**Глава 2.** Анализ существующих технических решений и выбор варианта, наиболее полно удовлетворяющего требованиям пользователей

2.1 Front-End

2.1.1. Язык разметки

2.1.1.1. XHTML

2.1.1.2. HTML

2.1.2. Каскадные таблицы стилей

2.1.2.1. CSS

2.1.2.2. SASS / SCSS

2.1.2.3. LESS

2.1.3. Язык скриптов

2.1.3.1. JavaScript

2.1.3.2. jQuery

2.1.4. Сборщики проектов

2.1.4.1. Grunt

2.1.4.2. GulpJS

2.2 Back-End

2.2.1. Wordpress

2.2.2. Joomla

2.2.3. X3M

2.3 Editors

**2.1. Front-End**

**2.1.1. Язык разметки**

Язык разметки (текста) в компьютерной терминологии — набор символов или последовательностей, вставляемых в текст для передачи информации о его выводе или строении. Принадлежит классу компьютерных языков. Текстовый документ, написанный с использованием языка разметки, содержит не только сам текст (как последовательность слов и знаков препинания), но и дополнительную информацию о различных его участках — например, указание на заголовки, выделения, списки и т. д.

В более сложных случаях язык разметки позволяет вставлять в документ интерактивные элементы и содержание других документов.

**2.1.1.1. XHTML**

XHTML (англ. extensible hypertext markup language — расширяемый язык гипертекстовой разметки) — семейство языков разметки веб-страниц на основе XML, повторяющих и расширяющих возможности HTML 4. Спецификации XHTML 1.0 и XHTML 1.1 являются рекомендациями консорциума Всемирной паутины. Развитие XHTML остановлено; новые версии XHTML не выпускаются; рекомендуется использовать HTML.

XHTML соответствует спецификации SGML, поскольку XML является её подмножеством.

Существует три типа документов XHTML: strict, transitional и frameset. Наиболее употребительной и универсальной из версий XHTML является переходная (англ. transitional), поскольку она позволяет использовать iframe (включение содержимого одной веб-страницы в другую) и атрибут target у ссылок (для указания того, например, что ссылке необходимо открываться в новом окне). Фреймовая версия (англ. frameset) представляет собой расширенный вариант transitional и добавляет к нему, как следует из названия, возможность установки frameset вместо body. DTD ((англ. document type definition) определение типа документа) строгой версии XHTML (англ. strict) не содержит многих тегов и атрибутов, описанных в DTD transitional и признанных устаревшими.

**2.1.1.2. HTML**

HTML (от англ. HyperText Markup Language — «язык гипертекстовой разметки») — стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML. Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

Язык HTML является приложением SGML (стандартного обобщённого языка разметки) и соответствует международному стандарту ISO 8879.

Язык XHTML является более строгим вариантом HTML, он следует всем ограничениям XML и, фактически, XHTML можно воспринимать как приложение языка XML к области разметки гипертекста.

Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам HTTP или HTTPS, в виде простого текста или с использованием шифрования.

Язык HTML был разработан британским учёным Тимом Бернерсом-Ли приблизительно в 1986—1991 годах в стенах ЦЕРНа в Женеве в Швейцарии. HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области вёрстки. HTML успешно справлялся с проблемой сложности SGML путём определения небольшого набора структурных и семантических элементов — дескрипторов. Дескрипторы также часто называют «тегами». С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа, в HTML внесена поддержка гипертекста. Мультимедийные возможности были добавлены позже.

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). В идеале, текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью (цветной экран современного компьютера, монохромный экран органайзера, ограниченный по размерам экран мобильного телефона или устройства и программы голосового воспроизведения текстов). Однако современное применение HTML очень далеко от его изначальной задачи.

Текстовые документы, содержащие разметку на языке HTML (такие документы традиционно имеют расширение .html или .htm), обрабатываются специальными приложениями, которые отображают документ в его форматированном виде. Такие приложения, называемые «браузерами» или «интернет-обозревателями», обычно предоставляют пользователю удобный интерфейс для запроса веб-страниц, их просмотра (и вывода на иные внешние устройства) и, при необходимости, отправки введённых пользователем данных на сервер.

