Geekbrains

**Исследование особенностей организации процесса ручного тестирования веб-приложений в условиях отсутствия технического задания на примере тестирования интернет-магазина парфюмерии и косметики РИВ ГОШ**

Программа:

Специализация

ФИО

Москва

2024

**Содержание**

[Введение 2](#_Toc161670394)

[1. Теоретическая часть 4](#_Toc161670395)

[1.1 Основные принципы работы с Vue.js и его преимущества 4](#_Toc161670396)

[1.2 Сравнение Vue с другими фронтенд-технологиями 6](#_Toc161670397)

[1.3 Архитектура приложения на Vue.js 6](#_Toc161670398)

[1.4 Возможности использования компонентов и маршрутизации в Vue 6](#_Toc161670399)

[2. Практическая часть 6](#_Toc161670400)

[2.1 Настройка среды разработки 6](#_Toc161670401)

[Теоретическая и практическая главы 6](#_Toc161670402)

[Заключение 7](#_Toc161670403)

[Список используемой литературы 8](#_Toc161670404)

[Пример 8](#_Toc161670405)

[Приложения 9](#_Toc161670406)

# Введение

Темой дипломного работы, выполненной по специализации "Frontend разработка (Vue.js)", является "Разработка сайта-каталога с использованием Vue.js". Проект представляет собой интерактивный веб-сайт каталог, который будет содержать информацию о различных марках и моделей электроавтомобилей.

Выбор темы проекта обусловлен актуальностью развития технологий фронтенд-разработки, особенно использование современного JavaScript фреймворка Vue для создания динамичных и отзывчивых пользовательских интерфейсов.

Целью данного дипломного проекта является разработка интерактивного веб-сайта каталога на Vue.js, который предоставит удобный инструмент для просмотра и поиска информации.

План работы включает в себя следующие этапы:

1. Проектирование интерфейса и структуры сайта;

2. Разработка архитектуры приложения и выбор подходящих технологий;

3. Разработка компонентов и функционала сайта на Vue.js.

4. Интеграция с API для получения информации.

5. Тестирование, оптимизация и доработка функционала;

6. Подготовка документации и презентации проекта.

Дипломный проект направлен на создание удобного и интуитивно понятного интерфейса для просмотра и поиска информации в каталоге, что позволит пользователям быстро и эффективно находить необходимую информацию. Предоставление эффективного и удобного способа доступа к информации о товарах повышает качество обслуживания и уровень удовлетворенности пользователя.

Для реализации проекта планируется использовать **инструменты разработки**, такие как Visual Studio Code, Git для контроля версий, Vue CLI для создания и настройки проекта, Postman для тестирования API.

**Технологический стек** проекта будет включать в себя: Vue.js в качестве основного фреймворка для работы с пользовательским интерфейсом, Vuex для управления состоянием приложения, Vue Router для маршрутизации, Axios для работы с HTTP-запросами, RESTful API для обмена данными, а также CSS и HTML для стилизации и структурирования веб-страниц.

**Состав команды:**

Евтеева Анастасия Алексеевна (Frontend-разработчик (Vue.js)).

# Теоретическая часть

## 1.1 Основные принципы работы с Vue.js и его преимущества

Vue.js - это современный фреймворк JavaScript, который позволяет создавать интерактивные веб-приложения с помощью простого и интуитивно понятного синтаксиса. Он широко применяется для разработки одностраничных приложений (SPA) и веб-интерфейсов.

Vue.js сравнительно легче изучить, особенно по сравнению с другими JavaScript фреймворками, такими как React.js и Angular.js. В отличие от других больших фреймворков, Vue разработан так, чтобы быть постепенно принимаемым.

Vue.js применяет несколько основных принципов, которые делают работу с этим фреймворком удобной и эффективной:

1. Декларативность: Vue.js позволяет описывать, что должно произойти, на основе состояния приложения, используя HTML-шаблоны. Вы просто говорите приложению, что хотите отобразить в браузере, на основе состояния Vue и он обновляет DOM за вас.
2. Компонентный подход: Во Vue.js приложения строятся из модульных, переиспользуемых компонентов. Вы можете создать компоненты для всего - заголовка, секции навигации, частей формы. Компоненты могут включать другие компоненты, что позволяет создавать сложные интерфейсы из простых частей. Проектирование с использованием компонентов позволяет разделить работу над проектом и делает код более легким для понимания и поддержки.
3. Реактивность: Vue.js позволяет легко и эффективно следить за изменениями стейта приложения и автоматически обновлять DOM при изменении данных.
4. Двунаправленная связь данных: Vue.js обеспечивает простой способ связывания данных между моделью и представлением, что позволяет автоматически обновлять пользовательский интерфейс при изменении данных
5. Директивы: Vue предлагает множество встроенных директив для управления DOM-элементами, обработки событий и условного отображения.
6. Управление состоянием: для управления состоянием приложения в Vue.js часто используется библиотека Vuex, которая предоставляет инструменты для централизованного хранения и управления данными приложения.
7. Маршрутизация: в Vue.js имеется возможность добавления маршрутизации в приложение с помощью плагина Vue Router, который позволяет создавать многостраничные приложения и управлять навигацией.

