

Предлагаем использовать чек-лист для разработки расширений для сайта, который включает: машинопись, сборку, подключение библиотеки компонентов, использование фреймворка, кроссбраузерность, горячую перезагрузку, изоляцию стилей расширения от стилей страницы, возможность работы с переменными окружения, возможность писать тесты, синхронизацию с магазином, и генерация API по схеме swagger.

1. Данил Вахрушев, Tinkoff – Менеджер Паролей. В кармане. Разработка. Реализация. Детали. – URL: <https://goo.su/mbaDB8W> (дата обращения: 12.02.2023).

2. Hello, Word! Разрабатываем браузерное расширение в 2021-м. Блог компании Яндекс Практикум JavaScript. Программирование. Расширения для браузеров. – URL: <https://goo.su/hD6Ctq> (дата обращения: 13.02.2023).

3. Пишем безопасное браузерное расширение. Блог компании Waves. Децентрализованные сети. JavaScript. Расширения для браузеров. Браузеры. – URL: <https://goo.su/te0b> (дата обращения: 12.02.2023).

4. Рудольф Коршун. Создаём расширение для Chrome. – URL: <https://goo.su/hx90> (дата обращения: 12.02.2023).

5. Ларс Колхер. Браузерное расширение своими руками. – URL: <https://goo.su/i6Sm5> (дата обращения: 12.02.2023).

6. Как написать кроссбраузерное расширение в 2022 году. Блог компании СберМаркет. Разработка веб-сайтов. JavaScript. Расширения для браузеров. Браузеры. – URL: <https://goo.su/JQG4rCV> (дата обращения: 14.02.2023).

7. Шахед Насер. Обеспечение совместимости вашего расширения как с Chrome, так и с Firefox. – URL: <https://goo.su/YRRvZPX> (дата обращения: 15.02.2023).

УДК 004.432.2

АКТУАЛЬНЫЕ КОНЦЕПЦИИ ФРОНТЕНД РАЗРАБОТКИ

А.Д. Рыбаков
бакалавр

*Владивостокский государственный университет
Владивосток. Россия*

В данной статье рассмотрены основные концепции веб-разработки, современная библиотека React, а также методология Feature-sliced. Описано, как эти инструменты могут повысить эффективность работы фронтенд-разработчиков и улучшить качество создаваемых ими проектов. Использование подхода Feature-sliced позволяет разбить проект на более мелкие части и упростить процесс разработки, тестирования и поддержки проекта.

Ключевые слова: фронтенд-разработка, feature-sliced, компонентный подход, язык разметки, язык стилей, react, styled-components.

ACTUAL CONCEPTS OF FRONT-END DEVELOPMENT

This article covers the main areas of web development, the modern React library, and the Feature-sliced methodology. It is described how these tools can increase the efficiency of the front-end developer and improve the quality of their project transfer. The use of Feature-sliced allows you to break the project into smaller parts and simplify the process of developing, testing and maintaining the project.

Keywords: front-end development, functional slicing, component approach, markup language, style language, react, styled-components.

Данная статья посвящена повышению эффективности в области веб-разработки, рассматривая основные концепции этой области. В работе представлены современные технологии разработки, применяемые крупными компаниями, а также рассмотрена методология Feature-sliced.

Развитие веб-технологий за последние годы привело к резкому увеличению количества задач, которые ставят перед фронтенд-разработчиками. Одним из ключевых моментов для повышения эффективности работы в этой области является усвоение основных концепций веб-разработки. Эти концепции включают в себя структуру и семантику веб-страниц, принципы каскадных таблиц стилей (CSS) и интерактивность с помощью JavaScript.

Кроме того, для повышения эффективности работы фронтенд-разработчиков важно уметь использовать современные технологии разработки. В данной работе рассматривается технология ReactJS, которая широко применяется крупными компаниями в своих проектах.

В качестве дополнительного инструмента для повышения эффективности работы фронтенд-разработчиков рассматривается методология Feature-sliced. Этот подход основан на создании компонентов, каждый из которых содержит все необходимые элементы для выполнения определенной функции. Это позволяет разбить проект на более мелкие части и упростить процесс разработки, тестирования и поддержки проекта.

В целом, применение основных концепций фронтенд разработки, использование современных технологий и методологии Feature-sliced позволят повысить эффективность работы фронтенд-разработчиков и улучшить качество создаваемых ими проектов.

