МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТЮМЕНСКИЙ индустриальный университет»

Институт сервиса и отраслевого управления

Кафедра Бизнес-информатики и математики

КУРСОВАЯ РАБОТА

Дисциплина: «Интеллектуальные системы и их проектирование»

Тема: «Разработка игры на движке Godot с процедурной генерацией в реальном времени»

Выполнил:

Студент 4 курса группы РИСб-20-1

направления 45.03.04 «Интеллектуальные

системы в гуманитарной сфере»

Герасимов Е.В.

Руководитель:

Доцент, канд. пед. наук

Спирин И.С.

Тюмень, 2023 г.

Содержание

Введение 3

1 Теоретико-методологические аспекты разработки веб-сайта 4

1.1 Основные понятия 4

1.2 Этапы создания веб-сайтов 6

1.3 Базовые технологии Web-страниц 9

1.4 Технологии Web-программирования 11

2 Проектирование и разработка веб-сайта 13

2.1 Бриф 13

2.2 Структура сайта 14

2.3 Разработка структуры сайта на языках HTML5, CSS3 15

2.4 Добавление динамических элементов с использованием JavaScript и jQuery 22

Заключение 24

# ВВЕДЕНИЕ

В настоящее время игры являются одним из самых популярных видов развлечения, и разработка игр становится все более востребованной сферой. Разработка игры требует тщательного планирования, решения проблем и творческого мышления.

Актуальность и отличительная черта этого проекта – процедурная генерация уровня. Это ускоряет процесс создания, повышает разнообразие и реиграбельность игры.

В данной курсовой работе будет рассмотрена разработка игры на движке Godot с процедурной генерацией в реальном времени. Будут рассмотрены основные принципы процедурной генерации, а также способы ее реализации на практике с использованием движка Godot.

Цель – Изучение принципов процедурной генерации и создание игры на движке Godot.

Объект курсовой – игра на движке Godot.

Предмет курсовой – Процесс разработки игры с процедурной генерацией

Задачи:

* Изучить возможности для создания игр
* Проанализировать принципы процедурной генерации
* Создать дизайн-документ
* Собрать необходимый список программ для работы
* Разработать первую версию игры

# Теоретические аспекты разработки видеоигр

## Основные понятия

Компьютерная игра – разновидность игр, в которых компьютер выполняет роль ведущего или партнера по игре, а возникающие по ходу игры игровые ситуации воспроизводятся на экране дисплея или телевизора.

Компьютерные игры бывают разных жанров, таких как:

* Стратегия;
* Экшен;
* Квест;
* Симулятор;
* Шутер;
* Защита замка;
* Roguelike;
* Платформер.

Платформер – жанр компьютерных игр, в которых основу игрового процесса составляют прыжки.

Сайд-скроллер – компьютерная игра, в которой игрок наблюдает за игровым процессом посредством расположенной сбоку виртуальной камеры

Графический интерфейс – система средств для взаимодействия пользователя с компьютером, основанная на представлении доступных пользователю функций в виде графических компонентов (окон, кнопок, меню, списков и т. п.).

Движок (игровой движок, game engine) – программное ядро комплексной программной системы (игры), содержащее базовую функциональность игры, но, при этом, не включающее код, специфичный для конкретной игры.

Дизайн-документ — это детальное описание разрабатываемой компьютерной игры.

## Этапы создания веб-сайтов

Основными этапами создания веб-сайта являются:

1. Постановка задачи. Наиболее частыми задачами являются:
   1. Узнаваемость, привлечение новых клиентов и партнеров.
   2. Презентация компании, её деятельности или услуг.
   3. Продажа товаров или продуктов.
   4. Привлечение целевой аудитории и перенаправление на друге каналы.
   5. Организация рекламной кампании конкретной продукции или бренда.
2. Проработка и утверждение технического задания:
   1. Описание проекта.
   2. Цели и задачи.
   3. Глоссарий терминов.
   4. Применяемые технологии и требования к хостингу.
   5. Подробная структура.
   6. Требования к дизайну.
   7. Требования к вёрстке.
   8. Функциональные возможности.
   9. Описание внутренних страниц.
   10. Графический и текстовый контент.
   11. Сроки сдачи проекта.
3. Разработка макета дизайна сайта.
4. Верстка страниц.
5. Программирование функциональной части.
   1. Back-end.
   2. Front-end.
6. Продвижение сайтов и наполнение контентом.
7. Тестирование и последующая настройка.
   1. Подготовительные работы.
   2. Функциональное тестирование.
   3. Тестирование верстки.
   4. Юзабилити тестирование.
   5. Тестирование безопасности.
   6. Тестирование производительности.
8. Внесение правок.