Преимущества работы с Vue.js включают в себя:

* Простота: Vue.js очень интуитивен и прост в изучении, особенно для разработчиков, знакомых с HTML и JavaScript.
* Гибкость: Vue.js может использоваться для создания как небольших, так и крупных масштабируемых приложений. Он прекрасно работает в сочетании с другими библиотеками и инструментами.
* Производительность: Vue.js обеспечивает высокую производительность приложений благодаря интеллектуальному отслеживанию зависимостей и асинхронному обновлению DOM.
* Поддержка сообщества: Vue.js имеет активное и быстрорастущее сообщество разработчиков, которые постоянно работают над улучшением фреймворка и создают плагины для расширения его функциональности.
* Большая информационная база: есть масса справочных материалов и обучающих ресурсов по работе с этим фреймворком.

В целом, Vue.js характеризуется гибкостью, функциональностью и несложностью в использовании, как для новичков, так и для опытных разработчиков, что обусловливает его непрерывно растущую популярность.

## 1.2 Сравнение Vue с другими фронтенд-технологиями

## 1.3 Архитектура приложения на Vue.js

## 1.4 Возможности использования компонентов и маршрутизации в Vue

# Практическая часть

## 2.1 Дизайн веб-интерфейса

Для того чтобы начать frontend разработку нам потребуется дизайн нашего будущего сайта или приложения.

Дизайн включает в себя разработку внешнего вида веб-сайта - от цветовой схемы и шрифтов до компоновки элементов на странице и графических элементов. Цель дизайна интерфейса состоит в создании привлекательного и удобного для использования внешнего вида, который ориентирован на потребности пользователей.

Возможно у Вас есть уже готовый дизайн, выполненный профессионалом или же Вы рассматриваете шаблонный дизайн, который можно скачать бесплатно на просторах Internet – любой из вариантов подходит, главное подобрать шаблон, который будет соответствовать задумке Вашего проекта.

Я буду использовать Макет[[1]](#footnote-1) в Figma, который как мне кажется подходит для сайта-каталога автомобилей.

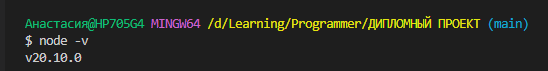
## 2.1 Настройка среды разработки

Для настройки среды разработки сайта на фреймворке Vue.js Вам понадобится выполнить несколько шагов:

1. Установите Node.js:

Скачайте и установите Node.js с официального сайта[[2]](#footnote-2): Node.js включает в себя npm (Node Package Manager), который понадобится для установки необходимых пакетов.

Проверьте его работоспособность введя в терминале или командной строке команду. Если все успешной установилось Вы увидите текущую версию Node.js.



1. Установка Vue CLI

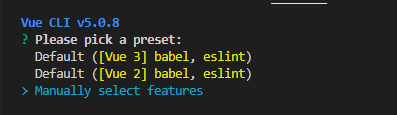
Следуем инструкции с официального сайта [[3]](#footnote-3)Vue.js

1. Создание проекта

Для создания нового проекта вводим в терминале или командной строке команду:

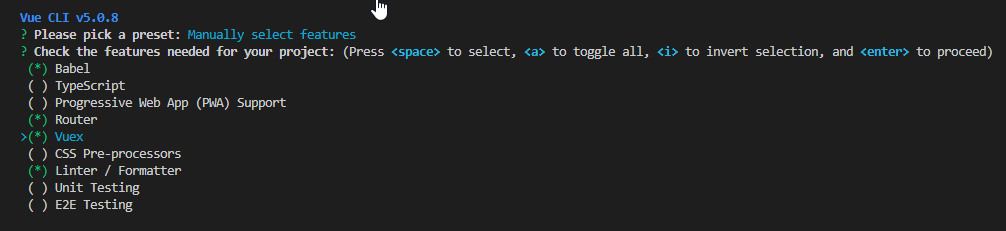


Далее выбираем необходимые опции. Вы можете выбрать как дефолтные пакеты, так и составить пакет самостоятельно. Я выберу последний вариант.

