**IT-Колледж “Сириус”**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ДОКЛАД**

по дисциплине “Введение в специальность”

на тему “Веб разработка”

Выполнил:  
Студент группы

1.9.7.2

Мукасеев Владислав Денисович

Принял:

Старший преподаватель  
Тенигин Альберт Андреевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IT-Колледж “Сириус”  
2022

# **Оглавление**

**[ЧТО ТАКОЕ ВЕБ РАЗРАБОТКА?](#_heading=h.macrtgvjew4f)** 3

**[FRONTEND DEVELOPMENT](#_heading=h.o6tfu3hs2uxh)** 3

[UX & UI дизайн.](#_heading=h.uayocwf3f6r) 4

[ВЁРСТКА](#_heading=h.aort5qt7yiy1) 7

[JAVASCRIPT](#_heading=h.ku9e0dnwiqc0) 12

**[BACKEND](#_heading=h.fr1v5itkf0us)** 13

[БАЗА ДАННЫХ](#_heading=h.aurq5xp9mqeg) 14

**[FULLSTACK](#_heading=h.akzc851om7mt)** 15

**[ВЫВОД](#_heading=h.e37micbou11i)** 16

**[Список использованной литературы](#_heading=h.2et92p0)** 18

# 

# **ЧТО ТАКОЕ ВЕБ РАЗРАБОТКА?**

**Веб разработка** - процесс создания сайтов или же веб приложений. Создание UX/UI концепта, вёрстка, разработка на серверной и клиентской сторонах описывает этот термин достаточно поверхностно. Если мы будем описывать целый процесс, то здесь нужно включить и тестировщиков, которые тестируют созданный концепт разработчиками, SEO оптимизацию, размещение на хостинге, его поддержка, обслуживание веб продукта, копирайтинг и им подобные. Разработка сайтов и веб приложений - не постановка изображений в нужных углах и на коленке построенная логика, а грамотное отображение серверной части за счёт клиентской, лёгкий доступ и удобное пользование концептом. Поскольку веб разработка довольно обширная тема, разработчики привыкли разделять её на три этапа, ответвления.

# **FRONTEND DEVELOPMENT**

**Frontend development** (клиентская разработка) - первый этап, о котором я начну рассказывать. Всё за то, что связано с пользовательским интерфейсом, отвечает Frontend разработчик. Грамотно отображённые кнопки, их различные поведения при определённых взаимодействиях, лёгкость в использовании, понятный и интересный визуальный концепт. Существуют некоторые стереотипы как об этих разработчиках, так и о других. Какой смысл в том, что ты ставишь блоки здесь, а этот здесь, за что вам платят деньги? Оказавшись перед пустым фреймом наедине, уверенность в собственных силах и словах значительно падает. Этот же стереотип присущ и самим фронтендерам касаемо Backend разработки На мой взгляд, люди которые больше предрасположены к гуманитарным наукам идут во Frontend, а другие, которым близки технические - Backend, про которую я расскажу потом. При поиске информации об ответвлениях веб разработки я не нашёл информации об одном очень важном направлении, который посчитал нужным включить. Из-за того что я предпочитаю эту сферу, считаю нужным внести свои некие идеи.

