

Общее описание задачи

Задача: необходимо разработать инструмент (скрипт) для автоматической генерации html страниц сайта на основании данных из таблицы Google Sheets.

Общие вводные данные:

Парсеру (скрипту) на вход подается таблица в Google Sheets, которая создается заранее и может быть отредактирована.

Контент для страниц подтягивается из определенных столбцов и с определенных вкладок.

(!) Чтение парсером контента происходит по таблице слева направо (<https://prnt.sc/6ph05zi33Sfy>)

(!) Контент в главной таблице (откуда скрипт будет брать контент) будет представлено в виде переменных (на основании формул Google Sheets), которые в свою очередь будут формироваться с контента из других (внутренних вкладок)

После обхода (обработки) данных, система формирует список страниц и публикует их следуя правилам и порядку формирования контента на странице согласно файла.

Важные моменты:

(!) Если в какой-то ячейке пустое значение - скрипт обрабатывает его (не выдавая ошибки) и в генерации контента он не участвует.

Другими словами, если значение ячейки пустое - соответствующий блок не выводится на странице и не нарушает структуру страницы (не оставляет пустых мест)

(!) Скрипт совершает "обход" данной таблицы по запросу (по переходу на определенный url, например `/?import_pages`).

Желательно предусмотреть возможность обновления определенного диапазона строк (например `/?import_pages&start=15&end=25`).

При этом в случае внесения изменений страницы должны быть обновлены в соответствии с таблицей.

Дефолтные параметры:

- значение атрибута **meta name="robots"** должно соответствовать значению **"index, follow"**
- значение атрибута **rel="canonical"** должно соответствовать основному URL страницы, которая должна отдавать 200 ответ сервера.

- хлебные крошки должны формироваться автоматически и соответствовать местоположению юзера на сайте.
- все сгенерированные страницы должны содержать микроразметку Schema.org (пример кода в пунктах ниже)
- после публикации, все страницы должны сохраняться на хостинге

Параметры, которые парсер будет брать из файла Google Sheets:

Типы страниц:

- Main Page <https://parhaatuudetnettikasinot.fi/>
- Страница Казино <https://parhaatuudetnettikasinot.fi/nettikasino/kingmaker-kasino/>
- Страница Слота <https://parhaatuudetnettikasinot.fi/kolikkopelit/plinko/>
- Страница Payments <https://parhaatuudetnettikasinot.fi/rahansiirrot/revolut/>

Доп. составляющие:

- Header
- Footer

1. Main Page

- Meta Title
- Meta Description
- title
- canonical (возможность прописать кастомный canonical для главных страниц, для всех остальных генерируется canonical сам на себя)
- Заголовок Hero
- Описание Hero
- Картинка Hero

- Текст на кнопке 1
- Текст на кнопке 2
- Заголовок блока с казино
- Заголовок блока с слотами
- Блок полного описания
- Заголовок перед таблицей
- Таблица Заголовок 1
- Таблица Заголовок 2
- Таблица контент
- Warning Заголовок 1
- Warning Описание 1
- Warning Заголовок 2
- Warning Описание 2
- Заголовок Блока FAQ
- Список вопросов и ответов

Пример верстки <https://parhaatuudetnettikasinot.fi/>

Дополнительная информация:

- в блоке Казино выводятся все казино, которые внесены в соответствующей вкладке в файле
- в блоке Слоты выводятся все слоты, которые внесены в соответствующей вкладке
- согласно текстовому блоку выводится содержание <https://monosnap.com/file/6w79E1c4ftVSVd7AvZsGhX9RfGTha6>

2. Slots Page

- Meta Title
- Meta Description
- title
- url
- Заголовок Hero
- Описание Hero
- Картинка Hero
- Картинка слота для листинга
- Дата
- Заголовок блока с казино
- Заголовок блока с слотами
- Текст на кнопке
- Warning Заголовок 1

- Warning Описание 1
- Warning Заголовок 2
- Warning Описание 2
- Блок полного описания
- Заголовок перед таблицей
- Таблица Заголовок 1
- Таблица Заголовок 2
- Таблица контент
- Заголовок Блока FAQ
- Список вопросов и ответов
- Рейтинг

Пример верстки <https://parhaatuudetnettikasinot.fi/kolikkopelit/plinko/>

Дополнительная информация:

- в блоке Казино выводятся все казино, которые внесены в соответствующей вкладке в файле
- в блоке Слоты выводятся все слоты, которые внесены в соответствующей вкладке
- согласно текстовому блоку выводится содержание <https://monosnap.com/file/6w79E1c4ftVSVd7AvZsGhX9RfGTha6>
- url формируется по шаблону /slots/**Значение URL с файла**

3. Page Payment

- Meta Title
- Meta Description
- title
- url
- Заголовок Hero
- Описание Hero
- Картинка Hero
- Дата
- Заголовок блока с казино
- Заголовок блока с слотами
- Текст на кнопке
- Warning Заголовок 1
- Warning Описание 1
- Warning Заголовок 2
- Warning Описание 2
- Блок полного описания