**2.1.2. Каскадные таблицы стилей**

**2.1.2.1. CSS**

CSS (англ. Cascading Style Sheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL.

CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля.

Правила CSS пишутся на формальном языке CSS и располагаются в таблицах стилей, то есть таблицы стилей содержат в себе правила CSS. Эти таблицы стилей могут располагаться как в самом веб-документе, внешний вид которого они описывают, так и в отдельных файлах, имеющих формат CSS.

**2.1.2.2. SASS / SCSS**

Sass (Syntactically Awesome Stylesheets) — это скриптовый метаязык, который компилируется в обычные CSS-стили.

Расширения SASS-файлов могут быть .sass и .scss — это зависит от выбранного синтаксиса. Браузер, впрочем, не понимает ни одного из них, поэтому для взаимопонимания нужно использовать компилятор. Его задача — привести SASS в понятный классический CSS, который будет распознан любым браузером.

Роль компилятора может выполнять серверный js или программа, установленная у вас на рабочей машине и мониторящая изменения в рабочих файлах.

Независимо от синтаксиса, SCSS обратно совместим с CSS. То есть любой CSS обязательно будет валидным SCSS-кодом.

Из-за отсутствия скобок и точек с запятой обратной совместимости у SASS-синтаксиса с CSS нет.

Преимущества SASS:

* SASS позволяет назначать переменные — и это одно из ключевых преимуществ. Переменная, по аналогии с php, начинается со знака доллара ($), значения присваиваются при помощи двоеточия. Переменные в Sass можно разделить на 4 типа:
  + число (int)
  + строка (string)
  + логический тип (да/нет, boolean)
  + цвета (имя, имена)
* Вложенные правила (nesting): возможность вкладывать правила в другие правила, что при компиляции создает нечто вроде пространства имен. Визуально вложенные в Sass внутрь родительского элемента правила будут сохранены с той же иерархией с учетом правил CSS.
* Дополнения (mixin): части кода, которые в CSS обычно приходилось дублировать, здесь можно сохранить в отдельной переменной и вставлять в нужных местах. Компилятор, встретив такую переменную, сохранит вместо нее нужный кусок кода.
* Наследование (extend): создав однажды любое правило, мы можем использовать его внутри другого. Наследуемый элемент получит все свойства исходного класса, которые мы можем дополнить любыми другими.

**2.1.2.3. LESS**

LESS – это надстройка над CSS. Это значит, что любой CSS код – это валидный LESS, но дополнительные элементы LESS не будут работать в простом CSS. Это замечательно, потому что существующий CSS уже является работоспособным LESS кодом, что уменьшает порог вхождения в новую технологию.

LESS добавляет много нужных динамических свойств в CSS. Он вводит переменные, операции, function-like элементы и примеси. Возможность писать таблицы стилей модульно избавит вас от многих хлопот.

Есть два способа использования LESS. Вы можете создать LESS файл и конвертировать его при помощи Javascript на лету или скомпилировать его заранее и использовать получившийся CSS файл.

LESS обеспечивает сравнимую функциональность с SASS, однако имеет ряд недостатков:

* не содержит логики/циклов
* не имеет helper (SASS позволяет использовать Compass)
* не имеет библиотеки адаптивных стилей (SASS позволяет использовать MQ-SASS)
* не позволяет проводить математические операции
* не позволяет генерировать спрайты

**2.1.4. Сборщики проектов (таск-менеджеры)**

Таск-менеджер — небольшое приложение, которое используется для автоматизации скучных и рутинных, но, от того, не менее важных, задач, которые приходится постоянно выполнять в процессе разработки проекта. Такие задачи включают в себя, к примеру, запуск модульных тестов, конкатенацию файлов, минификацию, препроцессинг CSS. Довольно простая идея, которая позволяет сэкономить очень много времени, и помогает сохранять фокус на задачах, связанных непосредственно с разработкой проекта.