## **UX & UI дизайн**

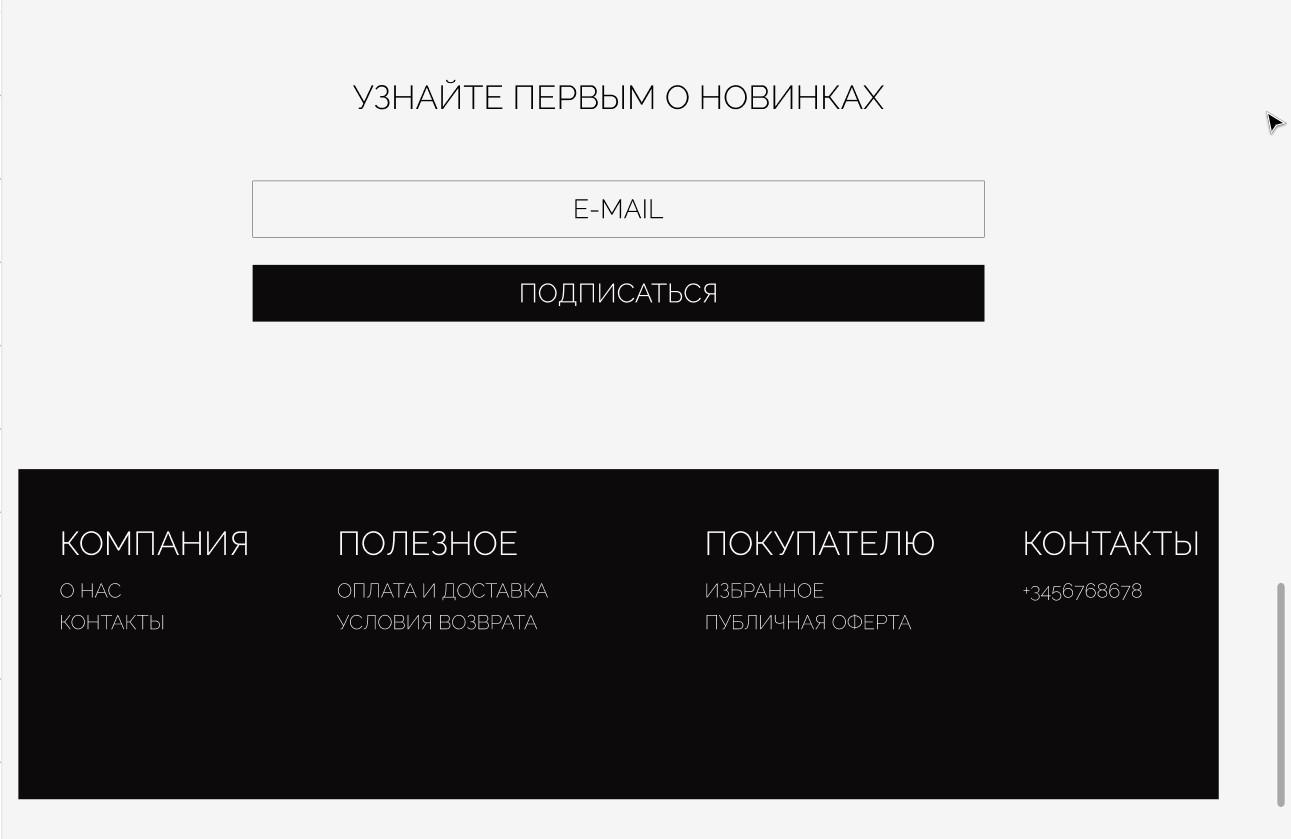
**UX** (англ. User eXperience - пользовательский опыт) ориентирован на логику продукта. Для него характерны подобные вопросы: получает ли пользователь то, за чем пришёл на наш сайт? Легко ли пользоваться концептом, который мы создали? Какова цель нашего продукта и какие проблемы посетителей мы можем помочь решить с помощью наших идей и их воплощений? Как мы облегчим путь к цели пользователей? Все похожие вопросы нужны для того, чтобы построить план разработки нового сайта или приложения. При этом, они помогают определить цели и задачи разработанного продукта, а соответственно и понять какого рода визуальный концепт будет отражать эту самую логику.