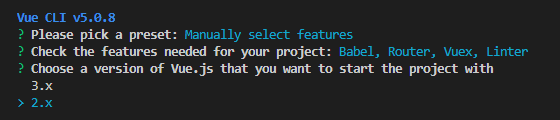


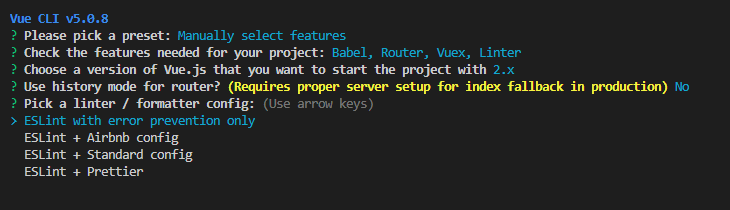
Как говорилось ранее для разработки будут использоваться такие плагины как Vue Router для маршрутизации и Vuex для управления данными. Их возможно установить на данном этапе при создании проекта, выбрав соответствующие опции.

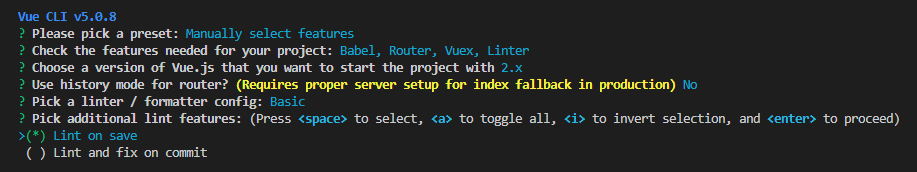
Если же Вы выбрали дефолтные пакеты, данные плагины можно доустановить отдельно следуя инструкциям[[4]](#footnote-4)[[5]](#footnote-5) на официальном сайте.



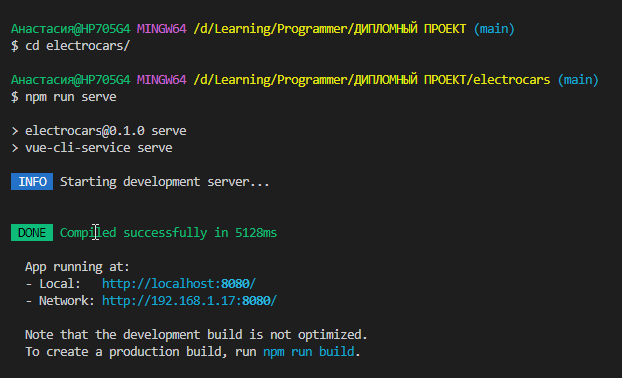
Для своего проекта буду использовать Vue 2.







После установки пакета нам необходимо перейти в директорию проекта и запустить сервер. Если все было сделано верно, Вы получите рабочие ссылки на свой проект.



# 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:

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

# Список используемой литературы

Интернет-сайты:

https://nodejs.org/

<https://www.figma.com/file/IGjuti65kHoQ30eOZ9TWRJ/ECOMOBIL-(Copy)-(2)-(Copy)?type=design&node-id=48-5&mode=design&t=S3tuj0CXDr4fHUFC-0>

https://vuex.vuejs.org/

<https://router.vuejs.org/>

https://cli.vuejs.org/ru/

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

Основные правила оформления использованной литературы и ресурсов:

1. Каждый источник упоминается единожды, независимо от того, насколько часто на него ссылаются.
2. Список литературы оформляется в алфавитном порядке по фамилии автора, сначала русскоязычная литература, затем иностранная, далее интернет-сайты.
3. 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:

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

2. Теоретическая часть

2.1. Обзор фреймворка Vue.js

2.2. Основные принципы работы с Vue и его преимущества

2.3. Сравнение Vue с другими фронтенд-технологиями

2.4. Архитектура приложения на Vue.js

2.5. Возможности использования компонентов и маршрутизации в Vue

3. Практическая часть

3.1. Проектирование интерфейса сайта

3.2. Разработка компонентов на Vue

3.3. Использование API для получения данных

3.4. Взаимодействие с бэкендом через REST API

3.5. Тестирование разработанных компонентов

3.6. Оптимизация производительности сайта

4. Результаты и выводы

4.1. Анализ полученных результатов

4.2. Выводы по исследованию фронтенд-разработки на Vue

4.3. Рекомендации по использованию Vue в проектах

Заключение

5.1. Обобщение основных пунктов работы

5.2. Перспективы развития и дальнейшие исследования

6. Список использованной литературы

1. https://www.figma.com/file/IGjuti65kHoQ30eOZ9TWRJ/ECOMOBIL-(Copy)-(2)-(Copy)?type=design&node-id=48-5&mode=design&t=S3tuj0CXDr4fHUFC-0 [↑](#footnote-ref-1)
2. https://nodejs.org/ [↑](#footnote-ref-2)
3. https://cli.vuejs.org/ru/ [↑](#footnote-ref-3)
4. https://vuex.vuejs.org/ [↑](#footnote-ref-4)
5. https://router.vuejs.org/ [↑](#footnote-ref-5)