Курсовая работа

РАЗРАБОТКА WEB-ИНТЕРФЕЙСА ИНТЕРНЕТ МАГАЗИНА

Студент гр. ИС-17 _____ И.П. Ивановский
подпись, дата

Руководитель _____ П.Г. Деркаченко
подпись, дата

Работа защищена «___» _____ 2020 г. с оценкой «_____».

Члены комиссии _____ _____
подпись Фамилия И.О.

_____ _____
подпись Фамилия И.О.

Витебск, 2020

СПИСОК СОКРАЩЕНИЙ

HTML - HyperText Markup Language - «язык гипертекстовой разметки»
CSS - Cascading Style Sheets - каскадные таблицы стилей
SCSS - Sassy CSS
JS - JavaScript
TS - TypeScript
SPA - Single Page Application - Одностраничное приложение
MPA - Multi Page Application - Многостраничное приложение
JSON - JavaScript Object Notation
NPM - Node Package Manager
CLI - Command Line Interface
VSC / VSCode - Visual Studio Code
URL - Uniform Resource Locator - унифицированный указатель ресурса
SVG - Scalable Vector Graphics - масштабируемая векторная графика
UI - User Interface - пользовательский интерфейс
Framework - остов, каркас, структура; программная платформа, определяющая структуру программной системы
PC, desktop - стационарный персональный компьютер, предназначенный для работы в офисе и дома
Open-source - открытое программное обеспечение

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Курсовой проект по теме:
«РАЗРАБОТКА WEB-ИНТЕРФЕЙСА ИНТЕРНЕТ МАГАЗИНА».

Прототип: https://nabludenie.by/

Текущий проект:
https://github.com/Domovikx/angular-nabludenie-by

Встречают «по одежке» не только людей. Внешний вид и удобство пользования сайтом оказывают непосредственное влияние на количество продаж и уровень дохода компании. Чтобы проект был успешным, нужно уметь грамотно представить товар и всю сопутствующую информацию, ориентируясь на конечного потребителя и его комфорт. Ведущую роль в восприятии онлайн-магазина играет пользовательский web-интерфейс, иногда его называют иначе User Interface (UI).

User Interface или пользовательский интерфейс (UI) - все, что видит посетитель, заходя на веб-сайт. Кнопки, картинки, текст, меню, разделы, блоки отражают ассортимент компании, указывают на возможности для покупателя и предоставляют способы их реализации.

Это наиболее важное средство взаимодействия с потенциальным клиентом, основной рекламный инструмент, который способен, как привлекать, так и отталкивать посетителей. В большинстве случаев именно на стадии знакомства с интерфейсом принимается решение о покупке товара или закрытии вкладки с ресурсом.

Работая над созданием или модернизацией функционирующего проекта, необходимо понимать, что каждый элемент структуры служит органичной частичкой крупного паззла. Несущественных деталей на сайте нет. Складываясь в единую общую картинку, они формируют положительное или отрицательное мнение.

UI онлайн-магазина выполняет **три основных задачи**:

1. Ознакомительная.
   Знакомит целевую аудиторию с компанией и конкретным товаром.

2. Маркетинговая.
   При условии соответствия требованиям клиента дает компании эффективную бесплатную рекламу.

3. Мотивирующая.
   Располагает пользователя к покупкам, ускоряет процесс конвертации случайного посетителя в постоянного покупателя.

При проектировании интерфейса должны быть учтены вероятные ожидания аудитории. Открывая страничку интернет-магазина, пользователь рассчитывает на:

- доступность, простоту подачи, быструю загрузку информации;
- логичное расположение элементов;
- эффективное решение актуальных проблем;
- отсутствие сложностей в процессе совершения каких-либо действий.

Правильный UI, удовлетворяющий потребности покупателя, составляет оптимальный баланс между навигацией и информационной архитектурой сайта. Благодаря этому клиент:

- легко взаимодействует с ресурсом;
- понимает процесс выполнения нужных операций на интуитивном уровне без дополнительных усилий;
- оперативно получает исчерпывающую информацию о товаре;
- совершает целевое действие, проходя минимальный путь.

С помощью только одного грамотно разработанного интерфейса интернет-магазин завоевывает доверие аудитории, получает прирост трафика и постоянных клиентов, приумножает объемы продаж и наращивает прибыль.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1. Веб-интерфейс

Веб-интерфейс (web-interface*) – это среда взаимодействия пользователя и программы или приложения, запущенной на удаленном сервере.*

Слово **интерфейс** пришло к нам из английского языка. В переводе interface обозначает *«место соприкосновения». Интерфейс - это набор инструментов, позволяющих пользователю взаимодействовать* с операционной системой компьютера, мобильного устройства или других видов техники.

В качестве подобных инструментов взаимодействия могут выступать:

- текстовые поля
- кнопки и галочки
- выпадающие списки
- всплывающие подсказки
- переключатели
- элементы меню программы или сайта

Чаще всего веб-интерфейс применяется для работы с различными онлайн сервисами: начиная с электронной почты и заканчивая системами веб-аналитики. Приставка «веб» означает, что элемент работает удаленно от компьютера пользователя на локальном или интернет-сервере. Взаимодействие с сервисом при этом происходит через «интерфейс» специальную графическую оболочку, состоящую из кнопок, окон, полей заполнения или любых других элементы.