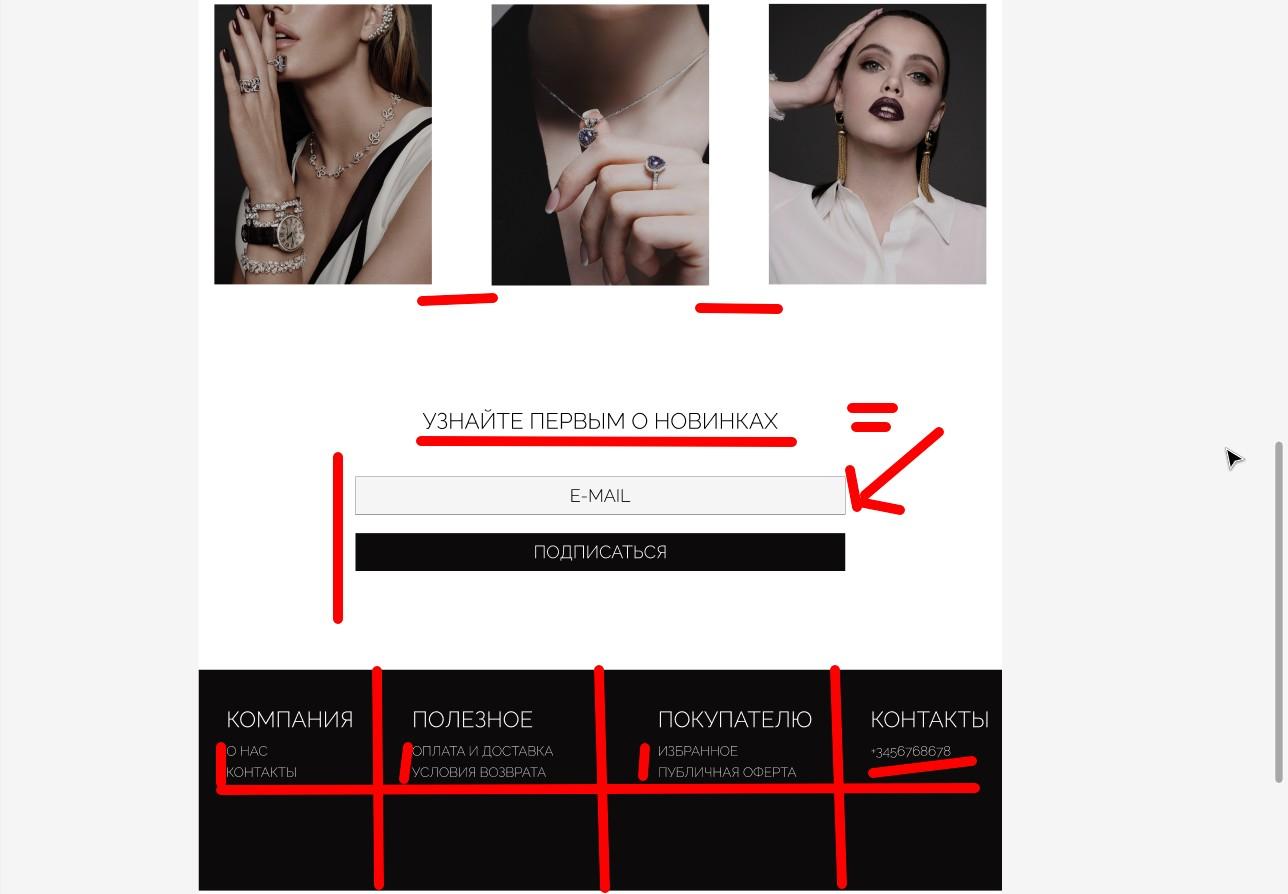
**UI** (англ. User Interface - пользовательский интерфейс) ориентирован на визуальную составляющую продукта. В чем собственно смысл разработанной логики, если визуально ничего не понятно? Замечаю за собой такую вещь, когда я стараюсь запомнить или выучить новую информацию, мне необходимо увидеть самому простые картинки, которые соответствуют содержанию или же наглядное объяснение. Наше восприятие информации больше всего ориентировано на глаза, а уши, осязание.. всё потом. Люди не могут увидеть созданную логику, ведь она ничем не закреплена, никакой визуальной составляющей. Пример сайта с визуальным ужасом:

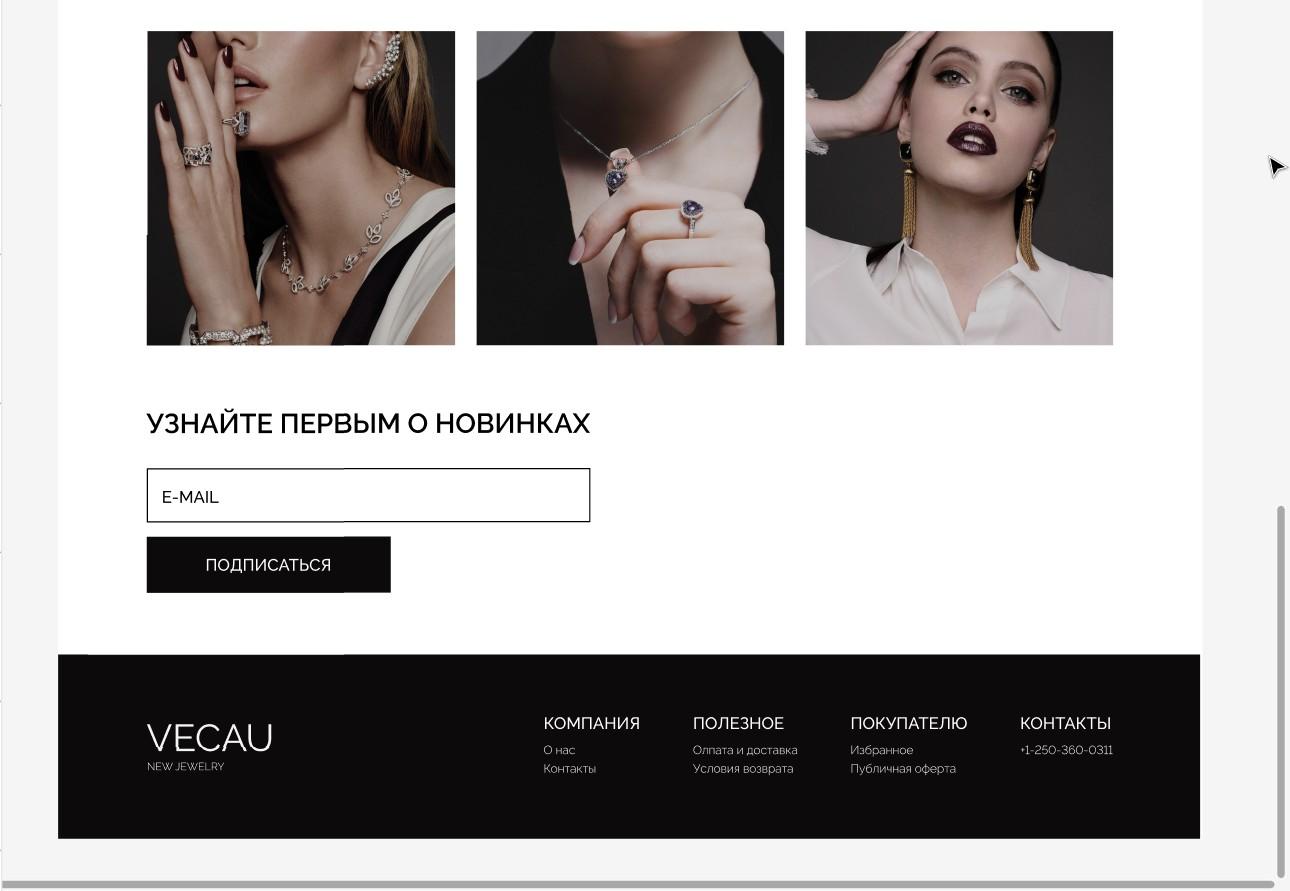


(рис. 1 <https://www.theworldsworstwebsiteever.com/> )

Почему пример настолько тошен, ведь чисто физически даже было бы стыдно сделать подобное.. Как раз таки задача создателя этого сайта была в том, чтобы показать начинающим дизайнерам как сайт НЕ должен выглядеть, то есть как визуальная часть может разрушить любое взаимодействие с пользователем, подорвать доверие и вовсе больше не обращаться за помощью к нашему продукту. Из моей практики и взаимодействием с начинающими дизайнерами хочу показать один пример, где потом я пометил ошибки, которые необходимо исправить и привёл свой вариант решения (весь скриншот было бы трудно разбирать и он не влез в его границы, поэтому прикрепил только часть проекта).







Есть разница? Подводя итог, UX полностью взаимодействуют с UI, нужно поддерживать баланс, в противном случае, могут получиться какие то из моих примеров, а то может и хуже [4].

## 

## **ВЁРСТКА**

Для того чтобы быть Frontend разработчиком, необходимо уметь **верстать**. То есть уметь структурировать элементы на странице сайта или веб приложения. Для этого нужно знать язык гипертекстовой разметки HTML и каскадных таблиц стилей CSS (его фрэймворки). Расскажу и приведу примеры из каждого перечисленного:

**HTML**

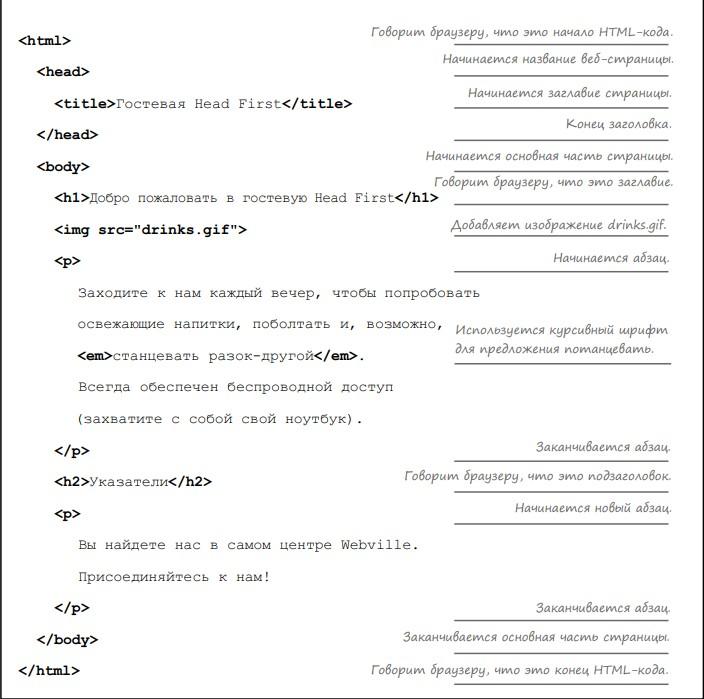
Единственное, что необходимо для того, чтобы успешно работать в Сети, — научиться говорить на ее специфическом языке: HyperText Markup Language (язык гипертекстовой разметки), или сокращённо **HTML**. [1]

● HyperText (гипертекст) — текст, который соединяет связующие между собой элементы.