Также, важным этапом является SEO-продвижение сайта. Данный этап идет после всех остальных и продолжается вплоть до закрытия сайта.

SEO-продвижение это комплекс мер, направленных на увеличение видимости сайта в поисковых системах по определенным запросам пользователей. Существует условно два направления методов оптимизации, но только при всесторонней оптимизации сайт действительно может оказаться в топ выдачи поисковых систем.

Внутренняя оптимизация. Это то, что можно сделать на самом сайте, чтобы улучшить положение в поисковой выдаче. Это устранение проблем с индексированием, перелинковка, т.е. связывание страниц гиперссылками, написание качественного контента, улучшение загрузки, адаптация под мобильные устройства.

Внешняя оптимизация. Это то, что можно сделать вне сайта, чтобы повысить его релевантность. Поисковые системы рассматривают все сайты именно по этому критерию. Основные методы: регистрация в профильных каталогах, обмен ссылками, размещение статей на других ресурсах, представительство в социальных сетях. Это методы, которые будут помогать сайту занять первые позиции в выдаче Гугла и Яндекса.

Отдельное внимание нужно уделить контенту. Это подручное средство оптимизации сайта, включающее текстовое содержание, изображения, аудио- и видеоматериалы, flash-баннеры.

Контент-наполнение предопределяет, насколько успешным будет продвижение ресурса среди пользователей. Поэтому требования к содержимому включают уникальность текстов, полезность, грамотную подачу и оформление, соответствие запросам поисковых систем.

## Базовые технологии Web-страниц

Основой любой веб-страницы являются:

HTML – стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере.

Язык гипертекстовой разметки HTML был разработан британским учёным Тимом Бернерсом-Ли приблизительно в 1986—1991 годах в стенах ЦЕРНа в Женеве в Швейцарии. HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области вёрстки. HTML успешно справлялся с проблемой сложности SGML путём определения небольшого набора структурных и семантических элементов – дескрипторов. Дескрипторы также часто называют «тегами». С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа, в HTML внесена поддержка гипертекста. Мультимедийные возможности были добавлены позже.

Первым общедоступным описанием HTML был документ «Теги HTML», впервые упомянутый в Интернете Тимом Бернерсом-Ли в конце 1991 года,. В нём описываются 18 элементов, составляющих первоначальный, относительно простой дизайн HTML. За исключением тега гиперссылки, на них сильно повлиял SGMLguid, внутренний формат документации, основанный на стандартном обобщенном языке разметки (SGML), в CERN. Одиннадцать из этих элементов всё ещё существуют в HTML 4.

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). В идеале, текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью (цветной экран современного компьютера, монохромный экран органайзера, ограниченный по размерам экран мобильного телефона или устройства и программы голосового воспроизведения текстов). Однако современное применение HTML очень далеко от его изначальной задачи. Например, тег <table> предназначен для создания в документах таблиц, но иногда используется и для оформления размещения элементов на странице. С течением времени основная идея платформонезависимости языка HTML была принесена в жертву современным потребностям в мультимедийном и графическом оформлении.

CSS – формальный язык декорирования и описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки (HTML).

Эта одна из широкого спектра технологий, одобренных консорциумом W3C и получивших общее название «стандарты Web». В 1990-х годах стала ясна необходимость стандартизировать Web, создать какие-то единые правила, по которым программисты и веб-дизайнеры проектировали бы сайты. Так появились языки HTML 4.01 и XHTML, и стандарт CSS.

