

## Задание 1.3

После анализа исходных файлов сайта кафедры было выявлено, что он построен на основе CMS Grav, написанная на языке PHP и основанная на модели плоских файлов, то есть не использующая какую-либо базу данных. Хранит информацию веб-сайта в текстовых файлах с использованием markdown-разметки. Для шаблонов сайта используется Twig, конфигурация сохраняется в YAML-файлах.

В проекте используются следующие плагины:

1. add-page-by-form (<https://github.com/bleutzinn/grav-plugin-add-page-by-form>)

Он позволяет пользователям добавить новую страницу, заполнив форму. Этот плагин использует возможности пользовательского frontmatter (это блок метаданных, который располагается в начале Markdown-файла. Он используется для описания и хранения информации о документе (метаданные документа), такой как заголовок, описание, автор, дата создания, дата публикации, категория, теги и другие параметры). Задав собственные переменные во frontmatter страницы формы и предоставив пользователям возможность переопределять значения переменных, заполняя соответствующие поля формы, можно перенести эти данные во frontmatter новой страницы.

Передача как настроек по умолчанию, так и значений полей формы, введенных конечным пользователем, на frontmatter новой страницы делает решение чрезвычайно настраиваемым.

Смешивая параметры по умолчанию и конфигурируя форму страницы, можно в значительной степени контролировать внешний вид и

поведение новой добавленной страницы, используя переменные frontmatter, присутствующие на новой странице в шаблоне Twig.

## 2. admin (<https://github.com/getgrav/grav-plugin-admin>)

Этот плагин администратора для Grav представляет собой пользовательский интерфейс HTML, который обеспечивает удобный способ настройки Grav и легкого создания и изменения страниц. Это совершенно необязательный плагин и никоим образом не требуется для эффективного использования Grav. Фактически, администратор намеренно предоставляет ограниченное представление, чтобы оно оставалось простым в использовании и не перегруженным.

## 3. admin-addon-user-manager (<https://github.com/david-szabo97/grav-plugin-admin-addon-user-manager>)

Простое расширение панели администратора, которое добавляет возможность управления пользователями и группами.

## 4. admin-power-tools (<https://github.com/Flamenco/admin-power-tools>)

Мощные инструменты для плагина администратора, включая «Редактировать этот раздел», «Добавить улучшения страницы», «Изменить порядок дочерних страниц» и «Отчеты».

## 5. archives (<https://github.com/getgrav/grav-plugin-archives>)

Плагин Grav, который автоматически добавляет таксономию «month\_year» ко всем страницам. Затем он предоставляет шаблон parts\archives.html.twig, который можно включить в боковую панель блога, который затем сможет создавать ссылки, которые будут отображать страницы за этот месяц/год.

## 6. bootstrapper (<https://github.com/getgrav/grav-plugin-bootstrapper>)

Плагин Grav, который можно использовать в качестве зависимости для других тем и плагинов для загрузки ресурсов Bootstrap Framework.

Основная цель этих плагинов — позволить теме Bootstrap зависеть от загрузочного CSS/JS и позволить плагину обновляться независимо от самой темы.

7. breadcrumbs (<https://github.com/getgrav/grav-plugin-breadcrumbs>)

Добавляет ссылки на предыдущие страницы в иерархии.

8. core-service-manager (<https://github.com/Flamenco/grav-plugin-core-service-manager>)

Он добавляет уровень менеджера сервисов в Grav API. Этот плагин сам по себе не предоставляет никаких функций конечному пользователю.

Он расширяет ядро Grav сервисами и функциями, которые могут использоваться другими плагинами.

9. data-manager (<https://github.com/getgrav/grav-plugin-data-manager>)

Плагин диспетчера данных для Grav добавляет возможность визуализации данных. Это особенно полезно для плагинов администратора и форм. Дополнительные плагины могут хранить содержимое данных, а плагин данных, правильно настроенный, также может отображать их данные.