● Markup (разметка) — руководство по стилю набора текста.

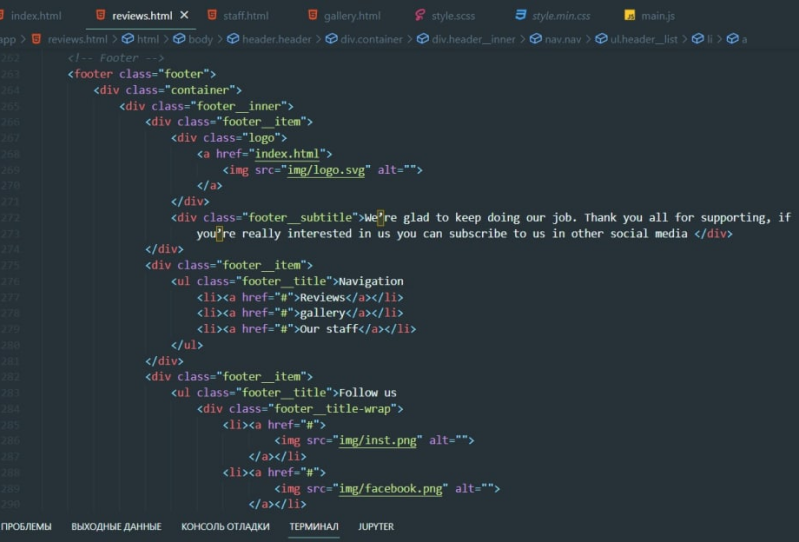
● Language (язык) — код, который компьютерная система использует для интерпретации различных команд [2].

С помощью HTML мы в состоянии “обращаться и передавать информацию компьютеру.” Как он вообще работает? Существуют отдельные теги (<h1></h1>) - элементы, к которым мы обращаемся для того, чтобы передать некую информацию. К примеру, добавить заголовок, кнопку, текст… Существуют парные теги, но они ничем не отличаются от основной идеи их использования. У тегов уже есть некий вид, который мы можем менять и дополнять, но об этом позже. Приведу в пример фотографии из книги:



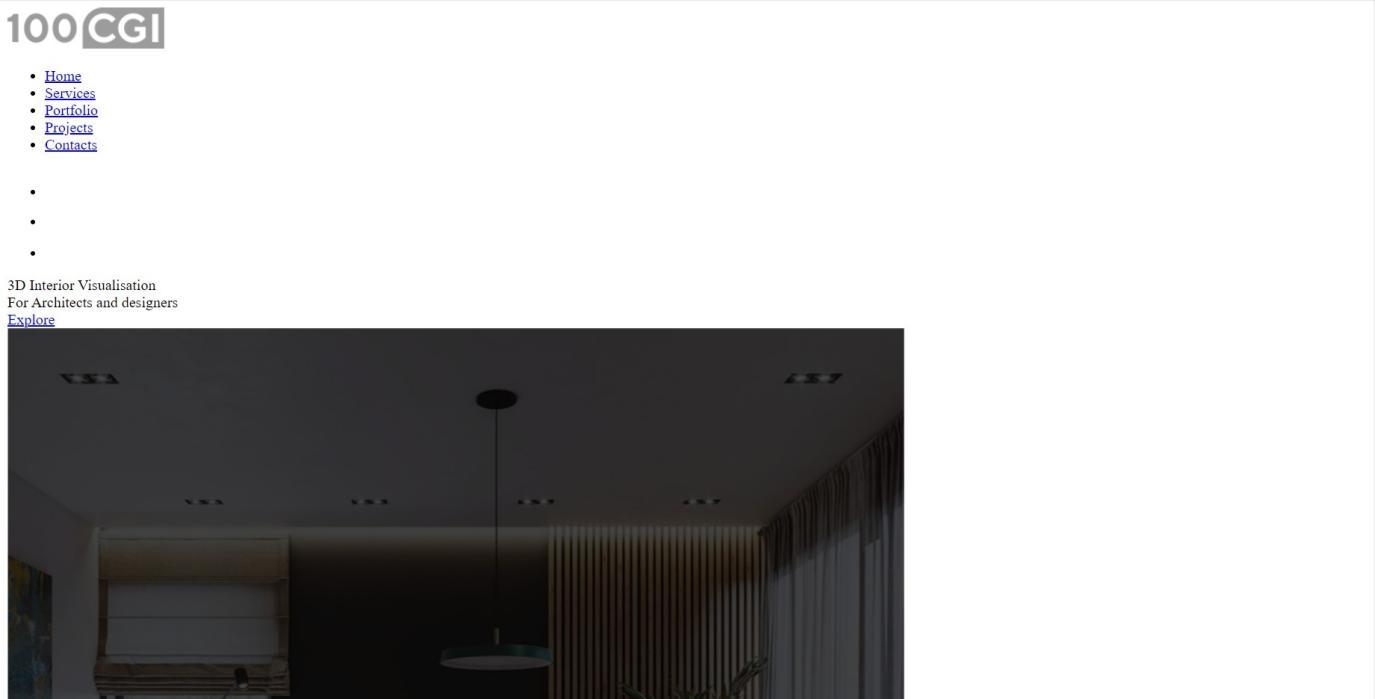
(рис. 2 страница 42/720 <https://vk.com/doc44301783_409697936?hash=XldcObu8K3KlTW7IxYvrcjTlhCnfcUBJHP9MHJHE5DT&dl=qZnhzNYtFz0gtk8dEEp5gLPByE0cnZZgD67zxkKzUTz> )

