

УДК 622

АНАЛИЗ СУЩЕСТВУЮЩИХ ПОДХОДОВ К РАЗРАБОТКЕ ВЕБ-САЙТОВ

Нагорных Р.И., магистрант гр. ПИМ-191, II курс
Кузбасский государственный технический университет
имени Т.Ф. Горбачева
г. Кемерово

Одним из главных вопросов на начальной стадии разработки проекта является выбор тех или иных средств разработки. Данная проблема будет актуальна всегда, так как невозможно разработать технологию, оптимально подходящую для решения любой задачи. На начальной стадии разработки веб-сайта у нас – разработчиков возникла данная проблема, ведь существуют разные способы создания сайтов, основными из которых на данный момент являются:

- Использование конструкторов сайтов;
- Применение систем управления контентом;
- Разработка сайта с нуля с использованием веб-фреймворков.

Конструктор сайта – программное решение, позволяющее построить сайт по модульному принципу, когда разработчик собирает сайт с помощью готовых частей, находящихся в конструкторе. Данный подход позволяет создать сайт без знаний языков программирования. Больше всего подойдет тем, кто хочет сделать простенький сайт о себе или своей компании, без претензий на качество программного кода и скорость работы сайта. Также, на конструкторах часто делают простенькие лендинги для небольших рекламных кампаний. Для разработки серьезного корпоративного решения или интернет-магазина конструкторы не подходят.

Преимущества:

- Низкая стоимость – большая часть компаний предоставляют бесплатное использование конструктора на начальном этапе, а стоимость подключаемых модулей очень низкая;
- Простота разработки – для того, чтобы создать свой сайт, достаточно уметь пользоваться браузером компьютера и иметь чувство вкуса.

Недостатки:

- Скрытые затраты – за недорогое использование конструктора на начальных этапах создания сайта приходится платить скрытыми тратами, сумма которых довольно значительна – размещение на хостинге, домен второго уровня и т. д.;
- Домен третьего и более высокого уровня – при использовании бесплатных аккаунтов в конструкторах, клиент получает размещение своего сайта только на домене не ниже третьего

уровня, например, `sitename.constructor.com`. Такие домены не вызывают большого доверия у интернет-сообщества, а домены второго уровня (напр. `sitename.com`) стоят дороже, чем если их приобретать у регистраторов доменов напрямую;

- Тяжеловесность сайта – сайт, сделанный на конструкторе всегда будет работать медленнее, чем сайт на CMS или разработанный самостоятельно. Данная проблема возникает из-за того, что конструктор содержит большое количество программного кода, не относящегося к созданному сайту, но необходимого для построения его итогового внешнего вида;
- Отсутствие SEO. Несмотря на заявления любого конструктора сайтов провести полноценное SEO сайта, сделанного на конструкторе, не представится возможным по причине того, что SEO – это комплекс мероприятий который включает в себя: работу с кодом, индексацию в поисковых системах, построение структуры сайта. Для этого всего требуется доступ к программному коду и разметке, доступ к которым на конструкторе получить невозможно.

CMS – это комплекс программных инструментов, позволяющие создавать и поддерживать веб сайт или приложение, обновлять контент и взаимодействовать с пользователями. С помощью CMS можно реализовать даже сложные сайты, например, интернет магазин или большой корпоративный сайт. Но и у данной технологии есть свои особенности.

Преимущества CMS:

- Бесплатный доступ – практически все CMS (кроме 1С-Битрикс) изначально предоставляются разработчикам бесплатно;
- Удобное управление контентом;
- Огромное количество готовых решений, которые можно адаптировать под свой проект.

Недостатки CMS:

- Уязвимость сайта – порой сайты на конструкторах имеют меньше уязвимостей, чем сайты, находящиеся под управлением WordPress или других CMS;
- Требование к знанию языков программирования – для успешной работы с CMS пользователю потребуются знания верстки и серверного языка.
- Наличие проблем с переносом проектов – как правило, для успешного переноса программного решения в другое место потребуется повторить полную установку сайта;
- Наличие затрат, связанных с дополнительным контентом – за различные расширения для CMS придется заплатить деньги, причем в случае с CMS стоимость расширений будет выше, чем на конструкторах.