1.2. Single Page Application (SPA)

Web-приложения делятся на одностраничные (SPA) и многостраничные (MPA). SPA-приложение, Single Page Application, или «приложение одной страницы» – это тип web-приложений, в которых загрузка необходимого кода происходит на одну страницу, что позволяет сэкономить время на повторную загрузку одних и тех же элементов.

То есть SPA подгружает все необходимые javascript и css файлы при первой загрузке страницы, а затем все общение между клиентом и сервером сводится к минимуму. При таком подходе большая часть работы сайта производится на стороне клиента, а если нужно получить данные с сервера, то это обычно делается с помощью JSON.

SPA-приложения обладают рядом преимуществ:

Доступность. Можно получить моментальный доступ к функционалу с любого типа устройства без проблем с совместимостью, достаточным объемом памяти, необходимыми вычислительными мощностями или с затратой времени на установку.

Универсальность. Использовать софт можно практически с любого устройства, если на нем есть доступ к интернету. Если при разработке интерфейса учитывались различные разрешения экрана, то использовать SPA одинаково удобно и с компьютера, и с планшета, и с телефона.

Возможность задействовать большие объемы данных. Размер приложения и используемых им данных не ограничен памятью устройства.

Скорость. Одна страница, содержащая весь необходимый интерфейс, не только экономит время на повторную загрузку данных, но и повышает производительность работы с веб-приложением.

Широта возможностей разработки. Разработчикам доступны библиотеки и фреймворки, которые упрощают создание архитектуры проекта и предоставляют немало готовых элементов для работы.

Понять, насколько удобными и полезными для пользователей бывают одностраничные приложения или SPA, можно на примере нескольких популярных сервисов гиганта Google: Gmail и Google Translate. Мы постоянно используем данные сервисы, и вряд ли у кого-то возникает желание перейти на десктопные аналоги.

Множество преимуществ и недостатков также зависят от качества разработки SPA-приложений и не обусловлены особенностями данного

вида софта. Мы же рассматриваем те плюсы и минусы, которые не обусловлены квалификацией специалистов, а являются общими для любого программного обеспечения данного типа. Об этом также стоит помнить.

Недостатки:

Необходимость интернет-соединения. Без доступа к сети использовать такой софт невозможно, это недостаток, если сравнивать с десктопным софтом, использующим только внутренний объем данных. Но если даже десктопное программное обеспечение использует в работе внешние базы данных, то доступ к интернету необходим в любом случае.

Трудности с SEO. Особенности SPA усложняют или делают невозможным процесс индексации поисковыми системами всех модулей приложения. Это может вызвать трудности с оптимизацией.

Не работает у пользователей с отключенной поддержкой JS. Многие отключают отображение JS-элементов у себя в браузерах, из-за чего Single Page Application, использующее их в работе, не функционирует.

SPA-приложения могут быть очень полезным инструментом для владельца, скорость и простота их использования в разы повышает количество потенциальных пользователей, а со временем и популярность сервиса. Сегодня их используют как дополнительные сервисы для потенциальных клиентов компании, повышая тем самым потребительскую лояльность и узнаваемость бренда, или же в качестве основного источника дохода – предоставляя уникальный функционал за абонентскую плату.

1.3. Выбор средств и инструментов разработки

В качестве основного средства разработки для проекта использован фреймворк Angular 8. Angular имеет низкий порог вхождения и высокую скорость разработки. Angular CLI упрощает работу Angular.

Одно из главных преимуществ CLI заключается в создание нового проекта, а также создании различных сервисов, компонентов и других нужд проекта. CLI Angular автоматически устанавливает собственный набор стандартов для проекта, это позволяет избежать ряда ошибок.

Angular использует TypeScript по умолчанию и это огромное преимущество. TypeScript дает вам отличную возможность использовать систему типов в вашем JavaScript коде.

Многие фреймворки, включая Angular, имеют большое количество библиотек компонентов.

1.3.1 HTML

HTML (HyperText Markup Language — «язык гипертекстовой разметки») — самый базовый строительный блок Веба. Он определяет содержание и структуру веб-контента. Другие технологии, помимо HTML, обычно используются для описания внешнего вида/представления (CSS) или функциональности/поведения (JavaScript) веб-страницы.

Под гипертекстом ("hypertext") понимаются ссылки, которые соединяют веб-страницы друг с другом либо в пределах одного веб-сайта, либо между веб-сайтами. Ссылки являются фундаментальным аспектом Веба. Загружая контент в Интернет и связывая его со страницами, созданными другими людьми, вы становитесь активным участником Всемирной паутины.

1.3.2 CSS, SCSS

CSS (Cascading Style Sheets) — язык таблиц стилей, который позволяет прикреплять стиль (например, шрифты и цвет) к структурированным документам (например, документам HTML и приложениям XML). Обычно CSS-стили используются для создания и изменения стиля элементов веб-страниц и пользовательских интерфейсов, написанных на языках HTML и XHTML, но также могут быть применены к любому виду XML-документа, в том числе XML, SVG и XUL. Отделяя стиль представления документов от содержимого документов, CSS упрощает создание веб-страниц и обслуживание сайтов.

CSS поддерживает таблицы стилей для конкретных носителей, поэтому авторы могут адаптировать представление своих документов к различным визуальным браузерам и устройствам.