В начале 1990-х различные браузеры имели свои стили для отображения веб-страниц. HTML развивался очень быстро и был способен удовлетворить все существовавшие на тот момент потребности по оформлению информации, поэтому CSS не получил тогда широкого признания.

Термин «каскадные таблицы стилей» был предложен Хоконом Ли в 1994 году. Совместно с Бертом Босом он стал развивать CSS.

В отличие от многих существовавших на тот момент языков стиля, CSS использует наследование от родителя к потомку, поэтому разработчик может определить разные стили, основываясь на уже определённых ранее стилях.

В середине 1990-х Консорциум Всемирной паутины (W3C) стал проявлять интерес к CSS, и в декабре 1996 года была издана рекомендация CSS1.

## Технологии Web-программирования

Основной технологией веб-программирования считается JavaScript и различные фреймворки на его основе.

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

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты:

1. Динамическая типизация – приём, используемый в языках программирования и языках спецификации, при котором переменная связывается с типом в момент присваивания значения, а не в момент объявления переменной.
2. Слабая типизация – характеристика языка программирования, означающая слабые или отсутствующие ограничения на приведение типов при присваиваниях и операциях.
3. Автоматическое управление памятью – система, предотвращающая накопление неиспользуемых объектов в оперативной памяти устройства.
4. Прототипное программирование – стиль объектно-ориентированного программирования, при котором отсутствует понятие класса, а наследование производится путём клонирования существующего экземпляра объекта – прототипа.
5. Функции как объекты первого класса. Объектами первого класса в контексте конкретного языка программирования называются элементы, которые могут быть переданы как параметр, возвращены из функции, присвоены переменной.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

jQuery – набор функций JavaScript, фокусирующийся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX.

Точно так же, как CSS отделяет визуализацию от структуры HTML, JQuery отделяет поведение от структуры HTML. Например, вместо прямого указания на обработчик события нажатия кнопки управление передаётся JQuery, которая идентифицирует кнопки и затем преобразует его в обработчик события клика. Такое разделение поведения и структуры также называется принципом ненавязчивого JavaScript.

Библиотека jQuery содержит функциональность, полезную для максимально широкого круга задач. Тем не менее, разработчиками библиотеки не ставилась задача совмещения в jQuery функций, которые подошли бы всюду, поскольку это привело бы к большому коду, большая часть которого не востребована. Поэтому была реализована архитектура компактного универсального ядра библиотеки и плагинов. Это позволяет собрать для ресурса именно ту JavaScript-функциональность, которая на нём была бы востребована. На официальном сайте «https://plugins.jquery.com/» существует более тысячи различных плагинов, часть из которых разработана командой, которая разработала и сам jQuery.

# Проектирование и разработка веб-сайта

## Бриф

Разработка любого сайта начинается либо с технического задания, либо с брифа. Ввиду того, что данный сайт представляет из себя относительно небольшой проект, выполняемый одним человеком для самого себя, было решено составить краткий бриф.

Таблица 2.1 – Бриф

|  |  |
| --- | --- |
| Вопрос | Ответ |
| Укажите общую информацию о заказчике. | Никитин Максим Денисович, студент, программист. |
| Кто ваша целевая аудитория? | Работодатели и частные лица, которым необходим программист в ценовом диапазоне <15 тыс. руб. |
| Кто ваши конкуренты? | Другие разработчики и программисты |
| Подробности заказа. | Есть ссылка на гитхаб, итч, вконтакте и почту. Также есть фотографию, которую хотелось бы видеть на странице.  Основным наполнением должны быть скриншоты уже завершенных проектов и общая информация обо мне. |
| Общие пожелания. | Планируется использовать сайт по его прямому назначению, как визитку. Прикреплять его в резюме или же отправлять заинтересованным людям. |
| Пожелания к дизайну. | Оформление должно быть стильным, минималистичным, желательно в темных тонах, отражающих профессионализм. |
| Дополнительная информация. | Отсутствует |

Исходя из данного брифа можно было приступать к разработке структуры и выбору стилистических решений для данного сайта.

## Структура сайта

Для удобства визуализации и редактирования, структура сайта была создана в бесплатной программе xmind в виде древа. Всего было решено создать 5 разделов, содержащих самую необходимую информацию.