**2.1.4.1. Grunt**

Grunt — это инструмент для сборки javascript проектов из командной строки с использованием задач. Релиз вышел совсем недавно, автор Ben «Cowboy» Alman, проект есть на github.

С Grunt-ом, JS разработчики имеют очень гибкий инструмент под рукой, который помогает при автоматизации повторяющихся задач. После настройки, вы можете настроить задачи по времени, сделать это волшебство в фоновом режиме, или интегрировать Grunt в ваш IDE или редактор для кодирования как внешнюю команду разработки (которая прекрасно работает в Sublime Text, например).

Код проверки и тестирования повышает общее качество и поможет вам поддерживать постоянный стандарт кодирования, что делает Grunt также идеальным инструментом для команд, работающих на крупномасштабных проектах.

Grunt устанавливается как NPM (Node Package Manager) модуль.

**2.1.4.2. GulpJS**

Gulp [/gʌlp/ /галп/] — система потоковой сборки. Он позволяет автоматизировать повторяющиеся задачи (сборка и минификация CSS- и JS-файлов, запуск тестов, перезагрузка браузера и другие). Тем самым Gulp ускоряет и оптимизирует процесс веб-разработки.

Сам по себе Gulp очень скуден на возможности. Но все, что вам нужно вынесено в отдельные плагины.

Gulp.task определяет наши задачи. В качестве аргументов принимает название, зависимости (массив) и функцию (основные действия). Зависимостей может и не быть.

Gulp.src указывает на файлы, которые мы хотим использовать. Он использует .pipe доступа к файлам через плагины.

Gulp.dest указывает на папку, в которую мы хотим сохранить измененные файлы.

Gulp.src и gulp.dest используется для простой копии файлов.

В gulp встроена система реагирования на изменения файлов (gulp.watch).

Благодаря своей архитектуре, основанной на потоках, справляется с идентичными заданиями на 50-80% быстрее, чем Grunt, а также содержит меньше кода в списке задач.

**2.1.3. Язык скриптов**

Сценарный язык (язык сценариев, жарг. скрипто́вый язык, от англ. scripting language) — высокоуровневый язык сценариев (англ. script) — кратких описаний действий, выполняемых системой. Разница между программами и сценариями довольно размыта.

Сценарий — это программа, имеющая дело с готовыми программными компонентами.

**2.1.3.1. JavaScript**

JavaScript изначально создавался для того, чтобы сделать web-странички «живыми». Программы на этом языке называются скриптами. В браузере они подключаются напрямую к HTML и, как только загружается страничка – тут же выполняются.

Программы на JavaScript – обычный текст. Они не требуют какой-то специальной подготовки.

В этом плане JavaScript сильно отличается от другого языка, который называется Java.

JavaScript может выполняться не только в браузере, а где угодно, нужна лишь специальная программа – интерпретатор. Процесс выполнения скрипта называют «интерпретацией».

Во все основные браузеры встроен интерпретатор JavaScript, именно поэтому они могут выполнять скрипты на странице. Но, разумеется, JavaScript можно использовать не только в браузере. Это полноценный язык, программы на котором можно запускать и на сервере, и даже в стиральной машинке, если в ней установлен соответствующий интерпретатор.

Современный JavaScript – это «безопасный» язык программирования общего назначения. Он не предоставляет низкоуровневых средств работы с памятью, процессором, так как изначально был ориентирован на браузеры, в которых это не требуется.

Что же касается остальных возможностей – они зависят от окружения, в котором запущен JavaScript. В браузере JavaScript умеет делать всё, что относится к манипуляции со страницей, взаимодействию с посетителем и, в какой-то мере, с сервером:

* Создавать новые HTML-теги, удалять существующие, менять стили элементов, прятать, показывать элементы и т.п.
* Реагировать на действия посетителя, обрабатывать клики мыши, перемещения курсора, нажатия на клавиатуру и т.п.
* Посылать запросы на сервер и загружать данные без перезагрузки страницы (эта технология называется "AJAX").
* Получать и устанавливать cookie, запрашивать данные, выводить сообщения…

JavaScript – быстрый и мощный язык, но браузер накладывает на его исполнение некоторые ограничения…

Это сделано для безопасности пользователей, чтобы злоумышленник не мог с помощью JavaScript получить личные данные или как-то навредить компьютеру пользователя.