Каскадные таблицы стилей описывают правила форматирования элементов с помощью свойств и допустимых значений этих свойств. Для каждого элемента можно использовать ограниченный набор свойств, остальные свойства не будут оказывать на него никакого влияния.

SCSS (Sassy CSS) является расширением синтаксиса CSS. Это означает, что любое допустимое значение в CSS стилях будет допустимо и в SCSS. Кроме того, SCSS понимает синтаксис вендорных префиксов.

1.3.3 JavaScript (JS), TypeScript (TS)

JavaScript - мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript (стандарт ECMA-262).

Изначально JavaScript был создан, чтобы «сделать веб-страницы живыми». Программы на этом языке называются скриптами. Они могут встраиваться в HTML и выполняться автоматически при загрузке веб-страницы. Скрипты распространяются и выполняются, как простой текст. Им не нужна специальная подготовка или компиляция для запуска.

Это отличает JavaScript от другого языка – Java.

Почему JavaScript? Когда JavaScript создавался, у него было другое имя – «LiveScript». Однако, язык Java был очень популярен в то время, и было решено, что позиционирование JavaScript как «младшего брата» Java будет полезно.

Со временем JavaScript стал полностью независимым языком со своей собственной спецификацией, называющейся ECMAScript, и сейчас не имеет никакого отношения к Java.

Сегодня JavaScript может выполняться не только в браузере, но и на сервере или на любом другом устройстве, которое имеет специальную программу, называющуюся «движком» JavaScript. У браузера есть собственный движок, который иногда называют «виртуальная машина JavaScript».

TypeScript - язык программирования, представленный Microsoft в 2012 году и позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript.

TypeScript является обратно совместимым с JavaScript и компилируется в последний. Фактически, после компиляции программу на TypeScript можно выполнять в любом современном браузере или использовать совместно с серверной платформой Node.js.

TypeScript отличается от JavaScript возможностью явного статического назначения типов, поддержкой использования полноценных классов (как в традиционных объектно-ориентированных языках), а также поддержкой подключения модулей, что призвано повысить скорость разработки, облегчить читаемость, рефакторинг и повторное использование кода, помочь осуществлять поиск ошибок на этапе разработки и компиляции, и, возможно, ускорить выполнение программ.

1.3.4 Angular 8, Angular CLI

Angular представляет фреймворк от компании Google для создания клиентских приложений. Прежде всего он нацелен на разработку SPA-решений (Single Page Application), то есть одностраничных приложений. В этом плане Angular является наследником другого фреймворка AngularJS. В то же время Angular это не новая версия AngularJS, а принципиально новый фреймворк.

Angular предоставляет такую функциональность, как двустороннее связывание, позволяющее динамически изменять данные в одном месте интерфейса при изменении данных модели в другом, шаблоны, маршрутизация и так далее.

Одной из ключевых особенностей Angular является то, что он использует в качестве языка программирования TypeScript.

Angular CLI - официальный инструмент для инициализации и работы с Angular-проектами. Избавляет от настройки сложных

конфигураций и инструментов сборки, таких как TypeScript, Webpack и так далее.

После установки Angular CLI нужно выполнить одну команду для генерации проекта, а другую - для его обслуживания с использованием локального сервера разработки для работы с вашим приложением.

Как и большинство современных инструментов веб-интерфейса, Angular CLI построен на основе Node.js.

Node.js - это серверная технология, которая позволяет запускать JavaScript на сервере и создавать веб-приложения на стороне сервера. Тем не менее, Angular - это интерфейсная технология, которая требует предварительной установки Node.js на компьютер для разработки, он предназначен только для запуска CLI.

1.3.5 Git, GitHub, gh-pages

Git - распределённая **система контроля версий**, которая даёт возможность разработчикам отслеживать изменения в файлах и работать совместно с другими разработчиками. Она была разработана в 2005 году Линусом Торвальдсом, создателем Linux, для того, чтобы другие разработчики могли вносить свой вклад в ядро Linux. Git известен своей скоростью, простым дизайном, поддержкой нелинейной разработки, полной децентрализацией и возможностью эффективно работать с большими проектами.

Подход Git к хранению данных больше похож на набор снимков миниатюрной файловой системы. Каждый раз, когда вы сохраняете состояние своего проекта в Git, система запоминает, как выглядит каждый файл в этот момент, и сохраняет ссылку на этот снимок.

Git бесплатная и open-source система. Это значит, что его можно бесплатно скачать и вносить любые изменения в исходный код. Небольшой и быстрый. Он выполняет все операции локально, что увеличивает его скорость. Кроме того, Git локально сохраняет весь репозиторий в небольшой файл без потери качества данных.

Резервное копирование. Git эффективен в хранении бэкапов, поэтому известно мало случаев, когда кто-то терял данные при использовании Git. Простое ветвление. В Git управление ветками реализовано гораздо проще и эффективнее.

GitHub - сервис онлайн-хостинга репозиториев, обладающий всеми функциями распределенного контроля версий и функциональностью управления исходным кодом - всё, что поддерживает Git и даже больше. Обычно он используется вместе с Git и даёт разработчикам возможность сохранять их код онлайн, а затем взаимодействовать с другими разработчиками в разных проектах.