10. datetools (<https://github.com/kalebheitzman/grav-plugin-datetools>)

Этот плагин Grav предоставляет инструменты даты, которые можно использовать внутри Twig для фильтрации страниц. С выпуском Grav 0.9.13 в анализ коллекции были введены startDate и endDate.

11. editor (<https://github.com/Flamenco/grav-plugin-editor>)

Позволяет редактировать исходные файлы Grav прямо в браузере.

12. email (<https://github.com/getgrav/grav-plugin-email>)

Позволяет отправлять письма используя пакет symfony/mailer.

13. error (<https://github.com/getgrav/grav-plugin-error>)

Позволяет использовать кастомные страницы для отображения ошибок.

#### 14.events (<https://github.com/kalebheitzman/grav-plugin-events>)

Позволяет создавать одиночные и повторяющиеся события, используя event: frontmatter на любой странице по выбору. Плагин «События» предоставляет шаблоны для календаря, событий и представлений событий, а также поддерживает создание событий с помощью плагина администратора.

#### 15.external-links (<https://github.com/Sommerregen/grav-plugin-external-links>)

Плагин для GetGrav.org, используемый для различия внутренних и внешних ссылок. Он также включает возможность добавления rel="nofollow" к ссылкам и определяет, как внешние ссылки должны открываться с помощью атрибута target.

#### 16.feed (<https://github.com/getgrav/grav-plugin-feed>)

Позволяет создавать ленты для страниц.

#### 17.filesource (<https://github.com/anza/grav-plugin-filesource>)

Позволяет прикреплять текстовые файлы на страницу.

#### 18.flex-objects (<https://github.com/trilbymedia/grav-plugin-flex-objects>)

Плагин, который позволяет создавать собственные коллекции объектов, которые можно изменять с помощью операций CRUD через плагин администратора, чтобы легко управлять большими наборами данных, которые не имеют смысла в качестве простых файлов конфигурации YAML или страниц Grav. Эти объекты определяются схемами, написанными на YAML, и отображаются с помощью набора файлов веток. Кроме того, как объекты, так и коллекции можно настраивать с помощью классов PHP, что позволяет определять сложное поведение и отношения между объектами.

19.form (<https://github.com/getgrav/grav-plugin-form>)

Позволяет создавать и использовать формы.

20.github (<https://github.com/getgrav/grav-plugin-github>)

Вспомогательный плагин для работы GitHub API.

21.gravstrap (<https://github.com/giansi/gravstrap>)

Плагин Grav, который предоставляет компоненты Bootstrap в виде шорткодов. Шорткод имеет простую структуру и добавляется на страницу Grav для визуализации полного компонента, такого как простая кнопка, или более сложного, такого как карусель или панель навигации, до полного сложного модуля, такого как портфолио.

22.highlight (<https://github.com/getgrav/grav-plugin-github>)

Добавляет подсветку кода, используя Highlight.js

23.instagram (<https://github.com/artifex404/grav-plugin-instagram>)

Позволяет добавить ленту Instagram на страницу

24.langswitcher (<https://github.com/getgrav/grav-plugin-langswitcher>)

Добавляет ссылки переключения языка интерфейса Grav

25.leaflet (<https://github.com/magikcypress/grav-plugin-leaflet>)

Позволяет использовать Leaflet.js, который позволяет использовать интерактивные карты

26.lightslider (<https://github.com/getgrav/grav-plugin-lightslider>)

Позволяет использовать слайдер с помощью одноименного jQuery-плагина

27.login (<https://github.com/getgrav/grav-plugin-login>)



Добавляет в Grav вход в систему, базовый ACL и сообщения всего сеанса. Он предназначен для обеспечения защиты внешнего и административного контента в Grav.

28.markdown-collapsible (<https://github.com/X-Ryl669/grav-plugin-markdown-collapsible>)

Позволяет создавать сворачиваемые блоки текста с помощью Markdown

29.markdown-color (<https://github.com/getgrav/grav-plugin-markdown-color>)

Позволяет использовать цветной текст с помощью Markdown

30.markdown-details (<https://github.com/bitstarr/grav-plugin-markdown-details>)

Позволяет создавать сворачиваемые блоки текста с помощью Markdown

31.markdown-fontawesome (<https://github.com/yoshikin/grav-plugin-markdown-fontawesome>)

Позволяет использовать значки Font Awesome, встроенные в Markdown, заключая имя значка в двоеточия.

