Geekbrains

**Разработка веб-приложения «VMAS» по учету проведения технического обслуживания транспортных средств и контроля исполнения организационных работ для ГБУ МО «Автохозяйство»**

Программа: Разработчик

Специализация: Программист

Зуев Максим Михайлович

Москва

2025

**Содержание**

Введение ............................................................................................................................................ 4

1. Анализ технологий и инструментов создания веб приложения............................................. 6

1.1 СПЕЦИФИКА ЦИФРОВОГО ПРОДУКТА – ИНТЕРНЕТ-МАГАЗИН ............... 6

1.2

ОБЗОР ТЕХНОЛОГИЙ РЕАЛИЗАЦИИ СОВРЕМЕННЫХ ВЕБ-ПРИЛОЖЕНИЙ . 9

1.3

ВЫПИСКА ИЗ ТЕХНИЧЕСКОГО ЗАДАНИЯ НА РАЗРАБОТКУ ВЕБ-

ПРИЛОЖЕНИЯ КНИЖНОГО ИНТЕРНЕТ-МАГАЗИНА .................................................. 16

ГЛАВА 2

РЕАЛИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ ИНТЕРНЕТ-

МАГАЗИНА СРЕДСТВАМИ ФРЕЙМВОРКА ANGULAR ........................... 21

2.1

ДИЗАЙН-МАКЕТ КНИЖНОГО ИНТЕРНЕТ-МАГАЗИНА ........................... 21

2.2

РЕАЛИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ НА ФРЕЙМВОРКЕ ANGULAR ............. 30

2.3

АПРОБАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ КНИЖНОГО ИНТЕРНЕТ-МАГАЗИНА .... 48

ЗАКЛЮЧЕНИЕ .................................................................................................... 52

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ ...................................... 54

**Введение**

Сегодня сложно представить динамичную, быстро развивающуюся компанию без использования IT – продуктов, так как информационные технологии (ИТ) участвуют практически во всех сферах бизнеса, от оптимизации процессов до прогнозирования.

Основной задачей информатизации компании является повышение эффективности управления ею, а также повышение производительности Труда и снижение издержек, увеличение капитализации компании и повышение конкурентоспособности продукции и услуг.

И одним из ключевых, направлений является веб-приложения. В настоящее время использование веб-приложений вышло на новый уровень и не умолило продолжает развиваться. Основными преимуществами веб-приложений стали: интерактивность, доступность, масштабируемость, обновляемость, безопасность.

Также стоит отметить появление и использование более продвинутого прогрессивного веб-приложения (англ. progressive web app, PWA) — технология в веб-разработке, которая визуально и функционально трансформирует сайт в приложение (мобильное приложение в браузере). PWA является гибридным решением и позволяет открыть приложение с помощью мобильного браузера. При этом полностью сохраняется функциональность нативного приложения.

Основной целью данной дипломной работы является (Разработка веб-приложения «VMAS» по учету проведения технического обслуживания транспортных средств и контроля исполнения организационных работ для ГБУ МО «Автохозяйство»). «VMAS» это сокращенная аббревиатура «Vehicle maintenance accounting system»- дословно(Система учета технического обслуживания транспортных средств).

На сегодняшний день на рынке ИТ можно встретить огромное множество готовых продуктов в сфере автомобильной индустрии. Все эти продукты по-своему хороши, но нарешают конкретных целей и задач конкретной компании.

Для достижения данной цели был поставлен ряд теоретических и практических задач:

* Проанализировать технологии реализации веб-приложений;
* Составить техническое задание на разработку с учетом пожеланий компании и поставленных задач;
* Создать дизайн-макет веб-приложения;
* Выполнить разработку веб-приложения и произвести его тестирование;

Таким образом, данная работа состоит из теоретической и практической части.

Теоретико-методологическую базу проекта составляют учебные пособия,

справочники, ресурсы в сети Интернет.

Практическим результатом настоящей работы является готовое веб-приложение с полностью рабочим функционалом.

**Теоретическая и практические главы**

**Анализ технологий и инструментов создания веб приложения.**

Основные технологии, которые используются в этом виде разработки – HTML, CSS, JavaScript. Для создания динамических и интерактивных веб-приложений применяются фреймворки и библиотеки, например, React. js, Angular или Vue. Js

Исследование охватывает различные инструменты и окружения, используемые для разработки веб-приложений, с целью выявить их преимущества, недостатки и области оптимального применения. В работе рассматриваются технологии, языки программирования, интеграционные возможности и уровень поддержки стандартов в каждом из рассматриваемых окружений. Результаты анализа помогут разработчикам и бизнес-аналитикам принимать обоснованные решения при выборе средств разработки веб-приложений в соответствии с требованиями проекта.

