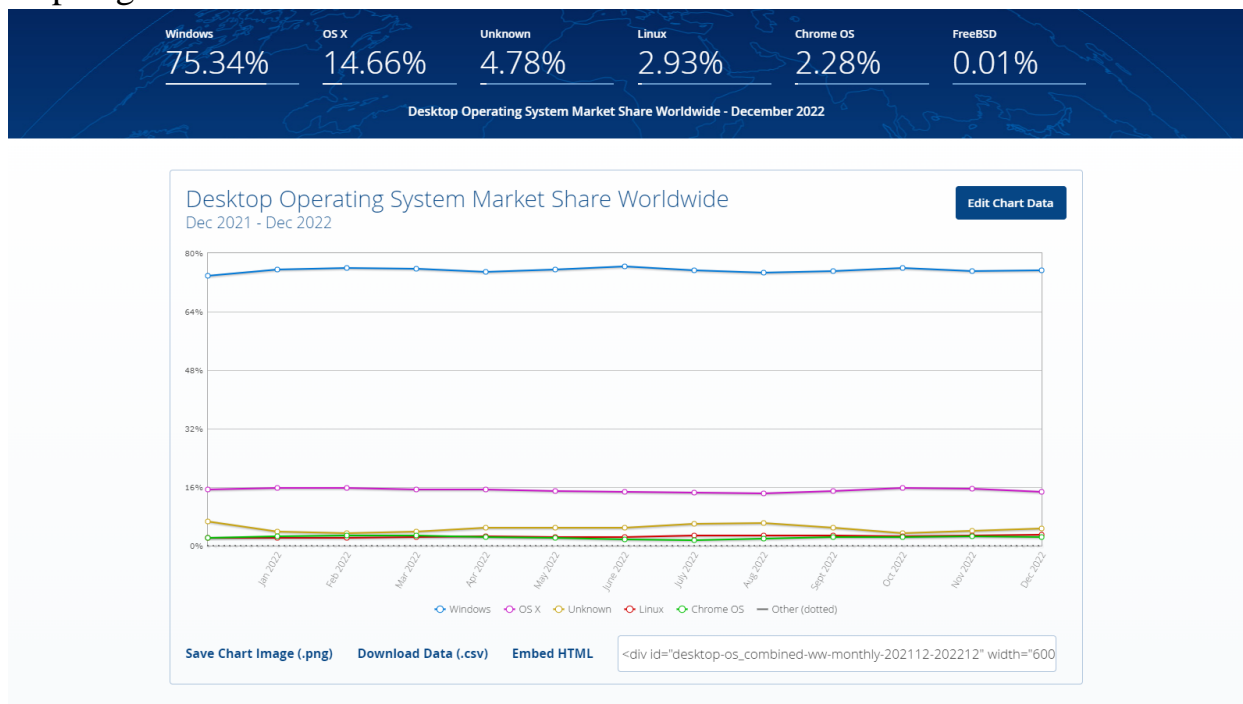


### 3 инварианта названия:

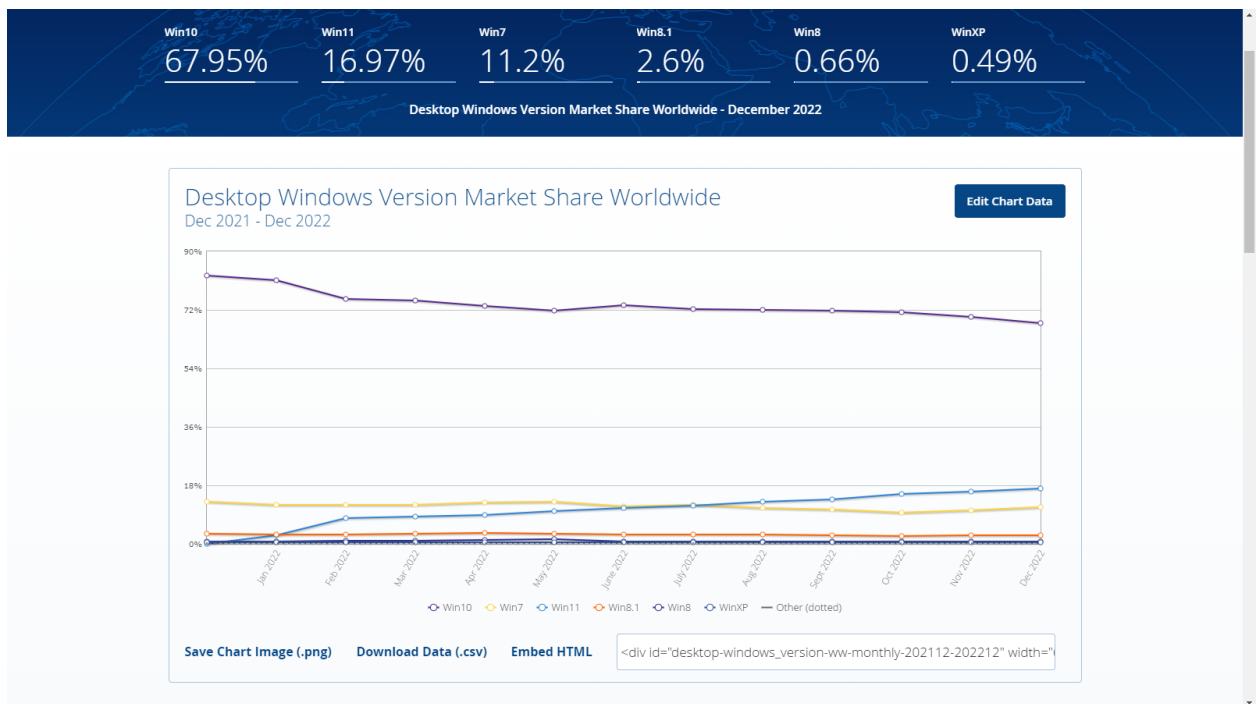
1. Web-приложение «Электронный журнал» с улучшенной кроссбраузерностью.
2. Веб-сервис для ведению списков успеваемости и посещаемости
3. Web-приложение «Электронный журнал»