Также GitHub может похвастаться контролем доступа, багтрекингом, управлением задачами и вики для каждого проекта. Цель GitHub содействовать взаимодействию разработчиков.

К проекту, загруженному на GitHub, можно получить доступ с помощью интерфейса командной строки Git и Git-команд. Также есть и другие функции, такие как документация, запросы на принятие изменений (pull requests), история коммитов, интеграция со множеством популярных сервисов, email-уведомления, эмодзи, графики, вложенные списки задач и т.д.

Git - это инструмент, позволяющий реализовать распределенную систему контроля версий, а **GitHub** - это сервис для проектов, использующих Git.

GitHub Pages - служба хостинга сайтов, которая позволяет вести персональные сайты, сайты организаций и сайты отдельных проектов GitHub. Публичные страницы пользователей, организаций и репозиториев, с бесплатным хостингом от GitHub с доменом github.io или с кастомным доменом.

1.4. Обоснование выбора среды разработки

Angular фреймворк - является наиболее зрелым из Frontend фреймворков, имеет хорошую поддержку с точки зрения участников и представляет собой полный пакет. Тем не менее, кривая обучения является

довольно крутой, и концепции развития в Angular могут оттолкнуть новых разработчиков. Angular - хороший выбор для компаний с большими командами и разработчиков, которые уже используют TypeScript.

1.5. Редактор кода Visual Studio Code

Visual Studio Code – кроссплатформенный редактор кода с поддержкой более 30 языков программирования и форматов файлов, а также обладающий рядом дополнительных, полезных возможностей. Быстрый и бесплатный редактор со множеством плагинов.

Visual Studio Code поддерживает локальное и удаленное Git хранилища. В контексте Visual Studio Code можно выполнить любую команду командной строки и просмотреть результаты работы прямо из среды разработки. Таким образом можно использовать внешние компиляторы, отладчики, средства тестирования и т.д.

Практическая часть

2.1. Установка необходимых программ и плагинов

Установка Node.js как обычный пакет. Для этого заходим на сайт https://nodejs.org и на главной странице скачиваем любую актуальную версию, нажав на соответствующую кнопку.

Установка Git. Имеется несколько способов установки. Официальная сборка доступна для скачивания на официальном сайте Git. Переходим на страницу http://git-scm.com/download/win, и загрузка установщика запустится автоматически.

Установка Visual Studio Code https://code.visualstudio.com/download необходимо скачать текущую актуальную версию и установить всё по умолчанию. Для разработки на Angular нет необходимых плагинов, однако у меня имеется личный набор рекомендованных плагинов, список которые я периодически обновляю, скачать плагины можно по ссылке ниже: https://github.com/Domovikx/Visual-Studio-Code#visual-studio-code-vs-code-for-angular

- Angular 8 Snippets - https://marketplace.visualstudio.com/items?itemName=Mikael.Angular-BeastCode
- Angular Files - https://marketplace.visualstudio.com/items?itemName=alexiv.vscode-angular2-files
- Angular Language Service - https://marketplace.visualstudio.com/items?itemName=Angular.ng-template
- Auto Close Tag - https://marketplace.visualstudio.com/items?itemName=formulahendry.auto-close-tag
- Better Material Theme Darker High Contrast - https://marketplace.visualstudio.com/items?itemName=CrazyFluff.bettermaterialthemedarkerhighcontrast
- Bracket Pair Colorizer - https://marketplace.visualstudio.com/items?itemName=CoenraadS.bracket-pair-colorizer

- Code Spell Checker - https://marketplace.visualstudio.com/items?itemName=streetsidesoftware.code-spell-checker
- Git Graph - https://marketplace.visualstudio.com/items?itemName=mhutchie.git-graph
- GitLens - https://marketplace.visualstudio.com/items?itemName=eamodio.gitlens
- JavaScript (ES6) code snippets - https://marketplace.visualstudio.com/items?itemName=xabikos.JavaScriptSnippets
- Log Wrapper - https://marketplace.visualstudio.com/items?itemName=chrisvltn.log-wrapper-for-vscode
- Prettier - Code formatter - https://marketplace.visualstudio.com/items?itemName=esbenp.prettier-vscode
- vscode-angular-html - https://marketplace.visualstudio.com/items?itemName=ghaschel.vscode-angular-html
- vscode-icons - https://marketplace.visualstudio.com/items?itemName=vscode-icons-team.vscode-icons

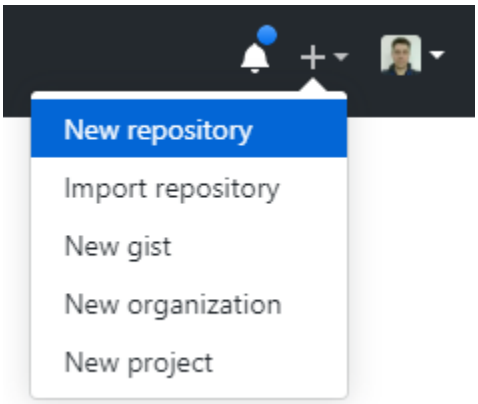
Установка Angular CLI - это npm-модуль, реализующий интерфейс командной строки для создания, разработки и поддержки Angular приложений. В системе он должен быть установлен глобально.

```
npm i @angular/cli -g
```