В современном мире веб-приложения стали неотъемлемой частью бизнеса и повседневной жизни. Существует множество средств для создания веб-приложений, каждое из которых обладает своими особенностями, преимуществами и недостатками. Этот сравнительный анализ посвящен изучению различных сред создания веб-приложений с целью выявления оптимального инструмента для успешного запуска проекта.

1. Определение целей проекта

Первый шаг при выборе среды для создания веб-приложения – четкое определение целей проекта. Некоторые среды могут быть более подходящими для крупных корпоративных приложений, в то время как другие могут быть оптимальными для небольших стартапов. Решение о выборе среды должно соответствовать требованиям конкретного проекта.

2. Технологии веб-разработки

Одним из ключевых аспектов сравнительного анализа являются используемые технологии веб-разработки. Рассмотрим популярные языки программирования, фреймворки и инструменты, доступные в каждой из рассматриваемых сред. Например, выбор между JavaScript, Python, Ruby или Java может существенно влиять на эффективность разработки.

3. Интеграционные возможности

Способность среды взаимодействовать с другими системами и сервисами также крайне важна. Интеграционные возможности могут существенно упростить разработку и внедрение веб-приложения. Рассмотрение совместимости с базами данных, сторонними API и облачными сервисами помогает предвидеть будущие трудности и избежать их.

4. Уровень поддержки стандартов

Соответствие стандартам веб-разработки обеспечивает стабильность и долгосрочную поддержку созданного приложения. Рассмотрим, насколько каждая из сред соответствует современным стандартам веб-технологий, таким как HTML5, CSS3, и другие [3, c. 15-19]

Angular – это фреймворк, разработанный компанией Google [1]. Он предоставляет инструменты для создания масштабируемых и высокопроизводительных веб-приложений. Angular использует язык TypeScript, который является надстройкой над Javascript, предоставляет более строгую типизацию и уменьшает количество ошибок в коде. Помимо этого, Angular использует RxJS для управления асинхронными операциями и работы с потоками данных [2]. Использование RxJS в Angular повышает производительность приложений, уменьшает количество ошибок и упрощает работу с асинхронными операциями.

В настоящее время Angular очень популярен и широко используется по всему миру. Благодаря обширному распространению для него имеется большое количество сообществ. Он содержит хорошо описанную документацию и множество встроенных функций, позволяющих создавать сложные приложения без необходимости использования сторонних пакетов. Еще одной из отличительных особенностей Angular является двустороннее связывание данных, которое позволяет мгновенно отобразить изменения, внесенные в модель.

Angular позволяет разделить задачи на логические блоки, что снижает время начальной загрузки веб-страницы. Используемая модель MVC обеспечивает разделение ответственности, при этом часть представления находится на стороне клиента, что значительно уменьшает запросы в фоновом режиме. Однако из-за сложности структуры фреймворка и его большого размера, он может сильно нагружать проекты и при неправильном использовании снижать производительностью. Кроме того, из-за многих встроенных функций, Angular может потребовать более высоких ресурсов для запуска.

React – это библиотека, созданная компанией Meta (ранее Facebook) и используемая для создания компонентов пользовательского интерфейса с состоянием и возможностью многократного использования для разработки больших веб-приложений, которые могут изменять данные без перезагрузки страницы [3].

Основная миссия библиотеки React – быть простым, легкообучаемым, масштабируемым и быстрым. Это позволяет React оставаться одним из самых используемых инструментов для создания веб-приложений уже несколько лет. Данные по количеству скачиваний библиотеки за последний год в сравнении с другими можно увидеть на графике (рис. 1) [4].

Рис. 1. Сравнительные данные о количестве скачиваний инструментов React, Angular и Vue за последний год

Одним из серьезных преимуществом использования React является его производительность. С объемом сжатого файла в 34,8 КБ, эта библиотека быстро приобрела известность благодаря своей невероятной скорости и множеству функций, которые делаю это возможным, таких как:

1. Использование виртуального DOM для максимальной эффективности путем перерисовки узлов только по мере необходимости. С помощью него разработчики могут легко повысить производительность и оптимизировать рабочую нагрузку приложений [1];
2. Поддержка Tree Shaking (древовидной оптимизации) для минимизации нагрузки на ресурсы конечного пользователя [1];
3. Поддержка серверного рендеринга (SSR) для улучшения производительности, видимости в поисковых системах и социального распространения [5].

