

*Головко Ярослав Юрьевич,
студент института инженерных и цифровых технологий
4 курс, направление подготовки «Бизнес-информатика»
Белгородский Государственный Научно-Исследовательский Университет
Россия, г. Белгород*

АНАЛИЗ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ РАЗРАБОТКИ ВЕБ-САЙТОВ

***Аннотация:** В данной статье были рассмотрены инструментальные средства разработки лицевой, а также серверной части веб-сайтов. Среди инструментов разработки лицевой части веб-сайтов описаны: HTML, CSS, JavaScript. Для реализации разработки серверной части рассмотрены следующие альтернативы: разработка «с нуля», SaaS-решение, CMS, framework. Рассмотрены основные особенности данных альтернатив, выделены их преимущества и недостатки.*

***Ключевые слова:** HTML, CSS, JavaScript, SaaS, CMS, framework.*

***Annotation:** This article examined the development tools for the front and server side of websites. Among the front-end website development tools are described: HTML, CSS, JavaScript. To implement the development of the back-end side, the following alternatives are considered: development from scratch, SaaS-solution, CMS, framework. The main features of these alternatives are considered, their advantages and disadvantages are highlighted.*

***Key words:** HTML, CSS, JavaScript, SaaS, CMS, framework.*

Традиционно, создание веб-сайта ведется по системе front-end и back-end, то есть иерархическое разделение процесса разработки сайта на две части: разработка пользовательского интерфейса (front-end) и программно-административной части (back-end).

Front-end разработка — это разработка публичной части веб-сайта, с который будет контактировать пользователь. Front-end составляющая часть сайта отвечает за вывод определенной информации пользователю ресурса и по факту совершения, им, каких либо действий на сайте, интерпретацию ее в вид понятный программам, относящимся к back-end.

HTML (HyperText Markup Language) – язык разметки (маркировки) гипертекста. Гипертекст своим развитием обязан интернету, хоть и создавался он совсем не для того. HTML дает возможность производить переход от одной части текста к другой, и, что замечательно, эти части могут храниться на совершенно разных компьютерах. HTML не стоит путать с языками программирования, он создан специально для разметки веб-страниц. Именно язык разметки дает браузеру необходимые инструкции о том, как отображать тексты и другие элементы страницы на мониторе. Важно заметить, что не только различные браузеры, но и различные их версии могут по-разному воспринимать и отображать на экране код. Код разметки в HTML состоит из так называемых «тегов». Теги предоставляют информацию браузерам о форматировании и разметке страницы. Название тега заключается в угловые скобки «<» и «>» [1].

CSS – это формальный язык, служащий для описания оформления внешнего вида документа, созданного с использованием языка разметки (HTML, XHTML, XML). Название происходит от английского Cascading Style Sheets, что означает «каскадные таблицы стилей». Назначение CSS – отделять то, что задает внешний вид страницы, от ее содержания. Если документ создан только с использованием HTML, то в нем определяется не только каждый элемент, но и способ его отображения (цвет, шрифт, положение блока и т. д.). Если же подключены каскадные таблицы стилей, то HTML описывает только очередность объектов. А за все их свойства отвечает CSS [2].

Заключительным этапом разработки лицевой части веб-сайта является написание JS скриптов. JavaScript – язык программирования, являющийся прототипно-ориентированным. Он отражает язык ECMAScript, чьим прототипом изначально и являлся. Первая вариация появилась ещё в 1995 году и

с тех пор постоянно совершенствовалась, пока не пришла к нынешнему виду. Чаще этот язык используется в разработке приложений и браузерах с целью придания им интерактивности и «живости». Базовой особенностью этого языка отмечается то, что на него повлияли другие (Python, Java и др.) языки программирования с целью придания максимального комфорта JavaScript и лёгкости в понимании его теми пользователями, которые не имеют соответствующего образования и глубинных знаний – не программистами [3].

Back-end разработка - это веб-программирование, целью которого является создание серверной стороны веб-сайта, интеграция базы данных и связь ее с пользовательской (front-end) стороной.

Back-end – разработка подразумевает:

- проектирование архитектуры кода и дальнейшая работа с ним;
- создание ядра сайта;
- разработка основы функционала;
- контроль серверов, баз данных, приложений, непрерывной интеграции данных.