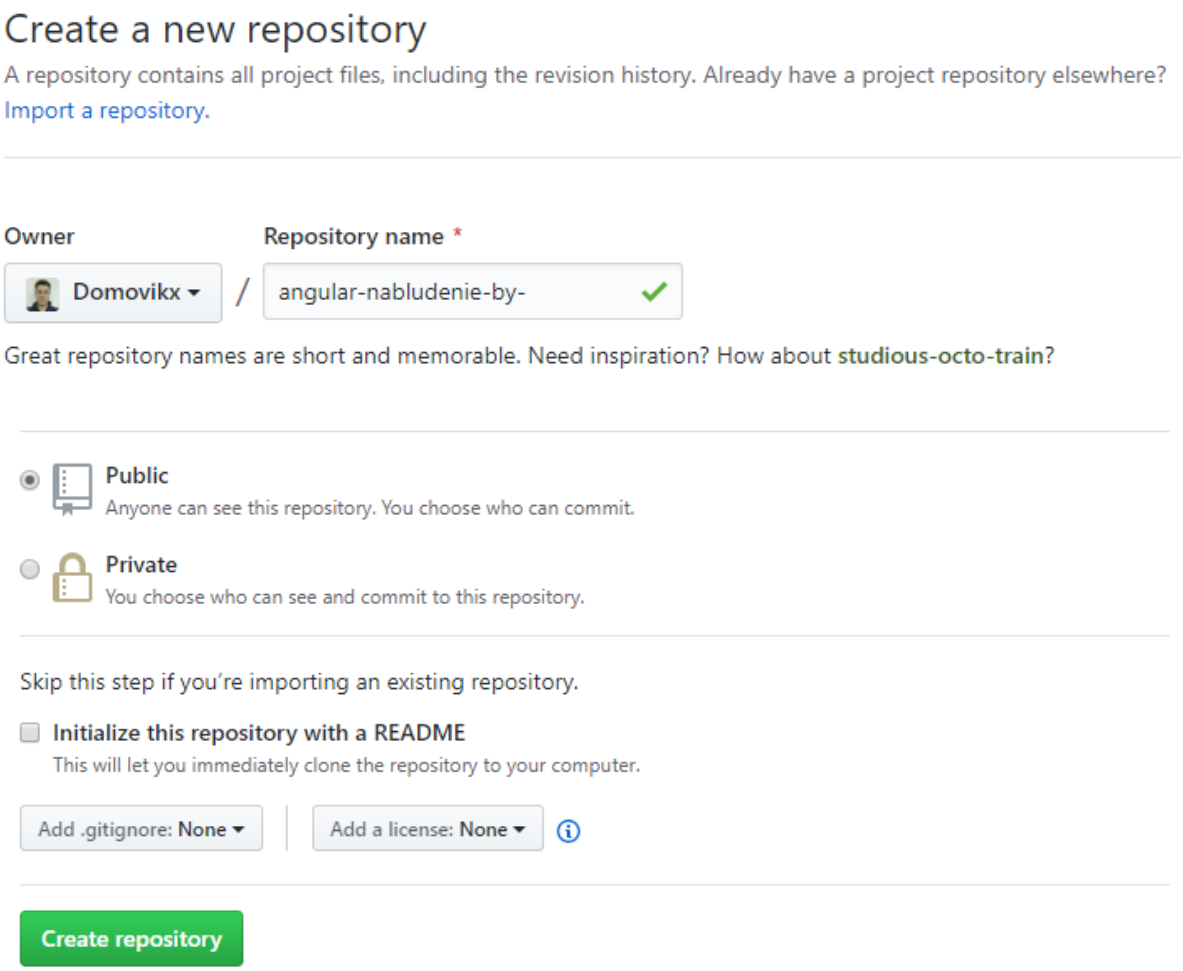
в случае необходимости есть подробное руководство:
https://xsltdev.ru/angular/tutorial/setup-and-configuration/
https://xsltdev.ru/angular/tutorial/angular-cli/

2.2. Создание репозитория на GitHub

Для создания репозитория требуется предварительная регистрируемся на https://github.com/.



Далее репозиторий создается простыми и понятными действиями, скриншоты прилагаются.



После этого репозиторий на GitHub готов к работе. При необходимости, для создания репозитория, можно воспользоваться простым и понятным руководством для начинающих:

https://zencod.ru/articles/first-step-git.

2.3. Генерация шаблона проекта (Angular CLI)

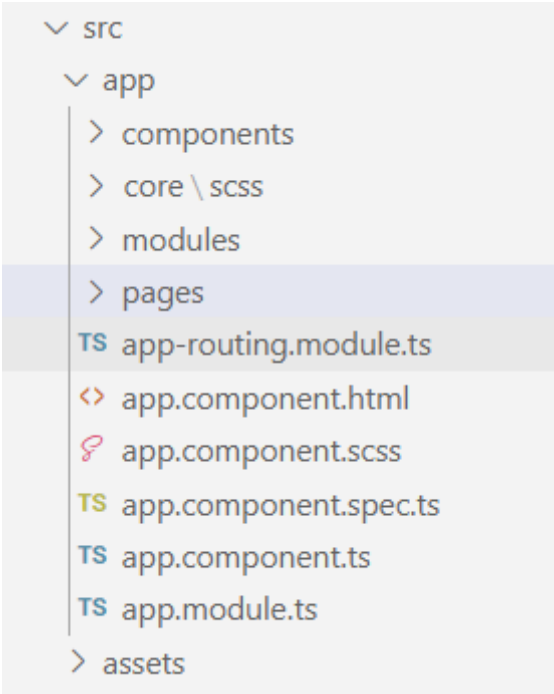
Для создания нового проекта с помощью **Angular CLI** нужно выполнить команду в выбранной папке, например, в созданной папке Git:

```
ng new angular-nabludenie-by
```