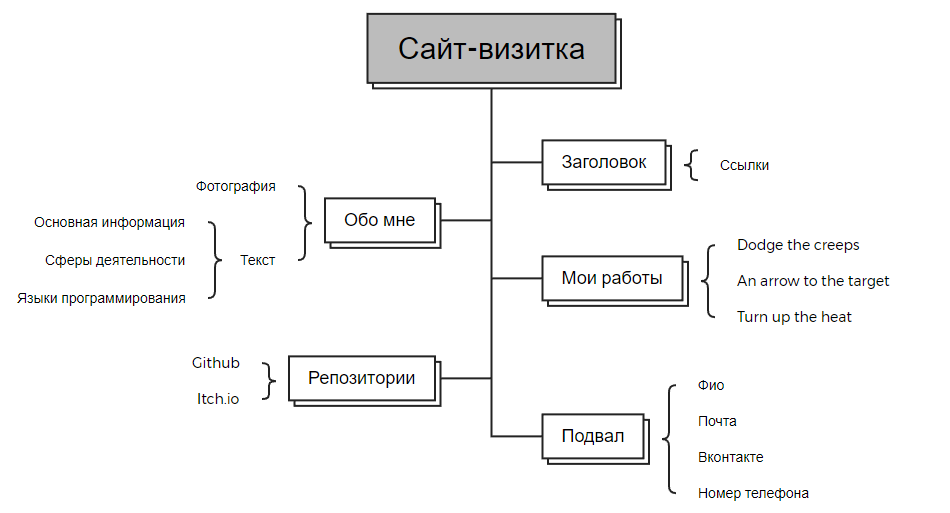


Рисунок 2.1 - Структура сайта-визитки

Данный сайт должен иметь небольшой объем, но при этом иметь всю важную информацию на виду, тем самым как можно эффективнее цепляя внимание потенциального клиента.

## Разработка структуры сайта на языках HTML5, CSS3

Первым делом, как и при создании любого другого сайта, оформляется раздел <Head>, в котором прописываются основные настройки и источники для сайта.

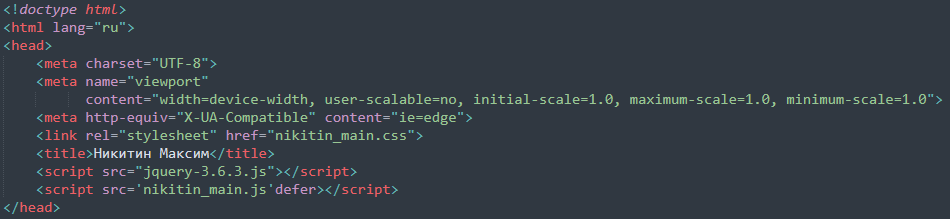


Рисунок 2.2 – HTML код раздела <Head>

Данный код отвечает за то, какой язык, набор символов, стиль, код javascript и так далее будут использоваться на странице.

Далее, в CSS были оформлены два основных селектора: body и container. Body распространяется на всю страницу и назначает стандартные значения некоторых характеристик. Container же используется для сепарации разделов сайта и настраивает их размер и отступы.

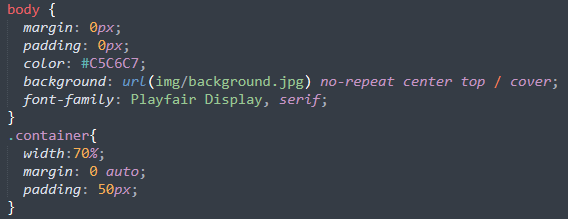


Рисунок 2.3 – CSS код для body и container

После этих приготовлений началась работа над основными разделами сайта по порядку, начиная с заголовка и ссылок на другие разделы. Их имена придумываются заранее в целях избежать конфуза позднее. Для разделов были выбраны id с названиями:

* Заголовок – «header»
* Обо мне – «personal\_info»
* Мои работы – «screenshots»
* Репозитории – «repositories»
* Подвал – «footer»

