

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

Кафедра математического обеспечения математических комплексов и
информационных систем

Интерактивный учебник по теории графов на ASP.NET Core

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование
информационных систем

факультета компьютерных наук и информационных технологий

Бугаева Егора Александровича

Научный руководитель:

Профессор,

д-р физ.-мат. наук, профессор

Андрейченко Д.К.

подпись, дата

Зав. кафедрой

математического обеспечения

вычислительных комплексов

и информационных систем

Андрейченко Д.К.

подпись, дата

Саратов 2025

ВВЕДЕНИЕ

Актуальность темы

Ввиду активного развития информационных технологий и повсеместного распространения Интернета, а также из-за наличия почти у каждого человека устройства для взаимодействия с сетевым пространством, актуален вопрос качественного получения информации.

Рассматривая сферу обучения и образования и стоящие перед ней задачи, очевидна необходимость современного подхода к обучению, в частности использования информационных технологий в образовании. Для школьников и студентов качественно структурированный и оформленный электронный учебник, представляющий собой интерактивный справочник по интересующей дисциплине, может существенно упростить получение знаний.

Цель бакалаврской работы — разработка WEB-приложения-учебника по теории графов с интерактивными элементами.

Поставленная цель определила **следующие задачи**:

1. Изучить предметную область, рассмотреть существующие аналоги по указанной тематике.
2. Разработать архитектуру и сформулировать требования к функционалу собственного веб-приложения.
3. Рассмотреть и выбрать необходимые для реализации средства и технологии.
4. Разработать приложение и убедиться в работоспособности функционала.

Методологические основы разработки учебных веб-приложений представлены в работах Гарькуши Н.В., Буториной Н.И., Кувшинова Н.Е., Галяутдинова А.А.

Теоретическая и/или практическая значимость бакалаврской работы

Практическая значимость работы заключается в возможности внедрения разработанного решения в образовательный процесс. Приложение превосходит аналоги по функциональным возможностям и удобству интерфейса.

Структура и объём работы. Бакалаврская работа состоит из введения, 6 разделов, заключения, списка использованных источников и 3 приложений. Общий объём работы — 78 страниц, из них 50 страниц — основное содержание, включая 10 рисунков, список использованных источников информации — 23 наименования.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Образовательный веб-сайт» посвящен исследованию структуры учебных приложений с веб-интерфейсом, их типичного внешнего вида и функциональных возможностей. Также в настоящем разделе рассматриваются существующие аналоги по тематике работы. Дополнительно приведена классификация подобных решений в зависимости от направленности.

В качестве вывода указываются преимущества и недостатки аналогичных решений и предлагаются возможности применения полученных данных при разработке. Ключевым недостатком существующих решений является полное или практически полное отсутствие элементов интерактивного взаимодействия с пользователем.

Второй раздел «Задачи создаваемого веб-приложения» посвящен чёткому определению задач, решаемых приложением. В разделе приведён макет внешнего вида страниц, определены требования к форматированию и вёрстке. Требования сформулированы с учётом замечаний, выявленных в предыдущем разделе при рассмотрении аналогичных решений по тематике «теория графов».

Третий раздел «Инструменты и технические решения для реализации» посвящён рассмотрению и выбору необходимых средств разработки. В контексте веб-приложения необходимы инструменты как для создания клиентской, так и серверной части.

Для клиентской составляющей рассматриваются такие технологии как HTML (язык разметки гипертекста, служит для определения элементов веб-страницы, её содержимого и порядка отображения), CSS (каскадные таблицы стилей, применяются для модификации внешнего вида страниц, поддерживают структурирование страниц по блокам, ячейкам, строкам и т.д.), JavaScript (используется для написания скриптов — программ на языке JavaScript, служащих для обработки пользовательского ввода и клиентской логики).

Среди инструментов разработки серверной части рассмотрены ASP.NET Core (фреймворк, платформа для реализации внутреннего функционала приложения, поддерживающая множество возможностей по конфигурированию и разработке), Entity Framework Core (инструмент для выполнения манипуляций с базами данных, тесно интегрированный с ASP.NET Core), а также PostgreSQL (система управления базами данных, предоставляющая функции хранения, обработки и изменения накопленной информации).

Дополнительно рассмотрены вспомогательные технологии, такие как:

- Razor Pages в составе ASP.NET Core (формат веб-страниц, расширяющих функции стандартного HTML путём возможности внедрения в код конструкций на языке C#);
- Bootstrap (расширение для CSS, позволяющее создавать адаптивные интерфейсы под различные форматы и разрешения экранов, упрощающее стилизацию элементов страниц);
- Docker (система контейнеризации, упрощающая упаковку и развёртывание приложений).