Актуальная по сей день версия HTML5. При помощи наглядного примера и его иерархии можно понять что тег <html></html> самый главный. Так и есть, благодаря ему мы в принципе можем увидеть наш код в браузере, мы же не можем зайти в дверь не открыв ее? <body></body> - тело сайта. В нем мы можем прописывать собственно команды, о которых я говорил выше. Есть ещё большое количество тегов, которые я не смогу вставить в доклад, но по этой ссылке с ними можно ознакомиться <http://htmlbook.ru/> Приведу пример из своего проекта:

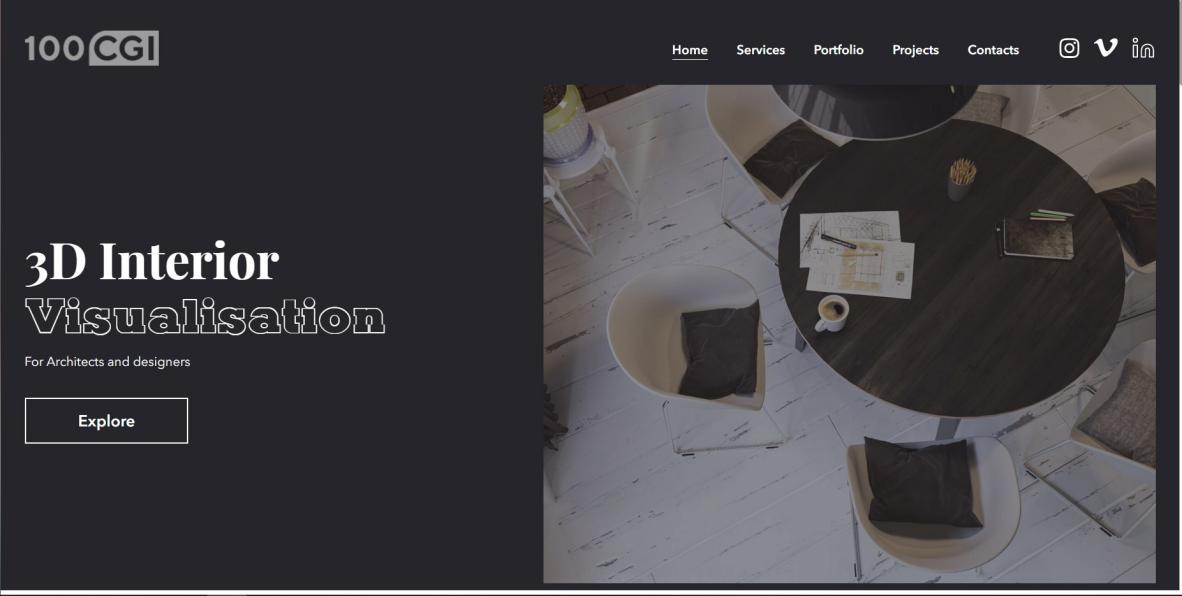


**CSS**

**CSS** (Cascade Style Sheets) — каскадные таблицы стилей. Если мы посмотрим на чистый HTML код в браузере, то увидим это.



Для того чтобы работать с ними нужно подключить файл с расширением .css с помощью специальной команды и её атрибута. <link rel="stylesheet" href="css/style.min.css"> Возьму собственный пример. Благодаря коду css мы получим это:



Как я уже упоминал ранее, у HTML тегов есть изначальное визуальное представление. Для того чтобы нам менять и создавать собственный дизайн продукта, мы должны пользоваться CSS. Как он работает? Тегам HTML присваивается класс (это есть в примере об HTML), мы записываем его и вешаем ему стили, свойства. Выглядит это так:



Существует также огромное количество свойств, которыми пользуются разработчики. Их никто не заучивает, просто поначалу стоит подглядывать в записи. Со временем мы сможем разговаривать на языке браузера. Ознакомиться со стилями можно также здесь <http://htmlbook.ru/>

На данный момент актуальная версия — это CSS(3), которая сильно продвинулась по своим возможностям, если мы будем сравнивать с прошлыми версиями.