Существует четыре варианта исполнения серверной части веб-сайта:

- разработка «с нуля»;
- SaaS-решение;
- CMS;
- разработка на фреймворке;

Рассмотрим подробнее каждое из них. Разработка серверной части «с нуля» без использования каких-либо платформ может быть правильным решением только в двух случаях: либо проект очень нагруженный и крайне требовательный к низкоуровневой оптимизации, либо проект слишком простой и не нуждается в последующем развитии. В остальных случаях стоит выбрать другой вариант, поскольку качественная разработка «с нуля» стоит больших денег.

SaaS (software as a service — программное обеспечение как услуга) - это способ распространения программного обеспечения на арендной основе. В случае разработки веб-сайтов, поставщик SaaS-решения предлагает сконфигурировать выданный веб-сайт и ежемесячно вносить абонентскую плату. Такой вариант имеет следующие преимущества: скорость создания готового веб-сайта и низкая цена. Однако SaaS-решение также имеет много существенных недостатков: шаблонный дизайн, жесткие рамки функциональных возможностей, и самое главное – низкая производительность и отказоустойчивость. Можно сделать вывод, что такой вариант совершенно не подходит для активно посещаемых проектов.

CMS — это система управления контентом, набор скриптов для создания, редактирования и управления контентом сайта. Если раньше большинство сайтов были статичными и требовали внесения правок в их содержимое вручную, то сейчас динамика развития проектов требует готовности быстро реагировать на изменения и внедрять их с максимальной оперативностью [4].

Программное обеспечение CMS позволяет создавать типовые веб-сайты из блоков, подобно конструктору. Такая платформа обладает следующими преимуществами:

- быстро и просто можно создавать те виды веб-сайтов, которые заложены в функционал CMS;
- реализация многих бизнес-процессов, необходимых веб-сайту, уже существует в системе в готовом виде;
- быстрое масштабирование в рамках заложенного функционала.

Однако использование при разработке CMS накладывает на проект следующие ограничения:

- тонкая настройка под индивидуальные бизнес-процессы затруднена, поскольку, процессы, заложенные в систему, обычно тесно связаны, поэтому модификация одного из них приводит к необходимости модифицировать и зависимые от него;

- реализация нешаблонного функционала зачастую либо затруднена, либо вовсе невозможна;
- избыточность функционала, что приводит к сложностям в использовании и управлении веб-сайтом;
- излишне «раздутая» структура хранения данных относительно реальных потребностей проекта, что сказывается на скорости работы веб-сайта.

Исходя из вышесказанного, можно сделать вывод о том, что CMS наиболее всего подойдет для типового веб-сайта, когда заложенный в систему функционал полностью отвечает требованиям разрабатываемого проекта.

Фреймворк (framework – каркас, структура) – это программная платформа, определяющая структуру разрабатываемой системы. Как правило, фреймворк содержит только базовые модули, а все индивидуальные для проекта компоненты разрабатываются на их основе. Веб-сайт, разработанный при помощи фреймворка, обладает следующими преимуществами:

- благодаря унифицированной структуре, веб-сайт будет прост в обслуживании и доработке;
- экосистемы фреймворков зачастую содержат большое количество готовых реализаций функциональных возможностей;
- возможно реализация индивидуальных бизнес-процессов, а не только тех, которые изначально заложены в систему;
- веб-сайты, разработанные на фреймворке обладают высокой производительностью, а также выдерживают большие нагрузки, чем другие платформы.

Недостатками разработки на фреймворке являются:

- разработка типового функционала на фреймворке длится дольше, чем при использовании CMS;
- разработка на фреймворке требует понимания реализуемых бизнес-процессов.

Из вышесказанного следует, что обладая большой гибкостью и высокой производительностью, фреймворки являются отличным решением для разработки нагруженных веб-сайтов.

Таким образом были исследованы инструментальные средства разработки веб-сайтов. В лицевой разработке front-end были рассмотрены такие инструменты как: язык разметки HTML, таблицы стилей CSS и язык интерактивных сценариев JavaScript. Также были проанализированы возможности реализации серверной части, выделены преимущества и недостатки альтернатив.

Список использованных источников:

1. Что такое HTML. Пояснение для новичков [Электронный ресурс] / – Электрон. текстовые дан. – URL: <https://blogwork.ru/chto-takoe-html/>
2. Что такое CSS, для чего нужны CSS [Электронный ресурс] / – Электрон. текстовые дан. – URL: <https://wiki.rookee.ru/css/>
3. JavaScript – что это такое, преимущества и недостатки [Электронный ресурс] / – Электрон. текстовые дан. – URL: <https://ipipe.ru/info/javascript>
4. Что такое CMS сайта [Электронный ресурс] / – Электрон. текстовые дан. – URL: <https://hostiq.ua/wiki/cms/>