Фронтенд – это часть веб-разработки, которая отвечает за создание пользовательского интерфейса (UI) и пользовательского опыта (UX) веб-сайта или веб-приложения. Фронтенд-разработчики работают с языками разметки, стилей и программирования, то есть HTML, CSS и JavaScript соответственно, чтобы создавать веб-страницы, которые могут быть просмотрены и взаимодействовать с пользователем через веб-браузер.

Основные концепции фронтенд-разработки включают в себя:

- HTML: это язык разметки, используемый для создания структуры и содержимого веб-страницы;

- CSS: это язык таблиц стилей, который используется для задания внешнего вида и макета веб-страницы;

- JavaScript: это язык программирования, который используется для добавления динамического поведения на веб-страницы, например, для создания интерактивных элементов, обработки событий и отправки запросов на сервер;

- Инструмент для создания интерфейса: библиотеки или фреймворки позволяющие разработчикам создавать, повторно использовать и поддерживать компоненты пользовательского интерфейса, которые эффективны и быстро реагируют на взаимодействие с пользователем;

- Адаптивный дизайн: это метод создания веб-страниц, который позволяет странице корректно отображаться на разных устройствах, таких как настольные компьютеры, планшеты и мобильные телефоны;

- Кросс-браузерность: это концепция, которая обеспечивает, что веб-страницы работают одинаково на разных браузерах и платформах;

- Оптимизация производительности: это процесс оптимизации веб-страницы для ускорения ее загрузки и улучшения пользовательского опыта;

- Контентный менеджмент: это процесс управления содержимым веб-страниц, включая его создание, редактирование и публикацию.

Все эти концепции являются важными для фронтенд-разработки и помогают создавать веб-страницы, которые были бы удобны в использовании и эффективны в работе.

HTML (HyperText Markup Language) – это язык разметки, который определяет структуру веб-страницы, включая заголовки, параграфы, списки, изображения и ссылки. Он обеспечивает основную структуру веб-страницы и содержит текстовое содержимое, которое отображается в браузере [3].

Синтаксис языка HTML содержит только теги, такие как `div`, `p`, `input`, `span`, `table` и т.д., обозначаемые как `<input>` и `</input>` или `<input />`, и атрибуты для этих тегов.

Тег – это элемент языка разметки HTML, который должен соответствовать набору тегов, относящихся к стандарту HTML5 [1].

Атрибут – это слово и значение, которое используется для управления поведением элемента HTML. Разберем один из популярных тегов для формы `input`. Данный тег был создан для ввода данных в форму, например использоваться как поле ввода для логина и пароля или служить чек-боксом для подтверждения согласия с правилами сайта и т.д. А также он может содержать такие атрибуты как: `type`, `value` и другие.

В HTML также доступны специальные теги для добавления изображений и видео на веб-страницу. Это позволяет создавать более интерактивный и визуально привлекательный контент для пользователей.

Кроме того, HTML является основой для других языков разметки, таких как XML и XHTML, которые используются для создания более сложных веб-страниц и веб-приложений. HTML также имеет множество расширений и библиотек, которые расширяют его возможности и облегчают создание и поддержку веб-страниц.

CSS (Cascading Style Sheets) – это язык таблиц стилей, который определяет внешний вид и макет веб-страницы. Он используется для управления цветом, шрифтом, размером, расположением и другими свойствами элементов на странице.

CSS имеет множество возможностей для создания красивого и функционального дизайна веб-страниц. Он позволяет разработчикам контролировать многие аспекты внешнего вида элементов на странице, что помогает создавать приятное и удобочитаемое пользовательское впечатление.

Одним из главных преимуществ CSS является возможность создания адаптивного дизайна, который подстраивается под различные размеры экранов устройств, таких как мобильные телефоны, планшеты и настольные компьютеры. Это важно для создания пользовательских интерфейсов, которые легко читаются и удобны в использовании на любом устройстве.

CSS также может быть использован для создания анимации и визуальных эффектов на странице. Это позволяет разработчикам создавать более интерактивные и привлекательные пользовательские интерфейсы, которые могут улучшить опыт пользователей.

JavaScript – это язык программирования, который является одним из самых популярных на сегодняшний день, он используется как для создания веб-приложений, так и для написания программного обеспечения на серверной стороне. Он поддерживается всеми современными браузерами и широко используется во многих индустриях, включая разработку веб-сайтов, игр, мобильных приложений и многих других.