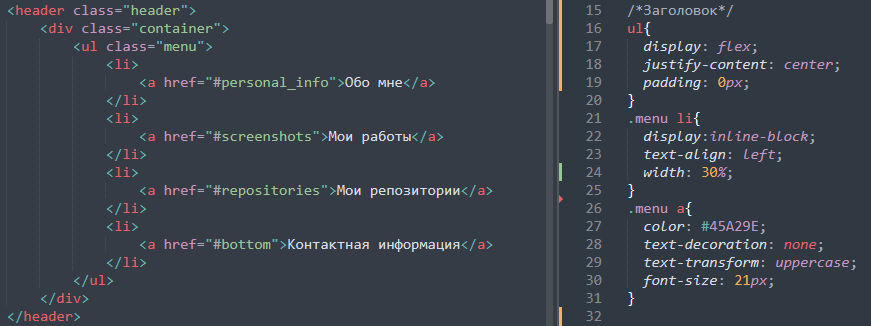


Рисунок 2.4 – Html код раздела «Заголовок» и соответствующий ему CSS код

Данный код выравнивает элементы списка <ul>, убирает подчеркивание и выделение ссылок, на которые пользователь уже кликнул ранее, а также выставляет расстояние между ссылками.

Следующий раздел состоит из трех основных элементов: основного контейнера, ограничивающего высоту раздела, двух колонн, одна из которых содержит только фотографию, и вторая, которая в будущем будет содержать три элемента аккордеона. Классы были созданы заранее для удобства добавления функционала в будущем.

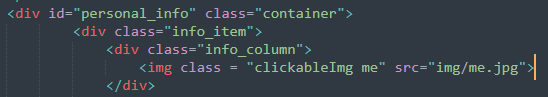


Рисунок 2.5 – Объявление раздела и первой колонны

Каждый аккордеон элемент состоит из нескольких элементов в данной иерархии:

1. accordion\_\_item – основа.
   1. accordion\_\_title – заголовок.
      1. accordion\_\_arrow – изображение стрелки, часть заголовка.
   2. accordion\_\_content – основной текст, спрятан по умолчанию.

Вторая колонка также имеет фиксированную ширину, с целью избежать искажения заголовка элементов аккордеона.

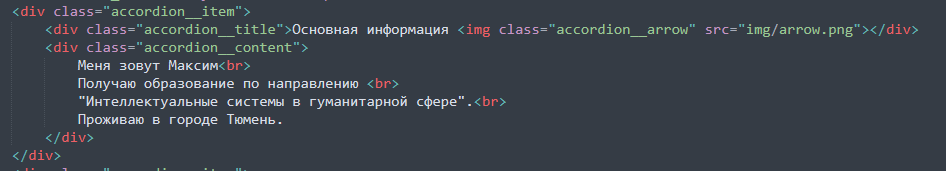


Рисунок 2.6 – Один из аккордеон элементов

Также, для стрелки был создан селектор accordion\_\_rotate, имеющий лишь одно свойство – transform: scaleY(-1).

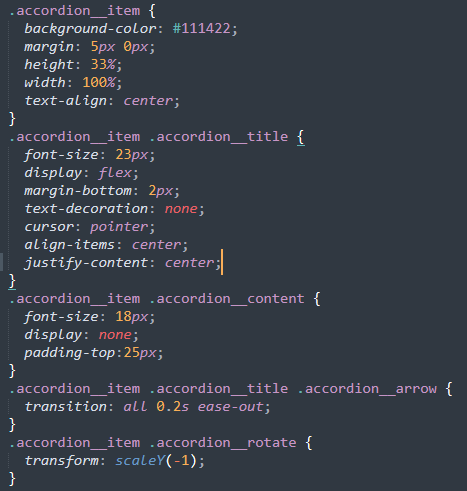


Рисунок 2.7 – CSS код элементов аккордеона

На этом раздел «Обо мне» был закончен и была начата разработка раздела «Мои работы». Сперва было необходимо создать модальное окно, которое открывалось бы при нажатии на картину.

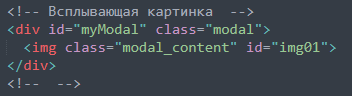


Рисунок 2.8 – Html код модального окна

Модальное окно должно было принимать в себя изображение, на которое кликнул пользователь, приблизить его и затемнить все вокруг себя. Зум, затемнение и анимация были выполнены в CSS.