Немного в свободной форме о работе. Это будет полноценный электронный журнал, с клиент-серверной архитектурой и улучшенной кроссбраузерной совместимостью, вопросы поддержки браузеров нужно будет исследовать после конечного результата, я ранее предварительно рассматривал возможности, но итоги где-то затерялись. По памяти я могу вспомнить, что была поддержка уровня Opera 15+, с современными браузерами, естественно проблем не возникнет. В проекте упор на поддержку Internet Explorer, на данный момент я часто работаю над увеличением возможностей в нём, поэтому могу сразу заявить о уровне IE6+.

Почему IE? Потому что, это браузер по умолчанию самой популярной системы в мире, даже на данный момент. Статистику возьму с сайта: <https://gs.statcounter.com>



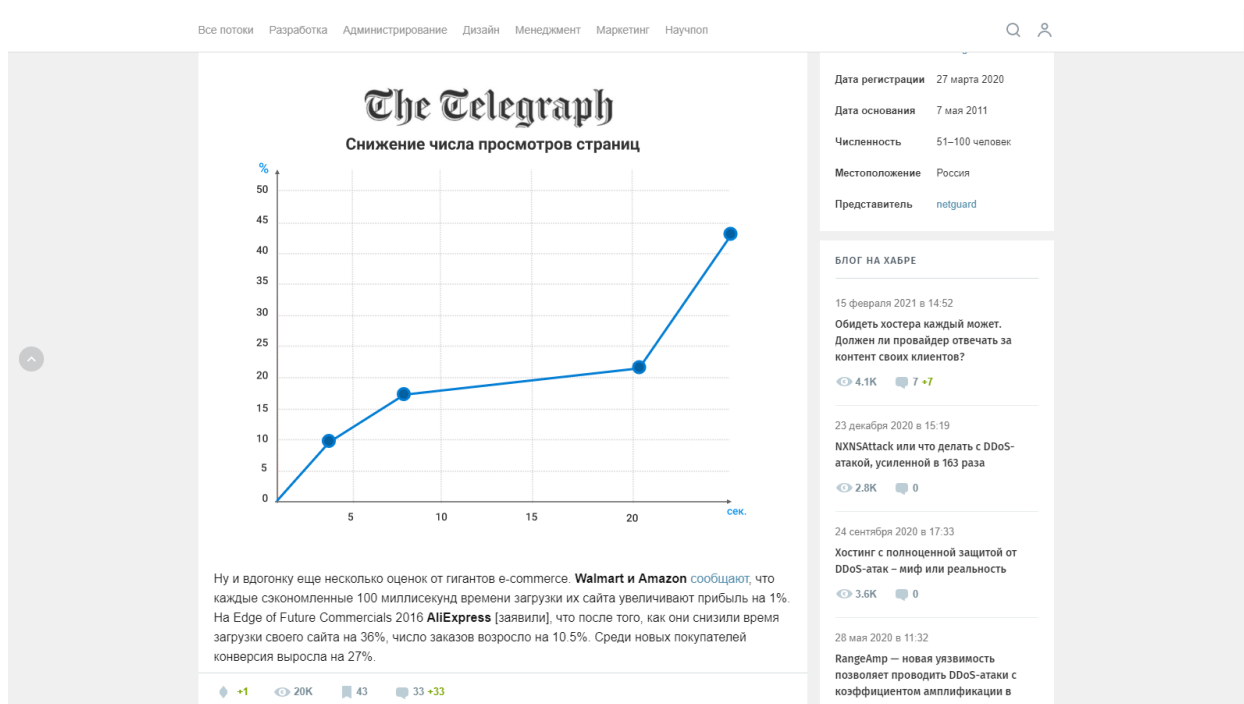
IE был браузером по умолчанию вплоть до Windows 8.1, к слову, старые системы до сих пор «в моде».