Команда ng new создает базовое приложение в папке angular-nabludenie-by. Подробное руководство по Angular CLI находится здесь: https://xsltdev.ru/angular/tutorial/setup-and-configuration/

2.4. Разработка структуры проекта

По умолчанию структура проекта (расположение файлов и папок) предоставляется самим фреймворком и формируется с помощью Angular CLI.



Главный модуль проекта - **src\app\app.module.ts** , головной модуль, в свою очередь, состоим из других модулей и/или компонентов.

Модуль - является своеобразной сборочной единицей и может собирать в себе другие модули и компоненты.

Компонент - это часть интерфейса приложения с собственной логикой. Вся видимая часть Angular App реализуется с помощью компонентов, поэтому часто можно услышать, что архитектура Angular компонентная.

2.5. Настройка роутинга страниц

В Angular маршрутизация представляет собой переход от одного представления (шаблона) к другому в зависимости от заданного URL. Причем навигация может осуществляться и внутри представления.

Навигация в Angular приложениях происходит без перезагрузки страницы. Ключевая роль в формировании URL принадлежит тегу <base>, указывающему путь к приложению относительно расположения файла index.html. Если index.html располагается в директории клиентского приложения, то тег должен быть записан следующим образом.
<base href="/" />

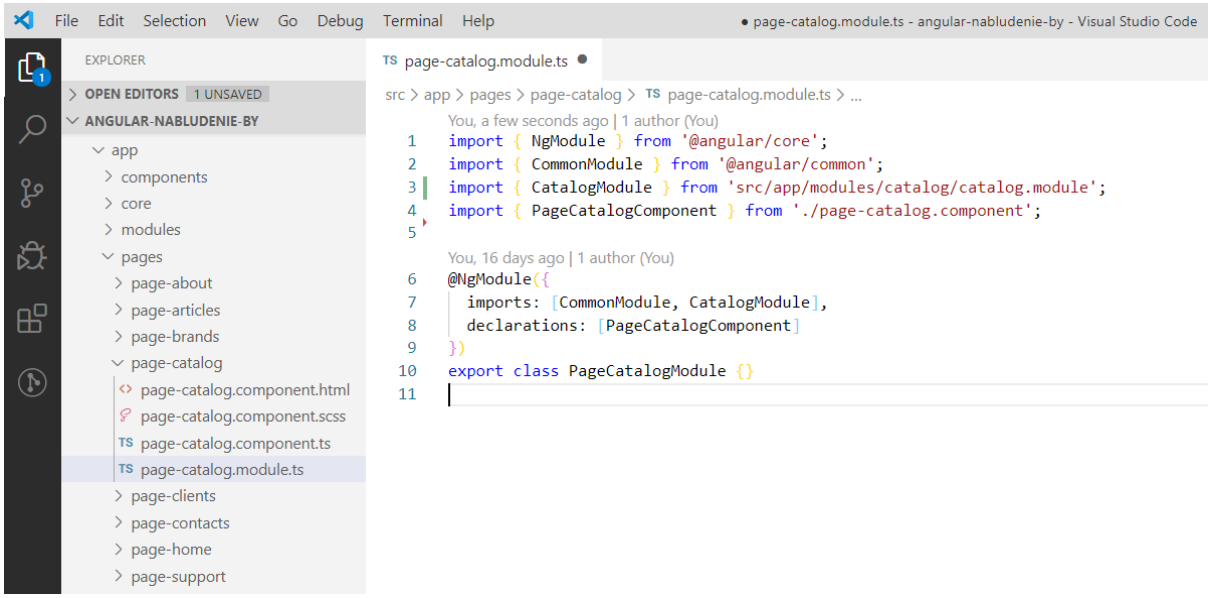
За организацию маршрутизации в Angular отвечает модуль RouterModule библиотеки @angular/router. URL организуются в специальные модули и определяются для каждого отдельного модуля приложения. Более подробно про роутинг и маршрутизацию, следуют читать здесь: **https://xsltdev.ru/angular/tutorial/angular-routing-basics/**

Основные файлы в которых настраивается маршрутизация:
src\app\app-routing.module.ts - основной роутинг модуль
src\app\app.component.html - добавляем тег <router-outlet></router-outlet>
src\app\app.module.ts - импортируем модуль AppRoutingModule

После настройки маршрутизации можно создавать рабочие ссылки. Ссылка на главную страницу выглядит так:

```
<a [routerLink]="['/']">
ссылка на каталог:
<a class="navigation__links" [routerLink]="['/catalog']">КАТАЛОГ</a>
```

Страницы проекта представляют собой модули состоящие из модулей и/или компонентов.