В отличие от Angular одностороннее связывание данных React обеспечивает однонаправленный поток данных, улучшая управляемость компонентами, обеспечивая более легкое обнаруживание и отлаживание ошибок в приложении и упрощая процесс тестирования кода.

Сравнивая размер файлов фреймворков (рис. 2), можно заметить, что React почти в 30 раз меньше Angular и в 15 раз меньше Vue [4].

Рис. 2. Сравнительные данные о размере файлов инструментов React, Angular и Vue за последний год

Наряду с очень малым размером файла библиотеки необходимо заметить, что React не реализует MVC паттерн. Это означает, что разработчики должны использовать дополнительные библиотеки для управления состояния и модели, реализации API слоя, и другого.

Vue.js – это прогрессивный фреймворк для создания пользовательских интерфейсов [6]. Он также является одним из наиболее популярных фреймворков Javascript в NPM благодаря своим функциям, относительно простому порогу вхождения и способности создавать эффективные, быстрые и сложные одностраничные приложения.

Vue легче сравнивать с React из-за их относительно схожей миссии и синтаксиса, например:

1. Они используют виртуальный DOM;
2. Они используют реактивные и композируемые компоненты представления;
3. Они предоставляют только основной функционал, делегируя остальные операции thrid party библиотекам.

Когда дело доходит до производительности, Vue является исключительно быстрым инструментом. Однако Vue – довольно новый фреймворк, поэтому по сравнению с сообществом React или Angular, у него небольшая аудитория и ему потребуется немало времени, чтобы получить всеобщее признание.

Выбирая технологию для создания веб-приложений, очень трудно найти решения, которые подходят для любой ситуации. При этом важным фактором при выборе фреймворка или библиотеки будет лежать на производительности. Если проект небольшой, то об этом можно не сильно беспокоиться. Однако, с увеличением объема и сложности проекта, этот фактор может стать одной из ключевых проблем.

На основе проведенного сравнительного анализа в JS Framework Benchmark (рис. 3) между Angular, Vue и React можно заметить, что Vue показывает себя лучше остальных по времени первоначального запуска приложения [7].

Рис. 3. Метрики по первоначальному запуску Angular, React и Vue

Когда речь идет об управлении памяти и времени загрузки, React и Vue показывают наилучшие результаты, в то время как Angular намного медленнее. Angular может занять от 150 мс для загрузки основного скрипта и требует больше памяти для работы.

Для наглядности рассмотрим показатели Google Chrome Labs (рис. 4), которые показывают данные производительности тысяч сайтов, написанных с использованием Angular, React и Vue [8].

Рис. 4. Анализ производительность веб-приложений с использованием Angular, React, Vue от компании Google Chrome Labs

По времени на первоначальную отрисовку (First Contentful Paint) сайты на Vue и React имеют более высокий рейтинг по сравнению с Angular, который может занимать значительно больше времени на запуск и представление контента пользователю.

По времени на отрисовку самого большого элемента содержимого страницы (Largest Contentful Paint) Angular также является самым медленным из трех фреймворков в рендеринге, при этом только 27 процентов веб-сайтов на Angular попадают в допустимый диапазон.

Если рассматривать задержку первого ввода пользователя (First Input Delay), то анализ показывает, для всех трех фреймворков более 80 процентов веб-сайтов находятся в допустимом диапазоне для первой задержки ввода, которая показывает, сколько времени занимает взаимодействие пользователя со страницей.

В заключение можно отметить, что выбор между React, Angular и Vue может быть довольно сложным, поскольку каждый фреймворк имеет свои преимущества и недостатки. Поэтому при выборе нужного инструмента в большинстве случаев лучше руководствоваться требованиями проекта и его размера, а также учитывать опыт команды разработчиков.

При сравнении производительности React является одним из самых быстрых благодаря использованию виртуального DOM, который позволяет эффективно перерисовывать нужные части приложения. Angular, в свою очередь, более подходит для крупных проектов, где требуется многофункциональность и гибкость. Vue же демонстрирует высокую производительность и простоту использования, но имеет более ограниченное сообщество и меньшую популярность на рынке.