32.markdown-notices (<https://github.com/getgrav/grav-plugin-markdown-notices>)

Позволяет создавать разноцветные блоки уведомлений с помощью Markdown

33.markdown-sections

34.markdown-typography (<https://github.com/N-Parsons/grav-plugin-markdown-typography>)

Позволяет применять типографские символы во время обработки Markdown.

35.mathjax (<https://github.com/Sommerregen/grav-plugin-mathjax>)

Позволяет включать математические формулы на ваши веб-страницы, используя обозначения TeX и LaTeX и/или MathML с помощью одноименной js-библиотеки

36.optimus (<https://github.com/getgrav/grav-plugin-optimus>)

Автоматически обрабатывает медиа-изображения страницы с помощью службы сжатия без потерь Optimus.

37.page-inject (<https://github.com/getgrav/grav-plugin-page-inject>)

Позволяет вставлять целые страницы или их содержимое в другие страницы, используя простой синтаксис в стиле Markdown или, альтернативно, синтаксис короткого кода.

38.pagination (<https://github.com/getgrav/grav-plugin-pagination>)

Позволяет разделять статьи на несколько страниц.

39.problems (<https://github.com/getgrav/grav-plugin-problems>)

Находит и отображает ошибки в исходном коде

40.qrcode (<https://github.com/cworreschk/grav-plugin-qrcode>)

Использует библиотеку endroid/QrCode для создания QR-кодов.

Позволяет просто создавать и вставлять QR-коды на свой сайт, используя короткий код [qrcode]

41.random (<https://github.com/getgrav/grav-plugin-random>)

Позволяет загружать случайную статью на основе фильтров таксономии.

42.relatedpages (<https://github.com/getgrav/grav-plugin-relatedpages>)

Вычисляет связанные страницы для текущей страницы. Он использует гибко настраиваемый набор параметров для определения оценки для каждой страницы, которая настраивается для просмотра и сравнения.

#### 43.shortcode-core (<https://github.com/getgrav/grav-plugin-shortcode-core>)

Позволяют разрабатывать простые, но мощные плагины для коротких кодов, которые используют общий формат, используемый WordPress и BBCode. Основной плагин загружает необходимые библиотеки и запускает новое событие, которое могут использовать другие плагины. Он также предоставляет механизм добавления ресурсов CSS/JS, которые кэшируются, чтобы короткие коды могли эффективно работать, даже если обработанное содержимое страницы кэшируется. Это гарантирует, что короткие коды обрабатываются только один раз и не влияют на производительность из-за выполнения ненужной работы на каждой странице.

#### 44.shortcuts

#### 45.simplesearch (<https://github.com/getgrav/grav-plugin-simplesearch>)

Добавляет возможности поиска. По умолчанию он может искать по заголовкам страниц, контенту, таксономии, а также по необработанному заголовку страницы.

#### 46.table-importer (<https://github.com/Perlkonig/grav-plugin-table-importer>)

Позволяет импортировать CSV, JSON и YAML таблицы на страницу.

#### 47.taxonomylist (<https://github.com/getgrav/grav-plugin-taxonomylist>)

Генерирует список связанных тегов, собранных по всему сайту.

#### 48.tidyhtml (<https://github.com/sourcesoldier/grav-plugin-tidyhtml>)

Интегрирует TidyPHP

#### 49.toc (<https://github.com/Sommerregen/grav-plugin-toc>)

Автоматически генерирует (уменьшенное) оглавление на основе специальных маркеров в документе и добавляет его в результирующий HTML-документ.



50.youtube (<https://github.com/getgrav/grav-plugin-youtube>)

Преобразует ссылку на видео в плеер с ним

В дизайне используются различные стилевые шаблоны Bootstrap и FontAwesome. В качестве шрифта на сайте используется Monsterrat.