Четвёртый раздел «Подготовка проекта» посвящён первоначальному конфигурированию приложения в рамках выбранной архитектуры (в качестве таковой выбран классический подход MVC, предполагающий разбиение компонентов приложения на три функциональных элемента — Model (модель — описывает используемые данные, а также логику их обработки), View (представление — отвечает за визуальную составляющую пользовательского интерфейса), Controller (контроллер — центральный компонент MVC, обеспечивает связь между пользователем и серверной логикой, обрабатывает данные и управляет их промежуточным хранением).

В разделе подробно описан процесс создания проекта в среде разработки Visual Studio, настройка конфигурационных файлов приложения,

включая настройку маршрутизации, параметров запуска, подключения к базе данных, контейнеризации.

Для приложения, базы данных и контейнера приведены фрагменты программного кода, отвечающие за ключевые параметры, необходимые для синхронизации их работы, обеспечения внешнего доступа к интерфейсу приложения.

Пятый раздел «Настройка функционала серверной части» посвящён настройке межсервисного взаимодействия в рамках приложения, созданию полного функционала серверной составляющей. Описан процесс разработки моделей данных, создание контекста базы данных и его использование для подключения, рассмотрен подход DTO (Data Transfer Object — объект передачи данных) для промежуточного хранения информации. Также приведён процесс создания и применения миграций базы данных (миграция — механизм управления базой данных посредством программного кода).

Разработаны и описаны контроллеры, которые определяют функции сервера в части перенаправления пользователя на необходимые страницы, регистрации и авторизации, записи необходимых при выполнении некоторых действий данных в базу. Дополнительно рассмотрен механизм хеширования — одностороннего шифрования пароля, обеспечивающий безопасность его хранения.

Шестой раздел «Настройка функционала клиентской части» посвящён процессу разработки компонентов, осуществляющих непосредственное взаимодействие с пользователем, отображение интерфейса и отправку или получение данных от сервера.

Рассматривается реализация представлений Razor Pages, возможности использования шаблонов страниц. Приведено описание разработки ключевых страниц приложения — регистрации и авторизации, учебных статей, автоматических тестов, интерактивной визуализации графа.

Особенно подробно описан процесс разработки центральной функции приложения — визуализатора графа, поддерживающего возможности по построению визуальной модели графа, манипулирования ей и выполнения алгоритмов. Для реализации этой функции были применены JavaScript-библиотеки Graphology и Sigma.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы была рассмотрена предметная область, существующие варианты сайтов-учебников, посвящённых теории графов, проанализировано их наполнение и функционал, в результате чего были определены их преимущества и недостатки. Ключевым недостатком аналогичных решений является полное или почти полное отсутствие взаимодействия с пользователем. В разработанном приложении эта проблема была устранена.

Были рассмотрены необходимые инструменты и технологии разработки, такие как C#, ASP.NET Core (и технология Razor Pages, в частности), JavaScript и фреймворк Bootstrap, PostgreSQL, с использованием которых в соответствии с современными требованиями было разработано полнофункциональное приложение с веб-интерфейсом: учебник по теории графов, превосходящий аналоги по числу технических возможностей.

В рамках работы были реализованы следующие функции приложения: хранение и отображение теоретического материала, возможность автоматизированной проверки знаний посредством тестов, интерактивное средство для отображения визуального представления графа с широким набором возможностей для его создания, модификации и исполнения алгоритмов.

Таким образом, были выполнены все поставленные задачи и достигнута цель работы.

Основные источники информации:

1. Гарькуша Н.В., Буторина Н.И., Особенности организации образовательного сайта // Наука и перспективы. – 2016. – №4. – с. 32–39
2. Кувшинов Н.Е., Галяутдинов А.А., Применение HTML5 при создании электронных учебников // Форум молодых учёных. – 2017. – №9 (13). – с. 444–447
3. Freeman E., Robson E. Head First HTML5 Programming: Building Web Apps with JavaScript 1-е изд.: O'Reilly Media, 2011. – 610 с. – Яз. англ.
4. Ефремов В.М. Разработка веб-сайта с использованием Bootstrap // Инновационная наука. – 2020. – №6. – с.40-42
5. Гюллинг А.О., Воронцова Н.В., Визуализация алгоритма Дейкстры // Известия Тульского государственного университета. Технические науки. – 2024. – №2. – с. 127–132