Фреймворк — это набор программных решений, упрощающих создание и поддержку технически сложных или нагруженных проектов. Фреймворк включает в себя только базовый набор библиотек и надстроек, а реализацией клиентской и серверной части занимаются разработчики.

Преимущества:

- Гибкость – с помощью фреймворков можно реализовать весь необходимый функционал, и сделан он будет под нужды конкретного заказчика;
- Широкие возможности продвижения – сайт, разработанный на фреймворке проще продвинуть в поисковой системе в отличие от CMS и конструкторов;
- Более высокая производительность в сравнении с CMS – реализация одного и того же проекта на фреймворке будет работать быстрее, чем на CMS, за счет отсутствия ненужных программных модулей.

Недостатки:

- Цена разработки – разработка сайта с использованием фреймворка как правило обходится дороже, чем разработка сайта на CMS или конструкторе;
- Высокий порог входа – разработка сайта с использованием фреймворка требует наличия большого количества знаний в области разработки и проектирования приложений;
- Большие временные затраты – разработка проекта с использованием фреймворков займет больше времени, чем разработка проекта на CMS или конструкторе [1,2].

Исходя из всех преимуществ и недостатков вышеперечисленных способов разработки сайтов, было принято решение использовать веб-фреймворк для реализации проекта.

Исследования, проведенные Stack Overflow показывают, что в тройке самых популярных бэкенд-фреймворков находятся:

- Express;
- Spring;
- Django.

Express/Node.js – серверная платформа, являющаяся частью стека технологий, которые покрывают все нужды разработки и базируются на языке Javascript. Основой особенностью Node.js является возможность исполнения серверного кода на стороне клиента.

Преимущества Node.js:

- Разработка клиентской и серверной части на едином языке JavaScript;
- Модульность;
- Масштабируемость;

- Высокая производительность, которая достигается за счет компиляции исходного кода.

Spring – фреймворк, предназначенный для разработки бэкенд-приложений. Данный фреймворк базируется на языке Java и используется для разработки микросервисов.

Преимущества Spring:

- Возможность декларативного программирования;
- Слабосвязанная архитектура;
- Свободное связывание и тестирование модулей;
- Поддержка различных способов конфигурации.

Django – бэкенд-фреймворк с открытым исходным кодом, разработанный на языке программирования – Python. Как известно, Python — это один из языков, которые любят разработчики. Такая любовь является одной из причин широкой популярности данного фреймворка.

Преимущества Django:

- Относительная простота разработки;
- Поддержка паттерна MVC;
- Производительность;
- Универсальность;
- Наличие мощных встроенных инструментов, обеспечивающих безопасность веб-приложения;
- Кроссплатформенность;
- Масштабируемость;
- Наличие большого сообщества [3, 4].

В результате сравнения фреймворков был выбран Django, так как для разрабатываемых проектов очень важны безопасность, масштабируемость и производительность.

Список литературы

1. Крупнейшая платформа для предпринимателей и высококвалифицированных специалистов малых, средних и крупных компаний – vc.ru [Электронный ресурс]. Режим доступа: <https://vc.ru/dev/78714-3-sposoba-razrabotki-veb-sayta>, свободный (Дата обращения 21.01.2021);
2. WEB CREATOR [Электронный ресурс]. Режим доступа: https://web-creator.ru/articles/about_frameworks, свободный (Дата обращения 21.01.2021);
3. Сайт с практическими курсами программирования на Java – TOP JAVA [Электронный ресурс]. Режим доступа: <https://topjava.ru/blog/spring-framework-vs-spring-boot-differences>, свободный (Дата обращения 21.01.2021);
4. Хабр [Электронный ресурс]. Режим доступа: <https://habr.com/ru/company/ruvds/blog/519478/>, свободный (Дата обращения 21.01.2021).