В дополнении к этому, JavaScript позволяет разработчикам создавать интерактивные элементы на веб-страницах, такие как выпадающие меню, всплывающие окна, анимированные элементы и многое другое. Он также позволяет обрабатывать пользовательские события, такие как клики мыши, наведение курсора, ввод текста и т.д. Это дает возможность создавать более динамичный и удобный интерфейс для пользователей.

Необходимо отметить, что JavaScript используется для работы с сервером и обмена данными между клиентом и сервером. Он позволяет отправлять асинхронные запросы на сервер без перезагрузки страницы, что позволяет создавать более плавные и быстрые пользовательские интерфейсы. Библиотеки и фреймворки, такие как jQuery, React и Angular, используются для упрощения разработки на JavaScript и позволяют создавать более сложные приложения.

Использование фреймворков и библиотек является эффективным способом ускорения процесса разработки и улучшения качества кода. Они предоставляют разработчикам готовые решения для реализации сложных функциональностей и упрощают процесс создания пользовательских интерфейсов. Фреймворки и библиотеки также обеспечивают повторное использование кода, что позволяет сократить время разработки и уменьшить вероятность ошибок. Они также обеспечивают совместимость с различными браузерами и устройствами, что делает приложение доступным для более широкой аудитории. Кроме того, использование фреймворков и библиотек позволяет разработчикам сосредоточиться на бизнес-логике приложения, не тратя время на написание низкоуровневого кода. Все эти преимущества делают использование фреймворков и библиотек необходимым для современного фронтенд-разработчика.

Наиболее популярными инструментами в настоящее время являются:

- ReactJS;
- Angular;
- Vue.

Была выбрана библиотека ReactJS, одним из главных преимуществ ReactJS является его компонентный подход, который позволяет разрабатывать большие и сложные веб-приложения в более простой и модульной форме. ReactJS также обеспечивает высокую производительность, что особенно важно для веб-приложений, которые должны быстро реагировать на действия пользователей.

ReactJS является JavaScript-библиотекой с открытым исходным кодом, специализирующейся на разработке пользовательских интерфейсов, которая основывается на концепции компонентного подхода. Данный подход является одним из ключевых методов переиспользования и настройки отдельных компонентов приложения путем передачи соответствующих пропсов и использования условного рендеринга [4].

Пропсы представляют собой входные данные React-компонентов, специально предназначенные только для чтения и передаваемые от родительского компонента к дочернему компоненту. Условный рендеринг, в свою очередь, представляет собой метод отображения только тех элементов, которые соответствуют заданному условию, например, при фильтрации. Предположим, что у вас есть интернет-магазин и вы хотите добавить на свою страницу карточку товара со стикером "Новинка" для новых моделей телефонов. Однако, у других товаров стикер не нужен. Если использовать обычный подход на основе HTML, CSS и JavaScript, то пришлось бы создавать отдельный блок для каждой карточки товара и добавлять стикер вручную только для новых моделей телефонов.

Вместо этого можно использовать компонентный подход и библиотеку ReactJS. С помощью пропсов и условного рендеринга, можно создать единственный компонент карточки товара, который будет проверять, нужно ли отображать стикер "Новинка". Это не только экономит время разработки, но и повышает производительность, так как используется только один компонент для всех карточек товаров, а не отдельный блок для каждого товара.

Таким образом, использование компонентного подхода с помощью библиотеки ReactJS позволяет легко и быстро добавлять дополнительный функционал на страницу, такой как стикер "Новинка" на карточке товара, что повышает пользовательский опыт и улучшает конверсии.

В разработке веб-приложений необходимо использовать эффективные методологии, чтобы обеспечить более быстрое и качественное создание проекта. Одним из таких методов является Feature-sliced методология, которая позволяет разделять проект на отдельные компоненты и упрощает процесс разработки и поддержки приложения.

Feature-sliced – это методология разработки, которая используется для создания веб-приложений. Она основана на концепции разделения проекта на отдельные компоненты, каждый из которых отвечает за выполнение определенной функции. Это позволяет разработчикам ускорить процесс разработки, улучшить качество кода и облегчить поддержку проекта в будущем [2].

Feature-sliced подразумевает, что каждый компонент содержит все необходимые элементы для выполнения своей функции. Например, компонент "Карточка продукта" может содержать все стили, изображения, кнопки и формы, необходимые для отображения информации о продукте. Это позволяет разработчикам избежать ошибок, связанных с несоответствием стилей или функциональности, и ускоряет процесс создания новых компонентов.