Этих ограничений нет там, где JavaScript используется вне браузера, например на сервере. Кроме того, современные браузеры предоставляют свои механизмы по установке плагинов и расширений, которые обладают расширенными возможностями, но требуют специальных действий по установке от пользователя

Большинство возможностей JavaScript в браузере ограничено текущим окном и страницей.

* JavaScript не может читать/записывать произвольные файлы на жесткий диск, копировать их или вызывать программы. Он не имеет прямого доступа к операционной системе. Современные браузеры могут работать с файлами, но эта возможность ограничена специально выделенной директорией – «песочницей». Возможности по доступу к устройствам также прорабатываются в современных стандартах и частично доступны в некоторых браузерах.
* JavaScript, работающий в одной вкладке, не может общаться с другими вкладками и окнами, за исключением случая, когда он сам открыл это окно или несколько вкладок из одного источника (одинаковый домен, порт, протокол).
* Из JavaScript можно легко посылать запросы на сервер, с которого пришла страница. Запрос на другой домен тоже возможен, но менее удобен, т. к. и здесь есть ограничения безопасности.

**2.1.3.2. jQuery**

jQuery — библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX. Сейчас разработка jQuery ведётся командой jQuery во главе с Джоном Резигом.

Основной целью создания jQuery Резиг видел возможность закодировать многоразовые куски кода, которые позволят упростить JavaScript и использовать их так, чтобы не беспокоиться о кросс-браузерных вопросах. Библиотека была представлена общественности на компьютерной конференции «BarCamp» в Нью-Йорке в 2006 году.

Библиотека jQuery содержит функциональность, полезную для максимально широкого круга задач. Тем не менее, разработчиками библиотеки не ставилась задача совмещения в jQuery функций, которые подошли бы всюду, поскольку это привело бы к большому коду, бо́льшая часть которого не востребована. Поэтому была реализована архитектура компактного универсального ядра библиотеки и плагинов. Это позволяет собрать для ресурса именно ту JavaScript-функциональность, которая на нём была бы востребована.

Работу с jQuery можно разделить на 2 типа:

* Получение jQuery-объекта с помощью функции $(). Например, передав в неё CSS-селектор, можно получить jQuery-объект всех элементов HTML, попадающих под критерий и далее работать с ними с помощью различных методов jQuery-объекта. В случае, если метод не должен возвращать какого-либо значения, он возвращает ссылку на jQuery объект, что позволяет вести цепочку вызовов методов согласно концепции текучего интерфейса.
* Вызов глобальных методов у объекта $, например, удобных итераторов по массиву.

**2.2. Back-End**

**2.2.1. Wordpress**

WordPress – одна из самых популярных в мире информационных систем для обеспечения и организации процесса создания, редактирования и управления содержимым ( CMS ). Она предоставляет все необходимые функции для создания блога, новостного сайта (и, разумеется, для сателлитов). Эта современная платформа сориентирована на удобство использования, сетевые стандарты и, естественно, эстетическую привлекательность. WordPress блог бесплатен и свободен к распространению. Язык написания – РНР, в качестве базы данных используется MySQL, распространяется под универсальной общедоступной лицензией GNU.

Процесс установки Wordpress, очень прост и сводится к тому, что после скачивания желаемой версии Wordpress, нужно разархивировать и скопировать пакет в нужную директорию на сервер

Возможности WordPress:

* Информационная система позволяет создавать публикации при помощи сторонних программ.
* Установка и настройка не представляют никакой сложности (см. раздел выше).
* Поддерживает функции создания автоматически обрабатываемых списков обновлений RSS, Atom, trackback, pingback.
* Уникальная простота системы взаимодействия с кодом подключаемых модулей (плагинов).
* Реализация «тем», как наборов файлов – шаблонов (на РНР), весьма заметно увеличивает скорость и гибкость.
* Существует возможность легко изменять внешний вид и способы вывода данных за счет поддержки «тем».
* Неисчерпаемые объемы библиотек «тем» и «плагинов»
* Доступна реализация сложных решений за счет заложенного потенциала архитектуры.
* Предлагается использование человеко-понятных URL.
* Присутствует система русского перевода.