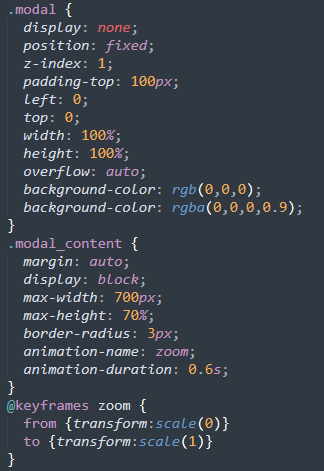


Рисунок 2.9 – CSS код модального окна

Остальной же раздел состоял из трех элементов: Заголовка изображений, находящегося по центру страницы и имеющего увеличенный шрифт, контейнера изображений, который выравнивал изображения внутри себя, а также обводил их по краю и самих изображений, которые добавляли отступы с краев, меняли цвет и курсор при наведении на себя мышью.

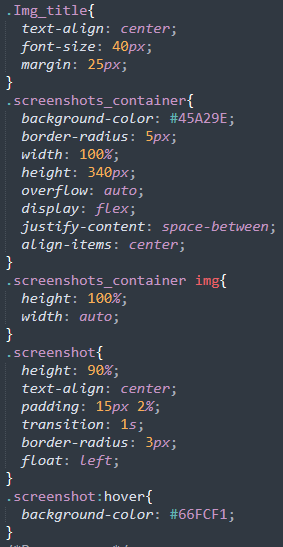


Рисунок 2.10 – CSS код блока скриншотов

В дополнение, всем изображениям был присвоен класс «clickableImg», использованный далее для добавления функционала кликабельности изображений.

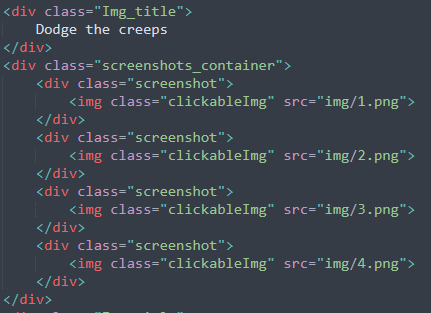


Рисунок 2.11 – Html код одного их блоков скриншотов

Предпоследним разделом сайта являются «Репозитории». Элементами этого раздела являются иконки сайтов и их названия, являющиеся кликабельными ссылками.

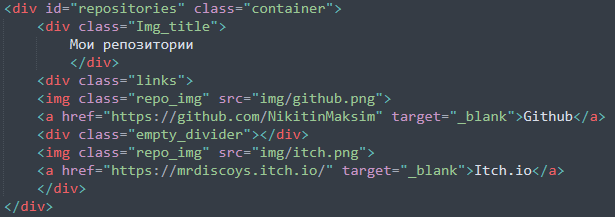


Рисунок 2.12 – Html код раздела «Репозитории»

Как и в случае с заголовком, было необходимо убрать выделение ссылок, на которые пользователь уже кликнул, а также установить одинаковый размер изображениям. Дополнительно, был создан пустой разделитель, добавляющий расстояние между двумя репозиториями.

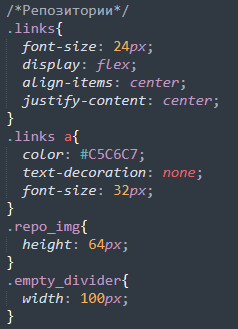


Рисунок 2.13 – CSS код раздела «Репозитории»

Последний раздел, «Подвал», состоял из трех разделителей и четырех колонн, содержащих всю необходимую контактную информацию.



Рисунок 2.14 – Html код раздела «Подвал»

Каждой колонне было присвоено изображение, соответствующее содержанию, а каждому такому изображению уникальный класс, необходимый далее для создания функции копирования ссылки одним кликом и общий класс «add\_pointer», меняющий курсор при наведении на изображение.