- Заголовок перед таблицей
- Таблица Заголовков 1
- Таблица Заголовков 2
- Таблица контент
- Заголовок Блока FAQ
- Список вопросов и ответов

Пример верстки <https://parhaatuudetnettikasinot.fi/rahansiirrot/revolut/>

Дополнительная информация:

- в блоке Казино выводятся все казино, которые внесены в соответствующей вкладке в файле
- в блоке Слоты выводятся все слоты, которые внесены в соответствующей вкладке
- согласно текстовому блоку выводится содержание <https://monosnap.com/file/6w79E1c4ftVSVd7AvZsGhX9RfGTha6>
- url формируется по шаблону /payment/**Значение URL с файла**

4. Casino Page

- Meta Title
- Meta Description
- title
- url
- Изображение казино
- Заголовок Казино
- Дата
- Краткое описание
- Плюс
- Минус
- Заголовок перед таблицей
- Таблица Заголовков 1
- Таблица Заголовков 2
- Таблица контент
- Заголовок слотов
- Список слотов казино
- Полное описание казино
- Заголовок сравнения казино
- Список казино для сравнения
- Заголовок Блока FAQ
- Список вопросов и ответов

- Список payments
- Бонус
- url-прокладка - в примере /go-to
- ссылка на оффер
- url-кросслинк - в примере /about
- Дата Казино
- Лицензия
- Налог
- Локализованный
- Без регистрации
- Рейтинг
- Оценка игроков
- Наша оценка

Пример верстки <https://parhaatuudetnettikasinot.fi/nettikasino/pistolo-casino/>

Дополнительная информация:

- в блоке Казино выводятся все казино, которые внесены в соответствующей вкладке в файле
- в блоке Слоты выводятся все слоты, которые внесены в соответствующей вкладке
- согласно текстовому блоку выводится содержание <https://monosnap.com/file/6w79E1c4ftVSVd7AvZsGhX9RfGTha6>
- url формируется по шаблону /casinos/**Значение URL с файла**

5. Header

- Logo
- Текст на кнопке
- Ссылка на оффер (кнопка header)
- url-прокладка (кнопка header)
- url-кросслинк (кнопка header)
- Меню (в файле вписываем необходимый html код)
- Языковая версия

6. Footer

- Logo
- Menu (в файле вписываем необходимый html код)
- Logo дополнительные
- Копирайт
- Языковая версия

Пример файла Google Sheets:

https://docs.google.com/spreadsheets/d/1UVwreKHrAmGbnTW6G6_8zthHE1HoAM0kDoHwDOTMOvk/edit?gid=1363817279#gid=1363817279

Технические моменты

1. Клоака - Исходящие ссылки офферов на кнопках

Что нужно: для гугл ботов+ботов Аhrefс и Юзеров реализовать разные сценарии ссылок на кнопках

Решение:

Если IP или User-Agent **НЕ** соответствуют Googlebot или Ahrefs, то настраиваем следующую схему:

- на всех кнопках, баннерах прописаны URL **домен/go-to (url-прокладка)**
- с урлов **/go-to** настроен 301 редирект на оффер

Если IP или User-Agent соответствуют Googlebot или Ahrefs, то настраиваем следующую схему:

- данным ботам на всех кнопках и баннерах показываем внутренний link на сайт - страница **/about (url-кросслинк)**

Список IP Googlebot -

<https://developers.google.com/search/apis/ipranges/googlebot.json>

Т.е. боты Googlebot или Ahrefs видят внутреннюю ссылку, указанную в файле (например, страница About). Юзеров кидает на страницу /go-to + 301 редирект на оффер.

В файле Google Docs:

Для каждой кнопки, согласно дизайну задаем все эти параметры:

- url-прокладка - в примере /go-to
- ссылка на оффер
- url-кросслинк - в примере /about

2. ЧПУ

Все url на сайте должны быть ЧПУ

3. Мультиязычность

При наличии мультиязычности необходимо добавить hreflang на страницы, что содержат другие языковые версии по рекомендациям ниже.

Шаблон формирования hreflang для языковой версии:

```
<link rel="alternate" hreflang="код языка-код страны" href="URL страницы" />
```

Важно! Код языка ставить обязательно, код страны — необязательно. Код языка ставится первым, код страны через дефис после языка (“код языка-КОД СТРАНЫ”).

Код языка обязательно должен быть в формате [ISO 639-1](#), для указания страны – используйте формат [ISO 3166-1 Alpha 2](#)

URL вариантов страниц должны быть указаны полностью, включая названия протокола (http или https): <https://example.com/foo>, а не [//example.com/foo](http://example.com/foo) или [/foo](http://example.com/foo).