**2.2.2. Joomla**

Joomla! (произносится джу́мла) — система управления содержимым (CMS), написанная на языках PHP и JavaScript, использующая в качестве хранилища базы данных СУБД MySQL или другие стандартные промышленные реляционные СУБД. Является свободным программным обеспечением, распространяемым под лицензией GNU GPL.

Система управления содержимым Joomla! является ответвлением широко известной CMS Mambo. Команда независимых разработчиков отделилась от проекта Mambo по причине несогласия в экономической политике. 16 сентября 2005 года в свет вышла первая версия Joomla!, являющаяся по сути переименованной Mambo 4.5.2.3 и включающая в себя исправления найденных на тот момент ошибок и уязвимостей.

CMS Joomla! включает в себя минимальный набор инструментов при начальной установке, который дополняется по мере необходимости. Это снижает загромождение административной панели ненужными элементами, а также снижает нагрузку на сервер и экономит место на хостинге.

Joomla! позволяет отображать интерфейс фронтальной и административной части на любом языке. Каталог расширений содержит множество языковых пакетов, которые устанавливаются штатными средствами администрирования. Доступны пакеты русского, украинского, белорусского и ещё некоторых языков пост-советского пространства.

Основные возможности:

* Функциональность можно увеличивать с помощью дополнительных расширений (компонентов, модулей и плагинов).
* Имеется модуль безопасности для многоуровневой аутентификации пользователей и администраторов (используется собственный алгоритм аутентификации и «ведения» сессий).
* Система шаблонов позволяет легко изменять внешний вид сайта: расположение модулей, шрифты и другое. Можно предоставить пользователям выбирать одно из нескольких отображений. В сети существует огромный выбор готовых шаблонов, как платных, так и бесплатных. Также существует программное обеспечение для самостоятельного создания оригинальных шаблонов.
* Предусмотрены настраиваемые схемы расположения модулей, включая левый, правый, центральный и любое другое произвольное положения блока. При желании содержимое модуля можно включить в содержимое материала.
* К преимуществам системы можно отнести то, что все компоненты, модули, плагины и шаблоны можно написать самому, разместить их в структурированном каталоге расширений или отредактировать существующее расширение по своему усмотрению.
* Происходит регулярный выход обновлений. Существует публичный «баг-трекер» (система отслеживания ошибок). Существуют также трекеры миграции со старых версий Joomla, трекер пожеланий расширения функциональности и так далее, где пользователи Joomla могут оставлять замечания по поводу работы CMS, которые впоследствии изучаются её разработчиками, при необходимости включающими в очередное обновление Joomla исправления, решающие те или иные проблемы.
* Начиная с версии 1.6 встроена многоязычность.
* Начиная с версии 2.5 расширена поддержка баз данных. Реализована поддержка Microsoft SQL Server, а с версии 3.0 — PostgreSQL[9]. В дальнейшем планируется добавить поддержку Oracle, SQLite.

**2.2.3. X3M**

Система управления сайтом X3M.CMS предназначена для создания интернет-проектов любой сложности. В основу административной части системы положен принцип асинхронного взаимодействия с пользователем (AJAX), X3M.CMS работает в браузере как обычное desktop приложение. Используя данный принцип, пользователи быстро привыкают к простому интерфейсу системы. Кроме того, значительно увеличивается скорость работы с сайтом. Таким образом система не требует каких либо специальных навыков от конечного пользователя — необходимым минимумом знаний может является программный пакет Microsoft Office и работа с «Проводником Windows».

