МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Алтайский государственный университет»

Колледж АлтГУ

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**(дипломная работа)**

Тема: Разработка сайта магазина мягких игрушек

с подключением базы данных MySQL

Выпускную квалификационную работу выполнил(а) студентка 3 курса,

группы к002в \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
  
 (подпись)  
  
Научный руководитель:  
Жданов Денис Николаевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
  
 (подпись)   
  
Выпускная квалификационная работа защищена:   
«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.   
  
Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
  
Председатель ГЭК:  
Печатнов Владимир Владимирович \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (подпись)

Барнаул 2023 г.

**СОДЕРЖАНИЕ**

# **ВВЕДЕНИЕ**

Современная эпоха цифровых технологий вынуждает нас активно взаимодействовать с виртуальным миром, и важной частью этого взаимодействия становится веб-пространство. Создание эффективных, мощных и интуитивно понятных веб-сайтов требует глубокого понимания современных технологий и инструментов. В рамках данной выпускной квалификационной работы представляется исследование различных технологий, применяемых при разработке веб-сайтов, с фокусом на использовании ведущих фреймворков и технологий, таких как Svelte, React, HTMX, Node.js, Bun, и PHP.

Технологический ландшафт веб-разработки постоянно эволюционирует, и понимание новейших фреймворков и языков программирования становится ключевым аспектом успешной реализации проектов. В этой работе мы глубоко исследуем фреймворки Svelte, React и HTMX, а также технологии Node.js, Bun и PHP, анализируя их возможности, преимущества и сферы применения.

Svelte его инновационным подходом к компиляции компонентов во время сборки, обещает более эффективное использование ресурсов и быстрое реагирование на изменения в коде.

Работа с React, мощным библиотекой компонентов, стала стандартом индустрии, предоставляя разработчикам гибкие инструменты для построения современных пользовательских интерфейсов.

Технология HTMX, основанная на идее использования HTML, CSS и JavaScript для создания динамичных веб-приложений, приобретает все большую популярность благодаря своей простоте и легкости внедрения.

Вместе с этим, серверные технологии Node.js, Bun и PHP предоставляют надежные среды выполнения кода, позволяя создавать масштабируемые и высокопроизводительные веб-приложения.

Данная исследовательская работа ставит перед собой целью не только анализировать отдельные технологии, но и рассматривать их в контексте совместного использования, выявляя потенциал для создания более эффективных и инновационных веб-приложений. Вместе с этим, применение разнообразных фреймворков и языков программирования демонстрирует гибкость и многообразие подходов к веб-разработке.

Основываясь на проведенном анализе, данная работа предоставит основательное понимание технологического ландшафта веб-разработки, а также рекомендации для выбора оптимальных инструментов в зависимости от поставленных задач. В современном мире веб-разработки, выбор правильных инструментов становится критическим для достижения успеха проекта.

Сравнительный анализ фреймворков Svelte, React позволит выявить их сильные стороны и особенности, что станет основой для эффективного использования в конкретных сценариях разработки. Обсуждение технологии HTMX позволит проанализировать подход, который делает веб-приложения более отзывчивыми, минимизируя при этом сложность кода на стороне клиента.

Погружение в мир серверных технологий с Node.js, Bun и PHP раскроет аспекты создания масштабируемых и эффективных серверных приложений. Особое внимание уделяется экосистеме Node.js, предоставляющей широкие возможности для построения высокопроизводительных веб-приложений на стороне сервера.

В контексте работы также рассматриваются практические аспекты применения этих технологий через создание прототипов веб-приложений. Реализация примеров с использованием различных фреймворков и технологий позволит получить практический опыт и оценить применимость рассматриваемых инструментов в реальных проектах. Все эти аспекты обеспечат полное понимание технологического ландшафта веб-разработки, начиная от клиентской стороны с использованием современных фреймворков, заканчивая серверной стороной с использованием мощных технологий, способных обеспечить высокую производительность и отзывчивость веб-приложений.

Целью данной выпускной квалификационной работы является предоставление комплексного обзора исследованных технологий, их взаимодействия и практического применения. Полученные результаты позволят разработчикам и IT-специалистам принимать обоснованные решения при выборе технологического стека для своих проектов, с учетом специфики задач и требований.

**1 ОСНОВЫ ПОСТРОЕНИЯ ВЕБ-САЙТА**

* 1. **История создания веб-страниц.**

История создания веб-страниц тесно связана с эволюцией интернета и стремительным развитием технологий. Рассмотрим этот процесс в двух ключевых аспектах: начале веб-разработки и последующих инновациях, которые сформировали современный ландшафт веб-страниц.