**Заключение**

**Список используемой литературы**

Великолепный PWA — Статья на vc.ru про PWA

Научная статья опубликованная в журнале «Известия Санкт-Петербургского государственного экономического университета» С.Л.Торосян «Роль информационных технологий в бизнес-проекте»

Научная статья журнала «Вестник науки» Терехина С.В. «Современные информационные технологии в бизнесе»

1. Документация Google по Angular. URL: https://angular.io/docs
2. RxJS for Reactive programming. URL: https://angular.io/guide/rx-library
3. Документация Facebook по React. URL: https://react.dev
4. Количество скачиваний React, Angular и Vue за последний год. URL: https://npmtrends.com/angular-vs-react-vs-vue
5. Server React DOM APIs. URL: https://react.dev/reference/react-dom/server
6. Progressive Javascript Framework. URL: https://vuejs.org
7. RawGit. JS Framework Benchmark URL: https://rawgit.com/krausest/js-framework-benchmark/master/webdriver-ts-results/table.html
8. Web Apps Performance Track Metrics by Google Chrome Labs. URL: https://perf-track.web.app/

**Приложения**

**Основные требования к оформлению дипломного проекта**

* Текст проекта оформляйте только с одной стороны листа формата А4. При наборе используйте межстрочный интервал 1,5.  
  Объемные таблицы/иллюстрации возможно размещать на листах А3, А2 и выносить в приложения.
* При наборе информации, используйте шрифт Times New Roman с выравниванием текста по ширине. Рекомендуем кегль — 12–14, для некоторых структурных единиц работы можно выбрать полужирное выделение.
* Поля страницы, где размещается текст, выставляйте по определенным показателям: нижнее/верхнее – 20 мм, правое – 10 мм, левое (для последующей прошивки) – 30 мм.
* Все листы дипломного проекта нумеруются по порядку, начиная с введения (используется сквозная нумерация). Номер указывайте в центре нижней части страницы без точки.
* Заголовок раздела с одним подразделом или пунктом не нумеруется.
* После текста перед следующим заголовком установите одиночный интервал.
* После заголовка перед подзаголовком проставьте двойной интервал.
* Имена собственные — наименования компаний/изделий, фамилии учёных и т. д., указывайте на языке оригинала.
* Каждый структурный элемент начинается с нового листа.

*Вся работа должна составлять 50+ страниц формата А4, включая приложения.*

|  |
| --- |
| 💡 Инструкция по работе с шаблоном.   * Создайте копию данного шаблона. Файл - Создать копию. * Изучите шаблон и начинайте работу. |

**Содержание**

**Как оформить содержание?**

В содержании дипломного проекта необходимо озаглавить основные блоки и подпункты работы с указанием страниц.

**Пример**

*Данный пример не является эталонным, лишь один из вариаций.*

Введение (2–3 стр.)

Глава 1. Основы ручного тестирования веб-приложений (~15 стр.)

1.1 Что такое веб-приложение, его особенности

1.2 Что такое тестирование, кто такой тестировщик. Зачем нужно тестирование и когда оно начинается

1.3 Этапы ручного тестирования веб-приложений

1.4 Источники требований: что такое техническое задание и как проводить ручное тестирование, если техническое задание отсутствует

Глава 2. Подготовка к ручному тестированию веб-приложения (~15 стр)

2.1 Составление чек-листов

2.2 Составление тест-кейсов

2.3 Разработка тест-плана

2.4 Тест-дизайн и тест-аналитика

2.5 Баг-репорты и багтрекинговая система

Глава 3. Проведение тестирования и разработка предложений по улучшению тестирования Интернет-магазина РИВ ГОШ (~20 стр.)

3.1 Тестирование функциональности

3.2 Тестирование юзабилити

3.3 Тестирование интерфейса

3.4 Тестирование совместимости

3.5 Тестирование производительности

3.6 Тестирование безопасности

3.7 Подведение итогов. Разработка предложений по улучшению ручного тестирования веб-приложения РИВ ГОШ

Заключение (~ 4 стр.)

Список используемой литературы

Приложения

Введение

**Введение** раскрывает обоснование необходимости исследования выбранной студентом задачи и представляет схему проведения дипломного исследования.