## **JAVASCRIPT**

**JavaScript** - язык программирования. Считается одним из самых популярных и самых применяемых в веб разработке. Здесь он применяется как и в клиентской, так и серверной части. Слайдеры, якорные ссылки, анимация на странице... создаётся с помощью js. Для того чтобы с ним работать необходимо подключить страницу с расширением .js к HTML странице: <script src="js/main.min.js"></script>. Пример кода:



(рис. 3 <https://u.9111s.ru/uploads/202205/16/f5652ce60d06a66db4cf1c6582f7bef6.png> )

Для простого объяснения в различении языков применяют небольшое сравнение с человеком. Если HTML - скелет, CSS - кожа, одежда, внешний вид, JS - движение, динамичность [3].

# **BACKEND**

**Backend development** (серверная разработка) - второй этап, о котором я расскажу. Самый сок. К сожалению или к счастью, мне не доводилось работать с этой средой. Её я лишь изучал поверхностно. В задачи Backend разработчика входит следующее:

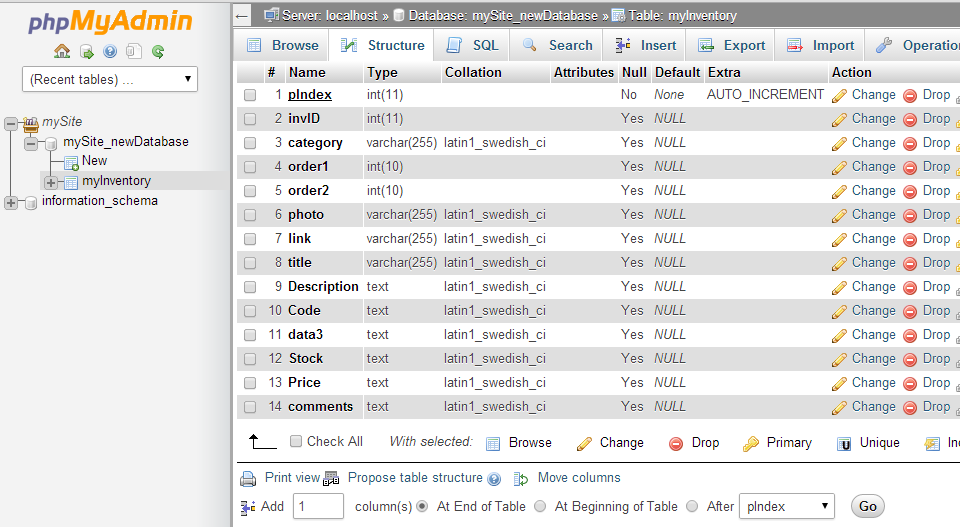
1. Безопасность веб сервисов
2. Настройка автоматических уведомлений
3. Контроль за корректной работой сайта, отладка его компонентов
4. Тестирование Frontend и его отдельных элементов на качество работы
5. Контролирование состояния серверов
6. Корректная работа форм и тому подобное
7. Управление и структуризация баз данных

По сути, Backend - все то, что мы не наблюдаем, так как пользователи видят и взаимодействуют только с Frontend частью. Второе ответвление направляет информацию, а мы наблюдаем её в виде визуального приятного концепта(первого). На самом деле в задачи Backend разработчика входят очень много обязанностей, но я не буду затрагивать все. Хочу рассказать то, что я пытался изучить из этой части веб разработки [5].

