Чек-лист проверки скорости сайта

Что проверить, если сайт загружается медленно

Скорость сайта зависит не от одного показателя. Проблема может быть связана с изображениями, кодом, сервером, базой данных, JavaScript, сторонними сервисами или самой архитектурой проекта.

Этот чек-лист поможет определить основные возможные причины замедления и понять, что стоит проверить в первую очередь.

1. Изображения

☐ Изображения оптимизированы перед загрузкой на сайт

☐ Используются современные форматы изображений, когда это возможно

☐ Размер изображения соответствует реальному размеру блока на странице

☐ Большие изображения не загружаются там, где достаточно небольшой версии

☐ Для изображений ниже первого экрана используется отложенная загрузка

☐ Нет изображений, которые загружаются на странице, но фактически не используются

> Что проверить: Какие изображения занимают больше всего места и действительно ли они нужны в таком размере.

2. Первый экран

☐ Основное изображение первого экрана не слишком тяжёлое

☐ Первый экран не зависит от большого количества сторонних ресурсов

☐ Критически важный контент загружается в первую очередь

☐ Нет ситуации, когда пользователь долго видит пустой экран

> Важно: Оптимизация первого экрана особенно заметна пользователю. Даже если вся страница загружается быстро, медленный старт создаёт ощущение «тормозящего» сайта.

3. CSS и JavaScript

☐ На странице загружается только необходимый JavaScript

☐ Нет большого количества неиспользуемого кода

☐ Сторонние библиотеки действительно нужны

☐ Скрипты не блокируют отображение основного контента

☐ CSS не перегружен ненужными стилями

☐ Нет нескольких библиотек, выполняющих одну и ту же задачу

> Что проверить: Не добавлялись ли новые функции постепенно без последующей проверки их влияния на производительность.

4. Сервер

☐ Сервер отвечает достаточно быстро

☐ Сервер имеет достаточные ресурсы для проекта

☐ Время ответа не увеличивается при обычной нагрузке

☐ Настройки веб-сервера оптимизированы

☐ Используется сжатие передаваемых данных

☐ Настроено кэширование там, где оно действительно необходимо

> Важно: Иногда сайт кажется медленным не из-за фронтенда, а потому что сервер слишком долго формирует ответ.

5. База данных

☐ Страницы не выполняют избыточное количество запросов к базе данных

☐ Нет повторяющихся запросов к одним и тем же данным

☐ Сложные запросы действительно необходимы

☐ Большие объёмы данных не загружаются целиком без необходимости

☐ Используются подходящие индексы

> Простой вопрос: Если одна страница делает десятки запросов к базе данных — действительно ли все они необходимы?

6. Кэширование

☐ Кэширование используется там, где данные не меняются постоянно

☐ Статические файлы корректно кэшируются браузером

☐ Повторное открытие страниц не требует каждый раз выполнять одну и ту же работу

☐ После обновления контента старый кэш корректно обновляется

> Важно: Кэширование — не способ скрыть плохую архитектуру. Оно должно уменьшать повторную работу системы, а не маскировать её проблемы.

7. Сторонние сервисы

☐ Проверены внешние шрифты

☐ Проверены аналитические системы

☐ Проверены карты и виджеты

☐ Проверены рекламные и маркетинговые скрипты

☐ Нет ненужных внешних библиотек

☐ Сторонние сервисы не блокируют загрузку основного контента

> Что проверить: Иногда сайт становится медленнее после добавления всего одного внешнего сервиса.

8. Шрифты

☐ Используется разумное количество начертаний

☐ Не загружаются ненужные варианты шрифта

☐ Размер файлов шрифтов оптимизирован

☐ Шрифты не задерживают отображение основного текста

☐ Нет большого количества разных семейств шрифтов

9. Мобильная версия

☐ Скорость проверялась на мобильных устройствах

☐ Мобильная версия не загружает ненужные ресурсы

☐ Изображения адаптированы под мобильный экран

☐ Размер JavaScript не создаёт проблем на мобильных устройствах

☐ Основной контент появляется достаточно быстро

> Важно: Сайт, который быстро открывается на мощном компьютере, не обязательно будет быстрым на смартфоне.

10. Архитектура сайта

☐ Новые функции не добавлялись без оценки их влияния на производительность

☐ Страницы не выполняют лишнюю работу

☐ Повторяющиеся операции вынесены в подходящие механизмы

☐ Тяжёлые операции не выполняются без необходимости при каждом открытии страницы

☐ Архитектура позволяет оптимизировать отдельные части сайта без полной переделки

> Главный вопрос: Не является ли сама архитектура причиной того, что сайт приходится постоянно «ускорять костылями»?

11. Как правильно искать проблему

Не стоит начинать оптимизацию с предположения: «Наверное, проблема в картинках». Правильнее пройти несколько этапов.

☐ Зафиксировать текущую скорость

☐ Определить, какая часть загрузки занимает больше всего времени

☐ Проверить первый экран

☐ Проверить сетевые запросы

☐ Проверить изображения

☐ Проверить JavaScript и CSS

☐ Проверить серверное время ответа

☐ Проверить запросы к базе данных

☐ Проверить сторонние сервисы

☐ После изменений повторить измерение

> Главный принцип: Сначала найти причину — потом оптимизировать.

12. Что не стоит делать

☐ Не удалять функции только ради красивого показателя скорости

☐ Не отключать важные скрипты без понимания их назначения

☐ Не устанавливать несколько систем кэширования одновременно без необходимости

☐ Не уменьшать качество всех изображений одинаково

☐ Не ориентироваться только на один тест

☐ Не пытаться исправить архитектурную проблему дополнительным кэшем

☐ Не считать сайт быстрым только потому, что он быстро открывается у разработчика

13. Итоговая проверка

Базовый уровень

☐ Изображения оптимизированы

☐ Первый экран загружается быстро

☐ Нет лишнего JavaScript

☐ Нет лишних внешних сервисов

☐ Настроено кэширование

☐ Проверена мобильная версия

Технический уровень

☐ Проверено время ответа сервера

☐ Проверены запросы к базе данных

☐ Проверены сетевые запросы

☐ Проверены CSS и JavaScript

☐ Проверена архитектура тяжёлых операций

☐ Повторное измерение выполнено после оптимизации

Если проблема всё ещё остаётся

Если после базовой оптимизации сайт остаётся медленным, стоит проверить уже не отдельные элементы, а архитектуру проекта целиком.

Возможная причина может находиться в серверной части, базе данных, структуре запросов, рендеринге страниц, сторонних сервисах, JavaScript, системе кэширования или самой архитектуре приложения.

> В таком случае бессмысленно бесконечно уменьшать изображения или удалять несколько килобайт JavaScript — проблема может находиться совсем в другом месте.

Главное

Быстрый сайт — это не сайт без функций.

Хорошая производительность достигается не отказом от возможностей, а правильной архитектурой, оптимизацией ресурсов и пониманием того, какую работу действительно должна выполнять система.

Перед запуском или оптимизацией сайта спросите себя:

☐ Знаю ли я, что именно замедляет сайт?

☐ Измерял ли я скорость до внесения изменений?

☐ Проверял ли я мобильную версию?

☐ Проверял ли я сервер и базу данных?

☐ Проверял ли я сторонние сервисы?

☐ Проверил ли я результат после оптимизации?

> Если на эти вопросы есть ответы — оптимизация становится управляемым процессом, а не попыткой «ускорить сайт на глаз».