта реализована на React с использованием упаковщика модулей Webpack. Данная часть проекта состоит из следующих файлов:

- а) `app.jsx` — основной файл, в котором описана структура приложения. Компонент `App` вызывает отображение других компонентов в зависимости от того, какую страницу сейчас просматривает пользователь.
- б) Папка `components` содержит компоненты приложения, каждый из которых отвечает за отображение и обработку отдельной части веб-страницы. В проекте есть следующие компоненты:
 - `Header` — шапка сайта, в которой есть ссылки на разные разделы сайта.
 - `Home` — главная страница, на которой доступны материалы курса: лекции, тесты и задачи.
 - `Login` — страница входа в аккаунт
 - `Lecture` — страница с лекцией.
 - `Task` — страница с задачей.
 - `Test` — страница теста.
 - `UserPage` — страница с данными о пользователе.
 - `Visualization` — страница с редактором графа.
- в) Папка `css` содержит файл `style.css` для оформления дизайна сайта.
- г) Папка `graphAlgorithms` содержит функции для визуализации алгоритмов. Эти функции возвращают последовательность операций, которые нужно применить к графу для корректной визуализации запуска алгоритма. Эти операции включают в себя перекрашивание

вершин и ребер, временную задержку, изменение информации о вершине (например, кратчайшего пути до нее) и т.д.

д) Папка `utils` содержит функции, которые не связаны с конкретным компонентом и могут использоваться в любом из них. Эта папка содержит следующие файлы:

- `deleteCookie.js` — файл с функцией удаления пары «ключ-значение» из `cookie`-файла документа.
- `getCookie.js` — файл с функцией поиска значения по ключу в `cookie`-файле.
- `getRandomInt.js` — файл с генератором случайного целого числа в полуинтервале от 0 до заданного числа.
- `toBase64.js` — перевод содержимого файла в формат `base64`.

е) Файл `index.html` содержит подключение `CSS`-файла и файла `bundle.js`, созданного упаковщиком модулей `Webpack`.

2. Описание работы приложения. Этот подраздел описывает функционал разработанного приложения.

При первом посещении сайта пользователь попадает на страницу с визуализацией графов. На ней он может редактировать граф с помощью текстового поля, в котором записаны ребра графа. Приложение поддерживает разные типы графов: ориентированные и неориентированные, взвешенные и невзвешенные. Также пользователь может выбрать один из 5 доступных алгоритмов (обход в глубину, обход в ширину, алгоритм Дейкстры, Форда-Беллмана или Краскала) и запустить его визуализацию на построенном графе.

Также пользователь может войти в аккаунт или зарегистрироваться (перейдя по ссылкам в правом верхнем углу). Для этого он должен ввести имя пользователя и пароль (и подтверждение пароля для регистрации). Изначально пользователю дают роль студента. Чтобы получить права преподавателя, их должен предоставить данному пользователю другой преподаватель.

После входа в аккаунт пользователь переходит на главную страницу. На ней студент может просматривать материалы курса. Курс разделен на темы, которые преподаватель может добавлять на главной странице. Каждая тема состоит из ссылок на лекции, задачи и тесты. При нажатии на ссылку

происходит переход к данному материалу курса.

Преподавателю также доступны функции добавления и удаления материалов. При нажатии на кнопки «Добавить лекцию», «Добавить задачу», «Добавить тест» появляются текстовые поля, которые преподаватель должен заполнить для успешного добавления.

При переходе к конкретной лекции пользователь может просматривать её содержимое. Лекции состоят из текста и изображений. В тексте поддерживаются некоторые функции форматирования, такие как выделение жирным или подсветка кода. Студент может только читать лекцию, а преподаватель может создавать и удалять блоки (текстовые и с изображениями), а также редактировать текст.

На странице задачи преподаватель может видеть условия всех ее вариантов и добавлять новые варианты. Студенту при открытии ссылки на задачу присваивается случайный вариант, и он видит номер варианта и его условие. Студент может отправлять попытку решения задачи, которую будет видеть преподаватель и сможет поставить баллы. Попытка состоит из файлов, которые можно добавлять и удалять. Если добавить файлы после проверки последней попытки, то создается новая попытка.