## 

## **БАЗА ДАННЫХ**

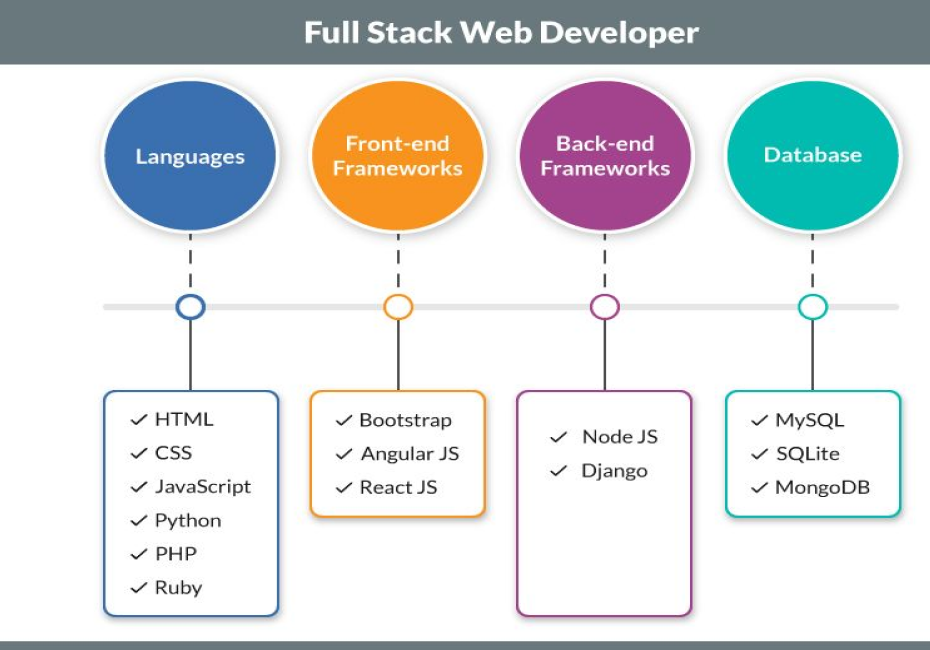
**База данных** это место где хранится информация, много информации неважно какого характера и размера, будь то информация о пользователе, его возраст, географические названия, фильмы и тому подобное. Предназначена для больших проектов, такие как интернет магазины или сайты о фильмах и их поиске... Выглядит она как таблица в которых написаны какие - либо данные. Когда я пытался знакомиться с Backend разработкой я работал с базой данных в программе PhpMyAdmin, своего примера у меня к сожалению нет, поэтому прикреплю из интернета:



(рис. 4 <https://www.briolin.com.ua/wp-content/uploads/2019/06/dbTable3.png> )

# **FULLSTACK**

**Fullstack** (от англ. Полный пакет) из контекста можно понять о чем идёт речь. Fullstack разработчики отвечают за все сферы веб разработки. Веб дизайн, анализ проекта, его внешний вид, поддержка баз данных, постоянная проверка Frontend структуры на качество работы и тому подобное. Конечно требования к таким разработчикам гораздо выше, потому что с подобной работой справится не каждый. Отвечать за две, казалось бы, противоположные сферы, при том чтобы все качественно работало, довольно тяжело, знаю по себе.. На картинке примерно изображено то, что необходимо знать Fullstack разработчику [6].



(рис. 5 <https://guruzarabotka.ru/wp-content/uploads/2022/04/full-stack-development-roadmap.jpg> )

# **ВЫВОД**

Веб разработка - интересное увлечение и увлекательная работа, на мой взгляд. Но несмотря на её привлекательность и прибыльность, это не значит что можно относится к ней только как к хобби. Поскольку сейчас интернет и интерфейсы заполнили нашу жизнь нам необходимо с ними взаимодействовать. Мы слишком дорожим своим временем, иначе говоря никому не хочется помещать места или сайты с непонятной, неструктурированной визуальной частью, ровно также как и с логикой. На разработчиках обоих ответвлений лежит огромная ответственность и регулярное обновление собственных знаний. Думаю я хотел бы сказать как и дизайнерам, так и остальным людям, создающие подобные проекты не переоценивать свою значимость и что нет необходимости недооценивать сферы, которые вы не выбрали, только потому что вы человек другого склада ума. Поистине классный сайт всегда будет радовать и все больше привлекать новых пользователей, а для этого придётся кооперировать всем вместе.

# **Список использованной литературы**

1. Эрик Фримен, Элизабет Робсон. Изучаем HTML (XHTML) и CSS. Издание 2. – СПб: Питер, 2014. – 8 стр.
2. Блок Яндекс Практикума. Что такое HTML: основы, необходимые веб-разработчикам - URL: <https://practicum.yandex.ru/blog/zachem-nuzhen-html/> (дата обращения - 09.12.2022)
3. JavaScript.ru. Введение в JS - URL: <https://learn.javascript.ru/intro> (дата обращения - 09.12.2022)
4. Стив круг. Веб дизайн или не заставляйте меня думать! Символ плюс, 2008.
5. Блок Яндекс практикума. Фронтенд или бэкэнд: по какому пути в разработке пойти - URL: <https://practicum.yandex.ru/blog/chem-otlichaetsya-backend-i-frontend/> (дата обращения - 09.12.2022)
6. GeekBrains, IT образование. Fullstack разработчик: кто такой и как им стать - URL: <https://gb.ru/blog/full-stack-razrabotchik/> (дата обращения: 09.12.2022)