## Процесс тестирования frontend

### Процесс тестирования можно разбить на несколько этапов:

1. Подготовительный этап и изучение документации.
2. Проверка разработчиком проекта
3. Проверка дизайнером проекта и тимлидом
4. Проверка тестировщиком с дополнительной имитацией возможных ситуаций по разным user-кейсам
5. Проверка по чек-листу после backend интеграции верстки и наполнения всех разделов контентом

### Можно выделить следующие параметры для проверки:

#### Соответствие макету

* + Расхождение макета и верстки в пикселях
  + Корректный вывод элементов интерфейса
  + Наличие элементов, выбивающихся из цветовой гаммы сайта

#### Работа в разных окружениях

* + Кроссбраузерность
  + Корректная работа на разных устройствах
  + Корректная работа на разных операционных системах
  + Корректная работа с разной скоростью интернета
  + Корректное отображение кнопок, полей ввода, выпадающих списков и радиокнопок на разных устройствах

#### Проверка на разных разрешениях экрана (проверка десктопной и адаптивных версий)

* + Корректное отображение на всех возможных размерах окна браузера
  + Область нажатия кнопок
  + Корректное отображение плавающих (закрепляемых за прокруткой) элементов
  + Корректное использование брейкпоинтов для стилей
  + Наличие сломанной вёрстки при изменении размеров экрана

#### HTML

* + Валидность HTML
  + Наличие корректного DOCTYPE
  + Наличие корректной кодировки
  + Наличие тега title и мета-тегов для SEO
  + Наличие атрибута lang у тега html
  + Наличие пустых и ненужных тегов
  + Наличие объёмных комментариев в коде

#### Javascript

* + Не должно быть ошибок javascript при работе с сайтом
  + Файл скриптов должен подключаться внизу страницы
  + Использование кода из неподходящей версии javascript
  + Корректное подключение сторонних библиотек
  + Некорректная работа плагинов
  + Адекватное отображение сайта при выключенном javascript
  + Наличие кода на «боевом» сайте, предназначенного для разработки на тестовом сервере
  + Корректная инициализация контентных слайдеров

#### CSS

* + Валидация CSS
  + Файлы стилей должны быть корректно подключены
  + Стили должны быть объединены в один файл и минифицированы
  + Использование контентных тегов для стилизации (br, b, strong и т.д.)
  + Стилизация элементов по атрибутам name или id
  + Использование одинаковых цветов, скруглений, отступов, размеров шрифтов
  + Использование unicode-символов в файлах стилей и скриптов
  + Дергающиеся и некорректно работающие анимации
  + Адекватные отступы между блоками контента
  + Наличие горизонтальной прокрутки

#### Фавиконки

* + Наличие favicon.ico и фавиконки больших размеров
  + Наличие manifest.json или manifest.webmanifest

#### Шрифты

* + Наличие малоиспользуемых или подключение неиспользуемых на сайте шрифтов
  + Правильное подключение шрифтов
  + Подключение шрифтов только из локальных источников
  + Предзагрузка шрифтов

#### Навигация

* + Битые ссылки
  + Ссылки, которые ведут на текущую страницу (на главной странице верхний логотип сайта не должен быть ссылкой)
  + Корректная верстка меню
  + Корректная верстка «хлебных крошек»
  + Корректное отображение меню при различном количестве пунктов меню
  + Наличие стилей для индикации текущего элемента в навигации и неактивных элементов

#### Заголовки

* + Наличие одного тега h1 на странице
  + Иерархия заголовков (должны идти по порядку)
  + Заголовки должны использоваться только в контентной части, например, заголовки страницы
  + Правильно настроенные стили и соотношение размеров заголовков

#### Контент

* + Стили для контентных элементов должны быть прописаны: параграфы, изображения, обычные и нумерованные списки, таблицы
  + Блоки должны корректно отображаться при любом количестве контента внутри них
  + Необходимо тестировать сайт с реалистичными изображениями и текстами
  + Проверка орфографии, в том числе и в самом интерфейсе

#### Ссылки и кнопки

* + Выделение интерактивных элементов при наведении и фокусе
  + Все нажимаемые области должны иметь cursor: pointer
  + Наличие специальных общих классов для кнопок и полей ввода
  + Наличие пустых ссылок
  + Ссылки, ведущие на небезопасные ресурсы
  + Некорректное поведение кнопок на тач-устройствах
  + Незастиленные ссылки при наведении
  + Наличие неактивных кнопок, на которые можно нажать

#### Формы

* + Наличие атрибутов для ограничения длины ввода
  + Наличие масок для полей ввода
  + Клиентская валидация полей ввода
  + Корректные атрибуты type у полей ввода
  + Различные стили элементов форм в разных браузерах и устройствах
  + Наличие уведомления после отправления формы
  + Корректный сброс формы после отправки

### Инструменты для тестирования frontend

1. PerfectPixel - расширение для Chrome, позволяющий накладывать изображение макета поверх сайта, таким образом можно сравнить приближенность верстки к макету.
2. Google Speed - комплексный инструмент для определения производительности, удобства и выбора эффективных путей оптимизации сайта.
3. Selenium — инструмент для автоматизации действий браузера и тестирования frontend.
4. Screaming Frog — SEO-инструмент, который позволяет находить некорректные и битые ссылки на сайте.
5. Яндекс.Вебвизор — анализ поведения пользователей на сайте.
6. Яндекс.Вебмастер– позволяет проверить семантическую разметку, внедренную на вашем сайте, и убедиться в том, что роботы поисковых систем смогут извлечь структурированные данные.
7. labrika - хороший сервис для проверки сайта как на наличие технических ошибок, так и seo
8. Chrome DevTools - дает множество возможностей для анализа ошибок и возможности узнать происхождение багов
9. lightshot - приложение позволяющее делать скриншоты

## Процесс тестирования backend

На данный момент возникают сложности в описании процесса тестирования backend.

Могу сказать, что в тестировании должна присутствовать проверка бд, api.

**Тестирование подразделяется на**

* Структурное тестирование имеет дело с внутренней структурой базы данных (т.е. метаданными). Он включает в себя проверку таблиц и всех других объектов базы данных, которые недоступны пользователям напрямую, таких как столбцы, ключи, индексы, схемы, триггеры, функции и т. д.
* Функциональное тестирование проверяет, как данные отображаются между внутренним и внешним интерфейсом. Например. Что касается веб-сайта электронной коммерции, если мы запрашиваем цену любого продукта, внешний интерфейс должен отображать правильную таблицу и возвращать действительные данные.
* Нефункциональное тестирование можно рассматривать как тестирование производительности, нагрузки и стресс-тестирования, когда тестировщики проверяют производительность базы данных на случай непредвиденных нагрузок. Тестирование безопасности также является частью нефункционального тестирования.

### Инструменты для тестирования backend

1. Chrome DevTools - дает множество возможностей для анализа ошибок и возможности узнать происхождение багов
2. lightshot - приложение позволяющее делать скриншоты
3. Postman - HTTP-клиент для тестирования API.