Начало веб-разработки: Появление HTML В начале 1990-х годов, когда интернет только начал становиться доступным для широкой аудитории, веб-страницы были простыми статическими документами. Термин "веб-страница" в то время описывал простой файл HTML (HyperText Markup Language), который содержал текст, ссылки и изображения. Создание веб-страниц сводилось к написанию кода HTML с использованием простых тегов для структурирования контента. С появлением браузеров, таких как Mosaic и Netscape Navigator, пользователи стали иметь возможность просматривать эти веб-страницы. Однако в те времена интерактивности и динамичности было мало.

Инновации в динамике веб-страниц: JavaScript и CSS Следующий важный этап в истории веб-страниц связан с появлением JavaScript и CSS в середине 1990-х. JavaScript добавил возможность создания динамических элементов и взаимодействия с пользователем, тогда как CSS (Cascading Style Sheets) позволил более гибко стилизовать и размещать элементы на странице. Эти инновации позволили разработчикам создавать интерактивные веб-приложения, открыв новые горизонты для пользовательского опыта. Появились различные методы и библиотеки для облегчения разработки, и веб-страницы стали более динамичными и сложными.

Эра динамических веб-приложений и AJAX. В начале 2000-х годов концепция AJAX (Asynchronous JavaScript and XML) стала ключевым фактором, изменяющим ландшафт веб-разработки. AJAX позволяет асинхронно обмениваться данными между браузером и сервером без необходимости полной перезагрузки страницы. Это привело к созданию более отзывчивых и быстрых веб-приложений, подняв уровень интерактивности на новую высоту.

Развитие фреймворков и SPA. С появлением более сложных веб-приложений возникла необходимость в эффективных инструментах. Вторая половина 2000-х и начало 2010-х принесли множество фреймворков, таких как Angular, React и Vue.js, позволяющих создавать одностраничные приложения (SPA) и управлять состоянием интерфейса. SPA позволяют пользователям взаимодействовать с веб-приложением, не перезагружая всю страницу. Это повышает производительность и создает более плавный пользовательский опыт.

Мобильная адаптация и респонсивный веб-дизайн. С развитием мобильных устройств стало ясно, что веб-страницы должны быть адаптированы под различные экраны. Родился концепт респонсивного веб-дизайна, который позволяет страницам автоматически адаптироваться к размеру экрана, обеспечивая удобство использования на различных устройствах.

* 1. **Выбор тематики веб-сайта**

В данной выпуской квалификационной работе была выбранна тема блога для веб-сайта, так как она отлично может показать отличия или схожести тех или иных технологий.

* 1. **Техническое задание на создание web-сайта**

1. **РАЗРАБОТКА САЙТА БЛОГА**

**2.1 Структура сайта**

**ЗАКЛЮЧЕНИЕ**

В заключение, история создания веб-страниц представляет собой захватывающий путь инноваций и технологического прогресса. С течением времени, веб-разработка превратилась из простого использования HTML в сложный и многоуровневый процесс, требующий глубоких знаний и умений. Рассмотрение этой эволюции позволяет нам осознать, какие технологии стали ключевыми в различные периоды и как они формировали пользовательский опыт.

Открывая страницу веб-истории, мы видим, как простые статические документы превратились в динамичные, многогранные веб-приложения благодаря появлению JavaScript и CSS. Идея асинхронного обмена данными через AJAX изменила подход к интерактивности, а фреймворки, такие как Angular, React и Vue.js, подарили нам возможность создавать более сложные SPA.

С мобильным вторжением респонсивный веб-дизайн стал неотъемлемой частью веб-разработки, позволяя веб-страницам мгновенно адаптироваться к различным устройствам и экранам. Эта реакция на растущее разнообразие устройств и экранов отражает не только технологический прогресс, но и стремление к улучшению пользовательского опыта.

Современная веб-разработка олицетворяет собой мозаичный ландшафт, в который входят разнообразные фреймворки, библиотеки и инструменты, о которых упоминается в данной работе. Веб-страницы перестали быть просто информационными документами – они стали динамичными, интерактивными, интуитивно понятными и максимально адаптивными к потребностям пользователей.

Наше погружение в историю создания веб-страниц становится ключом к пониманию текущего состояния веб-разработки и технологий, рассмотренных в данной исследовательской работе. Сегодняшние веб-приложения – это не только технологический триумф, но и воплощение стремления создать максимально удовлетворительный и современный пользовательский опыт. Следуя этим тенденциям, будущее веб-разработки обещает продолжение инноваций и улучшений, делая виртуальное пространство более доступным, эффективным и удовлетворительным для всех пользователей.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**