Поэтому, в целом, всё ещё имеется смысл поддержки этого старого браузера. Как цель, выбрана поддержка ОС до «разумных пределов», то есть до Windows XP. Максимальная версия Opera для Windows XP, Opera 36, поэтому уровень Opera 15+ вполне является адекватным, так как вряд-ли кто-то до сих пор остался на старых версиях браузеров, не предустановленных разработчиком ОС. Windows XP по умолчанию была с Internet Explorer 6. Остальные браузеры необходимо тестировать.

Ранее я упомянул про упор в IE, немного расскажу подробнее. Цель не просто запустить сайт без ошибок на условном IE6, но и ещё сделать полноценный адаптивный интернет ресурс, который не сильно будет уступать по функционалу и визуальной составляющей, версии сайта на более современных браузерах. К сожалению, старые браузеры не поддерживают CSS3, поэтому мне придётся воссоздать некоторые функции CSS3. Естественно кое-какой задел у меня уже имеется. Часть функций, вероятнее всего, самую сложную, мне уже удалось реализовать. И на данном этапе стало понятно, что скорость начальной загрузки ощутимо пострадала во всех браузерах, поэтому я принял решение создать свой скромный аналог, Webpack.

Почему важна скорость загрузки сайта? По данному поводу было проведено множество исследований, в качестве примера я приведу статью: <https://habr.com/ru/company/ddosguard/blog/499756/>



Каждая лишняя секунда увеличивает шанс потери клиента. Вывод из статьи: хорошее время загрузки составляет не более трех секунд. Конечно, в моём случае она не достигла критических значений, однако была на грани.

Немного о программе. С фреймворками работать не способна, но классические сайты переварить способна. Процесс в фоновом режиме следит за папкой проекта и, в случае изменения ресурсов проекта, создаёт две версии проекта: debug и reliz. Названия говорящие:) Программа имеет консольный интерфейс и способна выполнять классические манипуляции с проектом для его оптимизации. Сжатие изображений, минификация CSS, HTML, JS. В итоге, получилось потерянные секунды скорости загрузки, вернуть.

Теперь о самом веб-сервисе. Типичный электронный журнал. Задумывается с 5-тью ролями: обучающийся, родитель, педагог, завуч, администратор портала. На данный момент реализую прототип клиента на React. 3 из 5 ролей готовы. Пока серверную часть не успел сделать. В данной окончательное "видение" еще не пришло, поэтому вполне могут быть самые наизысканные изменения и смысла заранее в подробности рассказывать, не имеется.

**Какие технологии/программный инструментарий будет использоваться:**

Здесь тоже могут быть множественные дополнения, но это то, что видно на данный момент. Также вполне мог забыть упомянуть какие-то технологии.

Клиент:

- Notepad++
- JS(ES5)
- HTML(HTML 4.01 Transitional)
- CSS
- JQuery(1.4.3)
- LESS(1.3.3)
- CSS3 PIE(1.0)
- jquery.placeholder

Оптимизатор:

- Java
- IntelliJ IDEA
- JDK
- Gradle
- Lombok
- Google- closure-compiler
- Jsoup
- slf4j
- launch4jc

Предположения по серверу:

- Java
- IntelliJ IDEA
- JDK
- Gradle
- Lombok
- slf4j
- launch4jc
- PostgreSQL

**Как всё это будет работать (схематично):**

Представлю максимально общую схему. Перед релизом сайт будет пропущен через «оптимизатор», готовая версия будет опубликована в

интернете. Нельзя забывать про сервер, его необходимо запустить, после чего интернет ресурс станет доступен клиентам.

### **В чем видятся преимущества работы:**

Возможно, не самый актуальный, но масштабнейший и достаточно интересный проект. Имеет несколько различных частей: клиент, сервер, оптимизатор.

### **Какой есть задел:**

В изложении постарался в максимально краткой форме рассказать.

## **Post Scriptum:**

Простите, что затянул с выбором темы... Проект чрезвычайно масштабный, да и крайне сложный, некоторые идеи и вовсе казались невозможными, поэтому необходимо было их заранее реализовать, чтоб не попасть в ловушку при защите:) Но, конечно же, проект ещё далёк до релиза.