Преимущества системы:

* Скорость системы определяется "нагруженностью" проекта, для того чтобы оценить скорость работы X3M.CMS можно посмотреть графики тестирования в зависимости от нагрузки. По всем базовым показателям система превосходит большинство существующих аналогов.
* Система удовлетворяет современным требованиям по надежности, и имеет ряд механизмов защиты от несанкционированного доступа, систему резервирования данных и распределения прав доступа для различных групп пользователей. X3M работает устойчиво даже после заражения inline (вирусов распространяющихся по FTP) вирусами, которые в последнее время очень распространены.
* X3M построена на модульной архитектуре и состоит из ядра и 20 различных модулей, количество и функциональность модулей постоянно растет. Данная архитектура позволяет гибко подстраиваться под задачи любого проекта, обеспечивая надежный и доступный интерфейс управления.
* Сквозная система авторизации: аутентификация пользователей и администраторов осуществляется через единую систему авторизации. А система разграничения пользователей осуществляется по логину и паролю, который храниться в зашифрованном виде. Также существует возможность подключения защищенного соединения с использованием SSL-сертификатов.
* Система разграничения доступа: многоуровневая система разграничения доступа позволяет ограничить доступ по целому ряду признаков: по доступу к сайту, по доступу к модулю (на чтение, редактирование, публикацию, удаление), по страницам на сайте, а также по конкретной информации находящейся в каждом модуле.
* Система диагностики ошибок: система фиксирует все ошибки, которые возникают на сайте, что позволяет администраторам вовремя реагировать на возникающие проблемы в системе или на попытку взлома.
* Журнал операций: все операции администраторов системы фиксируются. Журналирование позволяет анализировать информацию о работе сотрудников по наполнению сайта информацией. Специальные фильтры позволяют эффективно оценить работу администраторов по необходимым параметрам. Используемые при разработке стандарты кодирования и архитектура исключают появление таких уязвимостей, как: SQL-injection или cross-site scripting.

**2.3 Редактор кода**

**Sublime Text 3**

Sublime Text — кроссплатформенный проприетарный текстовый редактор. Поддерживает плагины на языке программирования Python.

Программа часто используется как редактор исходного кода.

Официальный сайт позволяет бесплатно и без ограничений ознакомиться с продуктом, однако уведомляет о необходимости приобретения лицензии.

Возможности программы:

* Быстрая навигация (Goto Anything)
* Командная палитра (Command Palette)
* Python Plugin API
* Одновременное редактирование (Split Editing)
* Высокая степень настраиваемости (Customize Anything)
* Кросс-платформенность (Windows, OS X, Linux)

Sublime Text поддерживает большое количество языков программирования и имеет возможность подсветки синтаксиса для C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL и XML. В дополнение к тем языкам программирования, которые включены в первоначальный пакет, пользователи имеют возможность загружать дополнения для поддержки дополнительных языков.

Пользователи видят весь свой код в правом верхнем углу экрана и при клике на него могут переходить на различные его части.

Редактор предлагает несколько режимов экрана. Один из них включает от 1 до 4 панелей, с помощью которых можно показывать до четырёх файлов одновременно. Полноценный (free modes) режим показывает только один файл без каких-либо дополнительных вокруг него меню.

Когда пользователь набирает код, Sublime Text, в зависимости от используемого языка, будет предлагать различные варианты для завершения записи. Он также автоматически завершает созданные пользователем переменные.

Эта особенность позволяет пользователям запускать код для некоторых языков изнутри самого редактора, при этом избавляя от необходимости переключаться из командной строки и обратно. Это дополнение также может быть настроено для автоматического построения кода каждый раз при сохранении кода.

Sublime Text может быть оснащён менеджером пакетов, который позволяет пользователю находить, устанавливать, обновлять и удалять пакеты без перезагрузки Sublime Text. Менеджер пакетов поддерживает установленные пакеты в актуальном состоянии с использованием функции авто обновления и загружает пакеты с GitHub, BitBucket и настраиваемых JSON channel/repository систем. Он также поддерживает пакеты, клонированные с GitHub и BitBucket с использованием Git и Hg, и предоставляет команды для активации и деактивации пакетов. Менеджер пакетов также включает команды для упаковки любой директории пакета в .sublime-package файл.