2.6. Добавление модуля карты

Для подключения карты был использован Angular8-yandex-maps модуль: https://www.npmjs.com/package/angular8-yandex-maps

А в качестве центра карты были установлены координаты здания УО«ВГТУ»

```
<angular-yandex-map
    [center]="[55.178214, 30.234987]"
    [zoom]="16"
    [clusterer]="clusterer"
>
<angular-yandex-placemark
      [geometry]="[55.178214, 30.234987]"
      [properties]="placemarkPropertiesVSTU"
      [options]="placemarkOptions">
</angular-yandex-placemark>
```

2.7. Добавление SVG спрайта

SVG-спрайт, представляет собой один корневой элемент svg, внутри которого создаются области **symbol**, внутри которых содержится код **svg-иконки**. Каждому элементу **symbol** назначается уникальный **id**, по которому к нему в дальнейшем можно будет обратиться в документе.

Symbol создает шаблон из любых **svg-объектов**, который в дальнейшем можно многократно использовать с помощью тэга **use**. Контент, расположенный внутри **symbol**, не отображается на странице.

Вывести нужную иконку можно с помощью тэга **use** и **id** иконки:

```
<svg class="mail-svg-icon">
  <use xlink:href="#mail"></use>
</svg>
```

Возможно управлять стилями иконки

```
.mail-svg-icon{
    display: block;
    width: 150px;
    height: 150px;
    fill: #7db958;
 }
```

SVG sprite проекта расположен здесь: **src\assets\svg\sprite.svg**
Подробнее про создание SVG спрайта можно прочитать здесь: http://dreamhelg.ru/2017/02/symbol-svg-sprite-detail-guide/

2.7. Адаптивный дизайн

Адаптивный веб-дизайн (англ. Adaptive Web Design) - дизайн веб-страниц, обеспечивающий правильное отображение сайта на различных устройствах, подключенных к интернету, и динамически подстраивающийся под заданные размеры окна браузера.

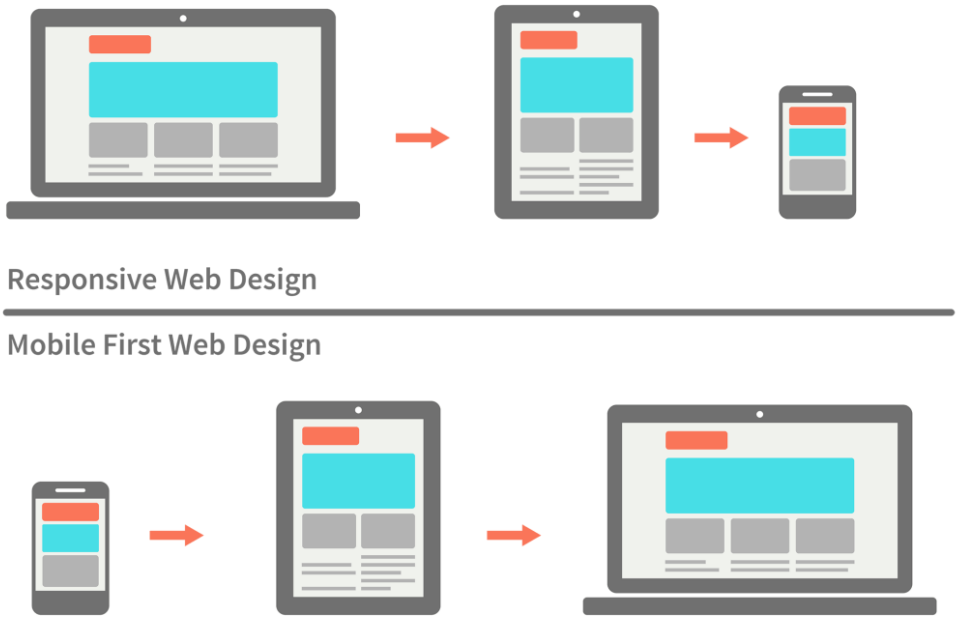
Целью адаптивного веб-дизайна является универсальность отображения содержимого веб-сайта для различных устройств. Для того, чтобы веб-сайт был удобно просматриваемым с устройств форматов и с экранами различных разрешений, по технологии адаптивного веб-дизайна не нужно создавать отдельные версии веб-сайта для отдельных видов устройств. Один сайт может работать на смартфоне, планшете, ноутбуке и телевизоре с выходом в интернет, то есть на всем спектре устройств.

Медиазапросы (media queries) - это правила CSS, которые позволяют управлять стилями элементов в зависимости от значений технических параметров устройств. Иными словами, это конструкции, которые

позволяют определять на основании некоторых условий какие стили необходимо использовать на веб-странице, а какие нет. Медиа запросы позволяют добавлять и использовать адаптивность сайта.

Существует два основных подхода при добавлении адаптивности сайта **desktop first** и **mobile first**.