В общих чертах, в этом блоке должно быть:

* Что из себя представляет проект.
* Обоснование темы проекта.
* Цель проекта.
* План работы.
* Какую проблему будет решать ваш проект.
* По какой специализации вы будете делать дипломный проект.
* Есть ли у вас полезный опыт для решения этой задачи. Если да, опишите его.
* Какими инструментами вы будете пользоваться.
* Какие технологии вы планируете использовать при выполнении проекта.
* Состав команды. Даже если делали все сами, напишите, какие роли вы выполняли на проекте.

Хороший пример: Участвовала в командном проекте в роли Java-разработчика. Также на проекте были задействованы: проджект-менеджер, дизайнер и тестировщик.

**Пример**

*Данный пример не является эталонным, лишь один из вариаций.*

**Тема проекта:** Исследование особенностей организации процесса ручного тестирования веб-приложений в условиях отсутствия технического задания на примере тестирования интернет-магазина парфюмерии и косметики.

**Цель:** Изучить особенности ручного тестирования веб-приложений в условиях отсутствия спецификации к программному продукту и разработать предложения по улучшению процесса ручного тестирования веб-приложения РИВ ГОШ.

**Какую проблему решает:** Случается такое, что необходимо протестировать и выпустить продукт без требований, или на их изучение нет времени.

**Задачи:**

* Изучить литературу, касающуюся темы исследования.
* Рассмотреть основные виды и методы ручного тестирования веб-приложений.
* Ознакомиться с основными принципами составления, такой тестовой документации, как чек-листы, тест-кейсы, тест-планы, баг-репорты.
* Составить план ручного тестирования веб-приложения РИВ ГОШ с использованием техник тест-дизайна.
* Выполнить ручное тестирование веб-приложения РИВ ГОШ.
* Разработать предложения по улучшению ручного тестирования веб-приложения РИВ ГОШ.

**Инструменты:** Postman, Qase.io, TestRail или Jira, DevTools, Git, Charles Proxy, FastStone Capture 7.2.

**Состав команды**: ФИО (Тестировщик)

**Теоретическая и практическая главы**

**Основная часть** составляет 80% от всего объёма написанного. В нём последовательно рассказывается, как были решены вопросы, перечисленные во вступлении, какие цели достигнуты и с каким результатом.

Содержит 2 главы — теоретическую и практическую, где автор выражает свою позицию относительно гипотез.

В практической части студент должен пошагово описать:

* как он делал проект;
* какие инструменты использовал в работе;
* на каком этапе использовал инструменты и как они помогли в достижении цели работы.

В основной части должны быть практические примеры, результаты и выводы после каждого элемента исследования.

**Заключение**

В заключение необходимо включить следующее:

* Краткие и ёмкие теоретические и практические выводы, которые были получены во время анализа теоретической базы и практического исследования.
* Оценка проведённого исследования, описание его результатов.
* Практическая значимость работы, рекомендации и планы на дальнейшие исследования.
* Общий итог — достижение цели, выполнение задач, доказательство гипотезы.
* Предложения по совершенствованию объекта исследования.

**Список используемой литературы**

Здесь нужно будет указатьсписок используемой литературы, ссылки на все ресурсы, которые нужны были для создания проектной работы.

Основные правила оформления использованной литературы и ресурсов:

* Каждый источник упоминается единожды, независимо от того, насколько часто на него ссылаются.
* Список литературы оформляется в алфавитном порядке по фамилии автора, сначала русскоязычная литература, затем иностранная, далее интернет-сайты.
* Библиографическая запись обязательно включает:
* Фамилию автора или фамилии их группы, инициалы (при наличии).
* Название статьи, книги, справочника, закона, иного документа.
* Населённый пункт, в котором был издан источник, наименование издательства.
* Год публикации.
* Число страниц.

**Пример**

* *Книга: Автор. Название книги. Город: Издательство, Год.*
* *Статья: Автор. "Заголовок статьи." Название журнала Том, номер (Год): страницы.*

**Приложения**

В **приложения** обычно входят артефакты, получившиеся в процессе создания проекта:

* Объёмные графики и таблицы, которые не помещаются на лист А4.
* Длинные математические формулы и расчёты по ним.
* Характеристики аппаратуры, которая использовалась для проведения исследования.
* Авторские методики.
* Вспомогательный материал: тесты, карточки, схемы, рисунки.
* Материалы, полученные на предприятии: отчёты, прочие документы.