Пример:

```
<link rel="alternate" hreflang="en" href="https://site.com"/>
<link rel="alternate" hreflang="de" href="https://site.com/de"/>
<link rel="alternate" hreflang="pl" href="https://site.com/pl"/>
```

Проверка реализации мультиязычности:

- при переключении языка пользователь попадает точно на такую же страницу на другом языке, а не, например, на главную;
- URL разные для каждой языковой версии страницы;
- корректно прописанные мета-теги и корректно переведен весь контент на соответствующий язык;
- атрибут hreflang соответствует описанным выше правилам.

Важно: Если в файле не указан дополнительный язык, то переключатель языка НЕ выводим.

4. Скорость сайта

Сайт должен быстро загружаться и находиться в зеленой зоне по [PageSpeed Insights](https://pagespeedinsights.com/).

5. Отображение на разных типах устройств

На сайте необходимо реализовать адаптивную верстку шаблона под разные типы устройств. После настройки, сайт должен проходить проверку по инструменту: <https://search.google.com/test/mobile-friendly?hl=ru>

6. Разметка микроданными

7.1. Микроразметка контактов

<https://schema.org/Organization>

```
<script type="application/ld+json"
{
  "@context": "http://schema.org",
  "@type": "Organization",
  "brand": "{Название_компании}",
  "logo": "{logo_URL}",
  "name": "{домен_сайта}",
  "url": "{page_URL}",
  "sameAs": [
    "https://www.facebook.com/{facebook_id}",
    "https://t.me/s/{twitter_id}",
    "https://www.linkedin.com/company/{linkedin_id}"
  ],
  "department": [{
    "@context": "http://schema.org",
    "@type": "Organization",
    "location": {
      "@type": "Place",
      "address": {
        "@type": "PostalAddress",
        "addressLocality": "{City_name}",
        "streetAddress": "{Street_name}",
        "telephone": "{telephone_number}",
        "name": "{company_name}",
        "image": "{main_image_URL}"
      }
    }
  }]
}
```

7.2. Микроразметка «Хлебных крошек»

<https://schema.org/BreadcrumbList>

```
{
  "@context": "https://schema.org",
  "@type": "BreadcrumbList",
  "itemListElement": [{
    "@type": "ListItem",
    "position": 1,
    "name": "{Main-Page-Name}",
    "item": "{URL}"
  }]
}
```

```

}, {
  "@type": "ListItem",
  "position": 2,
  "name": "{Second-Page-Name}",
  "item": "{URL}"
}, {
  "@type": "ListItem",
  "position": 3,
  "name": "{Third-Page-Name}",
  "item": "{URL}"
}, {
  "@type": "ListItem",
  "position": 4,
  "name": "{Last-Page-Name}"
}
}

```

7.3. Микроразметка страницы FAQ (вопросы-ответы)

<https://schema.org/FAQPage>

```

<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "FAQPage",
  "mainEntity": [
    {
      "@type": "Question",
      "name": "Question 1",
      "acceptedAnswer": {
        "@type": "Answer",
        "text": "Answer 1"
      }
    },
    {
      "@type": "Question",
      "name": "Question 2",
      "acceptedAnswer": {
        "@type": "Answer",
        "text": "Answer 2"
      }
    }
  ]
}
</script>

```

7.4. Микроразметка статьи

<https://schema.org/Article>

```

<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Article",
  "headline": "Title of an Article",
  "image": [
    "https://example.com/photos/1x1/photo.jpg",

```

```
"https://example.com/photos/4x3/photo.jpg",
"https://example.com/photos/16x9/photo.jpg"
],
"datePublished": "2015-02-05T08:00:00+08:00",
"dateModified": "2015-02-05T09:20:00+08:00",
"author": [{
  "@type": "Person",
  "name": "Jane Doe",
  "url": "http://example.com/profile/janedoe123"
}]
}
</script>
```

7.5. Микроразметка авторов в блоге

<https://schema.org/Person>

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Person",
  "name": "Author name",
  "sameAs": ["https://www.linkedin.com/in/Author name/", "https://twitter/jAuthor name/",
"https://facebook/Author name/"],
  "url": "https://site.com/blog/author/Author name/",
  "mainEntityOfPage": {
    "@type": "WebPage",
    "@id": "https://site.com/blog"
  }
}
</script>
```

Система должна быть масштабируемой. Т.е. в дальнейшем мы можем добавлять новые поля, блоки.

Критерии приема:

- 1) на вход даем эксель заданного формата и получаем результат в виде полностью рабочего сайта, без битых ссылок статического с адаптивной версткой
- 2) Если в каких то ячейках много или не хватает данных на статическом сайте мы либо траншей им данные либо дополняем верстка не должна страдать
- 3)

На выходе получаем полностью оптимизированный и корректно работающий сайт, который отображается на всех типах устройств, быстро подгружается (и физически и зеленая зона при проверках) и соответствует всем требованиям выше.