Преимущества методологии Feature-sliced для веб-разработки очевидны. Она ускоряет процесс разработки и улучшает качество кода, что позволяет создавать более качественные и надежные веб-приложения. Кроме того, методология облегчает поддержку проекта в будущем, упрощает процесс тестирования и уменьшает вероятность возникновения ошибок.

Пример использования feature sliced может быть следующим. Предположим, что команда разработчиков работает над разработкой электронной коммерции для продажи продуктов. Одна из главных функций, которую необходимо реализовать, – это функция оплаты.

Вместо того чтобы начинать сразу с разработки полной функции оплаты, команда разработчиков может применить методологию feature sliced и разделить функцию оплаты на более мелкие части. Например, команда может начать с создания базового макета страницы оплаты, затем добавить возможность выбора способа оплаты, затем добавить проверку данных оплаты, затем обработку платежа и т.д.

Каждый из этих этапов можно рассматривать как отдельную "фичу" (feature), которую можно быстро разработать, протестировать и запустить. Кроме того, такой подход позволяет более эффективно управлять рисками и сложностью разработки, а также быстрее получать обратную связь от пользователей и вносить необходимые изменения.

Однако, не стоит забывать о том, что методология Feature-sliced требует определенного уровня опыта в веб-разработке. Разработчикам, не знакомым с этой методологией, может потребоваться некоторое время для изучения ее принципов и привыкания к новому способу разработки.

В целом, методология Feature-sliced является эффективным инструментом для веб-разработки. Она помогает ускорить процесс разработки, улучшить качество кода и облегчить поддержку проекта в будущем. Если вы хотите повысить свою производительность в веб-разработке, то методология Feature-sliced может быть хорошим выбором для вас. В ходе анализа предметной области веб-разработки выявлены основные инструменты, которые используются для создания веб-страниц и приложений, установлено, что HTML, CSS и JavaScript являются базовыми языками веб-разработки, которые обеспечивают структуру, стилевое оформление и динамическое поведение веб-страниц. Кроме того, обнаружено, что существует множество библиотек и фреймворков, которые упрощают и ускоряют процесс разработки веб-интерфейсов, таких как ReactJS.

В результате можно сделать вывод о том, что веб-разработка является сложным и многогранным процессом, требующим знания и умения работы с различными языками программирования, инструментами и технологиями. Однако, благодаря использованию современных инструментов, таких как библиотеки и фреймворки, разработка веб-интерфейсов может быть ускорена и упрощена, что в свою очередь позволяет создавать более совершенные и функциональные веб-приложения.

-
1. Справочник по HTML. – Текст: электронный. – URL: <http://htmlbook.ru/html>
 2. Feature-sliced. – Текст: электронный. – URL: <https://feature-sliced.design>
 3. HTML. – Текст: электронный. – URL: <https://developer.mozilla.org/ru/docs/Web/HTML>
 4. ReactJS. – Текст: электронный. – URL: <https://react.dev/>

УДК 656.132.6, 629.341

ЛЬГОТНЫЕ ПАССАЖИРСКИЕ ПЕРЕВОЗКИ

К.К. Свиридова

бакалавр

О.В. Гриванова

канд. техн. наук, доцент

*Владивостокский государственный университет
Владивосток, Россия*

В данной статье рассмотрены льготные и субсидированные пассажирские перевозки. Предметом исследования являются льготы и субсидии, подходящие для студентов. Цель данной работы анализ существующих льготных программ, ознакомление с особенностями. Задача данной работы поиск информации и правовых документами, анализ и структурирование данных. Результат работы отбор актуальных программ, поиск малоизвестных льгот, структурирование их.

Ключевые слова: льготы, субсидии, молодежная субсидия, субсидия по месту жительства, студенческий билет, международное удостоверение.

PREFERENTIAL PASSENGER TRANSPORTATION

This article discusses preferential and subsidized passenger transportation. The subject of the study is benefits and subsidies suitable for students. The purpose of this work is the analysis of existing preferential programs, familiarization with the features. The task of this work is the search for information and legal documents, analysis and structuring of data. The result of the work is the selection of relevant programs, the search for little-known benefits, their structuring.

Keywords: benefits, subsidies, youth subsidy, subsidy at the place of residence, student card, international certificate.

Актуальность темы льготных пассажирских перевозок обусловлена государственными программами поддержки компаний перевозчиков, которые направлены на снижение стоимости проезда в транспорте. Делается это для того чтобы молодые люди и другие льготные кате-