На странице теста преподаватель может добавлять вопросы 3 типов: с текстовым ответом, с выбором одного варианта ответа и с выбором нескольких вариантов. После этого преподаватель может опубликовать тест, чтобы он стал виден студентам. После прохождения теста студенту автоматически выставляются баллы и показываются правильные ответы. В отличие от задач, в тестах доступна только одна попытка.

Также в приложении есть страница пользователя. На этой странице пользователь может видеть свою роль (студент или преподаватель). Студенту на этой странице доступны его общие баллы за курс (сумма максимальных оценок за попытку по каждой задаче и баллов за все тесты) и максимальное количество баллов, которые теоретически можно заработать за курс. Преподавателю доступны баллы всех студентов, которые отправляли решения задач или тестов на курс. Он может использовать эти баллы для выставления оценок за семестр. Также на этой странице преподаватель может сделать других пользователей преподавателями.

3. Подготовленные материалы курса. Для проекта были подготовлены лек-

ции, задачи и тесты на следующие темы: понятие графа (основные определения, частные случаи и способы представления графа), обход в глубину, обход в ширину, алгоритмы Дейкстры, Форда-Беллмана и Краскала. Всего подготовлено 6 лекций, 10 задач (в каждой от 1 до 6 вариантов) и 2 теста. Все материалы курса находятся на flash-носителе в приложении к работе.

В результате было разработано приложение для курса теории графов с визуализацией алгоритмов, описана его структура и функционал, а также созданы материалы курса: лекции, задачи и тесты.

ЗАКЛЮЧЕНИЕ

В работе были проанализированы существующие образовательные ресурсы, рассмотрен стек технологий для реализации веб-приложений. Было создано веб-приложение для курса теории графов. В приложении имеется возможность размещать материалы лекций, задания по курсу и тесты для проверки знаний студентов, реализован механизм для визуализации алгоритмов на графах. Реализована регистрация и идентификация при входе в курс. Права доступа разграничены для преподавателя и студента. Размещенный в данный момент на сайте материал может быть изменен и дополнен преподавателем в зависимости от потребностей. По мере прохождения курса студенты могут выкладывать на сайт результаты выполненных заданий, преподаватель же имеет возможность оценивать эти задания, выставя баллы. Таким образом, преподаватель может оценить результаты прохождения курса студентом по выполненным заданиям и тестам.

Преимущество приложения по сравнению с существующими аналогами — наличие визуализации графов с возможностью запускать на них алгоритмы, изучаемые в курсе. В настоящий момент в приложении реализована визуализация пяти алгоритмов с возможностью ввода и редактирования графа, редактирования его типа (ориентированный/неориентированный, взвешенный/невзвешенный).

В дальнейшем планируется расширение возможностей приложения:

- расширение функционала, например, средств общения студентов с преподавателем;
- улучшение редактора графов, например, генерация позиции вершин более оптимальным способом;
- увеличение количества алгоритмов с визуализацией.

Основные источники информации:

1. *Чиганов, Д. Р.* Spring: мощный фреймворк для разработки Java-приложений / Д. Р. Чиганов // Вестник науки. — 2023. — Т. 4, № 7. — С. 274.
2. *Суханов, В. И.* Разработка веб-приложений на платформе Spring: учебно-методическое пособие / В. И. Суханов. — Екатеринбург: Изд-во Урал. ун-та, 2023.
3. *Цыганкова, Д. С.* Основы HTML: изучаем основной язык веб-разработки / Д. С. Цыганкова // Форум молодых ученых. — 2023. — № 10(86). — С.190.
4. *Зудилова, Т. В.* Web-программирование HTML / Т. В. Зудилова, М. Л. Буркова. — СПб.: НИУ ИТМО, 2012.
5. *Макфарланд, Д.* Новая большая книга CSS / Д. Макфарланд. — СПб: Питер, 2016.
6. *Горбачев, А. А.* Сравнение классического процесса реализации веб-приложений и подхода с использованием библиотеки React / А. А. Горбачев, Е. С. Горбачева // Молодой исследователь Дона. — 2020. — № 1(22). — С. 29.