В текущем проекте использовался подход **desktop first**. На практике это означает последовательность применения стилей, пример список media запросов для фреймворка Bootstrap 4:



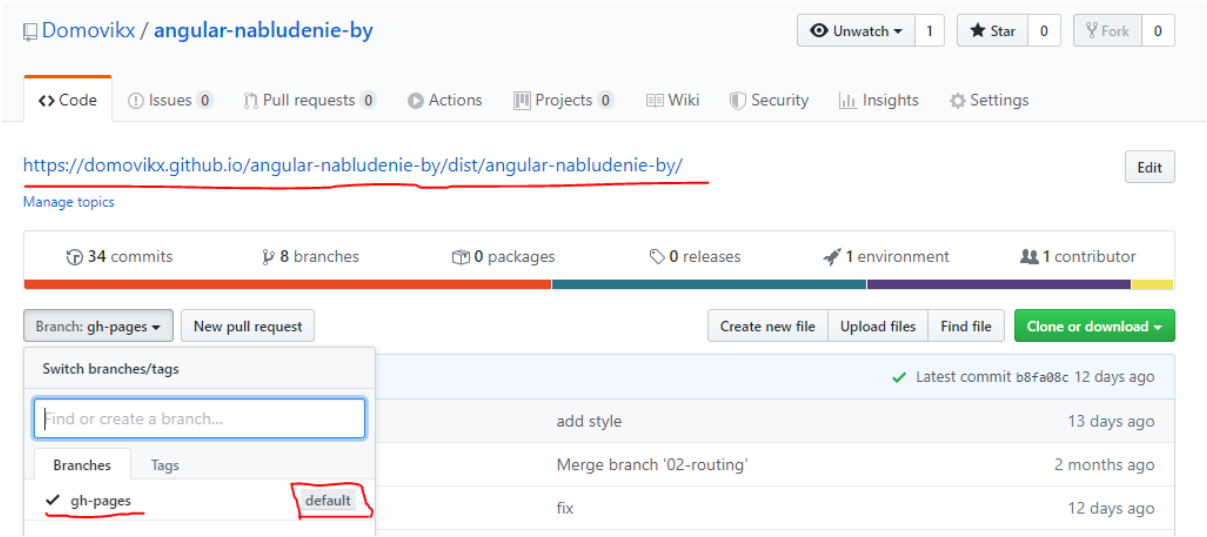
xl-размер (>=1200px) CSS для >=1200px
lg-размер (<=1199px)
@media (max-width: 1199px) { CSS для ширины от 992px до 1199px }
md-размер (<=991px)
@media (max-width: 991px) { CSS для ширины от 768px до 991px }
sm-размер (<=768px)
@media (max-width: 767px) { CSS для ширины от 576px до 767px }
xs-размер (<=575px)
@media (max-width: 575px) { CSS для ширины до 575px (включительно) }

2.8. Публикация проекта на gh-pages

Для публикации проекта будем использовать стандартные возможности GitHub, а именно gh-pages branch. Для разворота проекта выполняем команду:

ng build --prod --build-optimizer --base-href ./
после этого проект будет собран в папке - /dist.

Любым удобным способом переносим проект в **gh-pages** ветку, достаточно переместить в эту ветку только содержимое папки /dist, но в нашем случае мы перенесем весь проект в **gh-pages** и установим эту ветку - веткой по умолчанию.



Теперь, для удобства создаем ссылку на проект и проверяем его работу в интернет:
https://domovikx.github.io/angular-nabludenie-by/dist/angular-nabludenie-by/

ЗАКЛЮЧЕНИЕ

В результате курсового проекта на фреймворке Angular 8 был реализован веб-интерфейс интернет магазина, образец-прототип был выбран случайным образом (https://nabludenie.by/).

Разработка проекта велась на публичном GitHub репозитории:
https://github.com/Domovikx/angular-nabludenie-by

Проект развернут и доступен по адресу:
https://domovikx.github.io/angular-nabludenie-by/dist/angular-nabludenie-by/

Список использованных источников

1. https://xsltdev.ru/angular/tutorial/angular-cli/
2. https://myrusakov.ru/what-is-spa.html
3. https://wezom.com.ua/blog/chto-takoe-spa-prilozheniya
4. https://semantica.in/blog/veb-interfejs.html
5. https://developer.mozilla.org/ru/docs/Web/HTML
6. https://developer.mozilla.org/ru/docs/Web/CSS
7. https://html5book.ru/osnovy-css
8. https://sass-scss.ru/documentation/sintaksis/
9. https://learn.javascript.ru/intro
10. https://ru.wikipedia.org/wiki/TypeScript
11. https://xsltdev.ru/angular/guide/intro/start/
12. https://dev-gang.ru/article/vvedenie-v-angular-cli-bwnjvpdhhu/
13. https://tproger.ru/translations/difference-between-git-and-github/
14. https://ru.hexlet.io/courses/html/lessons/github/theory_unit
15. http://markshevchenko.pro/articles/github-pages/jekyll/
16. http://prgssr.ru/documentation/22_github_pages
17. https://habr.com/ru/company/microsoft/blog/268837/
18. https://zencod.ru/articles/first-step-git
19. https://xsltdev.ru/angular/tutorial/angular-routing-basics/
20. http://dreamhelg.ru/2017/02/symbol-svg-sprite-detail-guide/
21. https://itchief.ru/lessons/html-and-css/css-media-queries
22. https://medium.com/the-code-times/why-you-should-choose-angular-for-back-end-projects-b32e74a5c9cb
23. https://www.insales.com.ua/blogs/blog/interfeys-onlayn-magazina

Содержание электронного носителя

- angular-nabludenie-by.7z
- is_17-course_work-ivanouski_ilya.docx