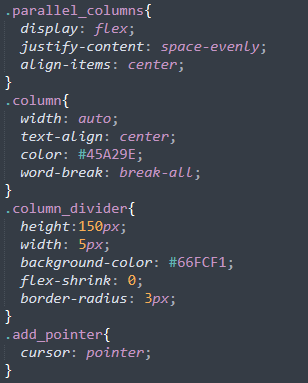


Рисунок 2.15 – CSS код раздела «Подвал»

## Добавление динамических элементов с использованием JavaScript и jQuery

Начиная сверху вниз, первым динамическим элементом был аккордеон. Используя библиотеку jQuery, создание аккордеона становится очень простым и быстрым, требуя лишь небольшое количество кода.



Рисунок 2.16 – JS код аккордеона

Данный код при нажатии на кнопку, если данный элемент активен, закрывает все остальные элементы и в любом случае меняет состояние своего содержания и стрелки.

Следующим динамическим элементом было модальное окно для изображений, принципом работы было изменение статуса display у модального окна и изменение источника изображения на то, которое было нажато пользователем.

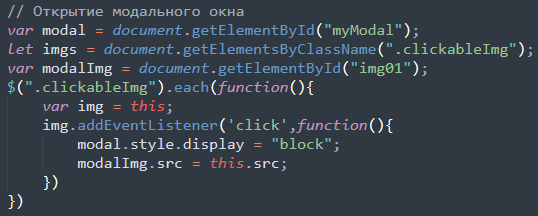


Рисунок 2.17 – JS код открытия модального окна

Для закрытия модального окна выполняется проверка, нажал ли пользователь на само модальное окно или же сбоку от него. Если клик выполнялся сбоку, то модальное окно закрывается.

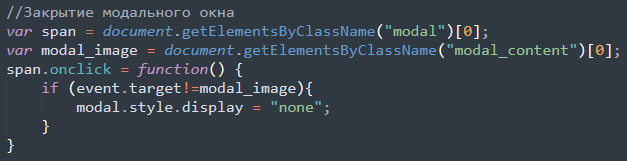


Рисунок 2.18 – JS код закрытия модального окна

Последняя функция имеет самый длинный код ввиду незнания полного объема функций библиотеки jQuery и написана на чистом JavaScript.

Принцип работы относительно прост, при нажатии на нужную картинку, сайт копирует почту/номер телефона или ссылку вконтакте в буфер обмена и предупреждает пользователя об этом.



Рисунок 2.19 – JS код функции копирования ссылок

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы был закреплен материал, изученный в процессе обучения, изучен дополнительный материал в свободном доступе, проанализирован рынок сайтов-визиток и разработан желаемый стиль будущего сайта-визитка личного пользования, а также выполнены все поставленные ранее задачи:

* Рассмотрены основные понятия веб-разработки
* Рассмотрены этапы создания веб-сайтов
* Рассмотрены технологии разработки веб-страниц
* Оформлено техническое задание на разработку веб-сайта
* Разработан дизайн веб-сайта
* Разработан веб-сайт

# Используемые источники

1. «jQuery tutorial» [Электронный ресурс]. Режим доступа: https://www.w3schools.com/jquery/ – (Дата обращения: 18.10.2022).
2. Онлайн-гайд «Изучение веб-разработки» [Электронный ресурс]. Режим доступа: https://developer.mozilla.org/ru/docs/Learn – (Дата обращения: 18.10.2022).
3. Веб-страница «Этапы создания сайта» [Электронный ресурс]. Режим доступа: https://webcase.com.ua/blog/iz-chego-sostoit-razrabotka-sajta/ – (Дата обращения: 18.10.2022).
4. Онлайн-гайд css и html [Электронный ресурс]. Режим доступа: https://html5book.ru/css-css3/ – (Дата обращения: 18.10.2022).
5. «Современный учебник JavaScript» [Электронный ресурс]. Режим доступа: https://learn.javascript.ru/ – (Дата обращения: 18.10.2022).
6. «Как создать цветовую схему для сайта» [Электронный ресурс]. Режим доступа: https://skillbox.ru/media/design/kak\_vybrat\_i\_sozdat\_tsvetovuyu\_skhemu\_dlya\_sayta/ – (Дата обращения: 18.10.2022).