1. **Планирование успеха и измерение его эффективности.**

Сеть интернет – это отдельный мир, в котором большая часть правил обычного мира не работают, но тем не менее «онлайн» мир тесно переплетен с миром «оффлайн». В полной мере эту закономерность можно увидеть в сфере монетизации интернет сайтов. Заработок на интернет блоге в современном мире является перспективным направлением.

Основными способами заработка на блоге являются:

1. Партнерские программы – продажа не своих товаров и услуг по реферальной программе. В этом варианте особо интересна реклама чужих товаров или услуг. Реализация может заключаться в следующем: статья в блоге, рассказывающая о путешествиях может включать скрытую рекламу туроператора или гостиниц, упомянув которые в контексте статьи будет являться скрытой рекламой.
2. Торговля товарами и услугами – продажа своих собственных товаров (например, изделия ручной работы). Статья в блоге может содержать прямые указания на продаваемый товар, включая фотографии и видео товара, а также напрямую говорить о талантах автора статьи или иметь описание процесса производства, тем самым привлекая читателей мотивируя их на покупку.
3. Реклама в блоге – заработок, основанный на показе пассивной рекламе, представленной в виде баннеров, рекламных ссылок или контекстной рекламы. Обычно, используются баннерные сети, предоставляющие услуги по заработку на показе рекламы, например, Google AdSense.
4. Брендирование – помещение на фон (подложку) блога большой картинки или анимации, занимающей большое место. Один из самых доходных способов монетизации.
5. Размещение рекламных статей – размещение в своем блоге специальных рекламных статей, наряду с авторскими статьями. Как правило. Это уже написанные статьи, один-два абзаца рассказывающие о товаре или услуге.
6. Продажа ссылок – размещение ссылок в своих статьях. Ведущих на другие сайты. Суть этой системы заключается в том, что, чем больше источников ссылается на сайт, тем выше его рейтинг в поисковых запросах.

Это самые перспективные, на мой взгляд, способы монетизации блога, существуют и другие, но их эффективность не высокая.

1. **Изучение конкуренции на примере блога** [**https://www.bravebird.de/**](https://www.bravebird.de/)
2. **Разработка программы-поискового робота для сбора информации о блоге конкурента.**

Рассмотрев возможные варианты платформ для создания поискового-робота, я принял решение о реализации программы в виде двух модулей: сборщик данных и визуализатор собранных данных. Рассмотрим оба модуля отдельно.

* Модуль сбора данных. На мой взгляд, данная задача является реализацией задачи сбора и обработки большого количества данных, Современное название такого подхода – BigData. В настоящее время наиболее подходящим языком программирования для работы с BigData является Python. Конечно, подобное может быть реализовано и на других ЯП, но проведя сравнение наиболее подходящих ЯП для моего проекта, я выбрал именно Python. Результаты моего сравнения ЯП представлены в таблице:

|  |  |  |
| --- | --- | --- |
| **Язык программирования** | **Достоинства** | **Недостатки** |
| **R** | Большая гибкость и свободный код. Наличие большого количества библиотек на CRAN | Язык. Более подходящий для сложных статистических вычислений. Сложность обучения. Интерфейс командной строки. |
| **Python** | Популярный язык в науке. Низкий порог вхождения. Огромный набор готовых фреймворков В отличие от R, Python является традиционным объектно-ориентированным языком программирования, поэтому большинству разработчиков будет комфортно с ним работать. | Основной недостаток – медленное выполнение программ на языке Python, обусловленное тем, что язык является интерпретируемым. |
| **Java** | Предоставляет доступ к огромной экосистеме библиотек профилировщиков, отладчиков, все из которых используются давно и много. | Основными претензиями к Java являются многословность и отсутствие REPL. |

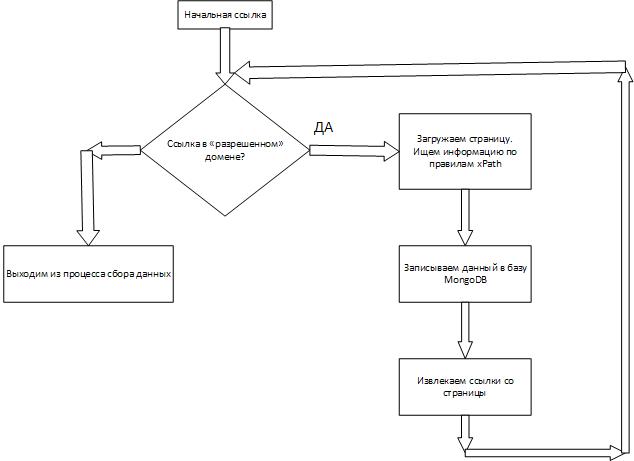
Таким образом, я остановил свой выбор на Python и фреймворке Srapy. В качестве хранилища собранных данных будет использована MongoDB. MongoDB выбрана потому, что позволяет хранить данные в формате JSON в виде коллекций и просто интегрируется с любым другим фрейворком.

* Для визуализации данных выбран NodeJS – как самый популярный фреймворк для Front-end разработки.

Реализация сборщика данных.

Общий принцип работы сборщика (в терминах Scrapy – «паук»), заключается в следующем: «паук» загружает страницу и ищет на ней данные, указанные с помощью правил xPath. После сбора всех данных на странице, выделяет ссылки на странице и пробует загрузить страницы по ссылкам и собрать данные с них. Для того, чтобы не выйти за пределы интересующего нас сайта, применяется ограничение для «паука». В моем случае, ограничение выглядит как: *allowed\_domains = ["bravebird.de"]* означающее, что «пауку» нельзя переходить по ссылкам, если имя домена в них не начинается с *bravebird.de,* т.е. мы ограничиваем переходы «паука» только доменом *bravebird.de.*

Блок-схема алгоритма работы «паука»:



Для загрузки страницы мы должны указать «пауку» его имя и начальные настройки:

class BravebirdSpider(CrawlSpider):

#имя паука в проекте

name = 'bravebird'

#разрешенный домен, для механизма обхода страниц сайта, чтобы не выйти за пределы сайта

allowed\_domains = ["bravebird.de"]

# Стартовый адрес сайта

start\_urls = ['https://www.bravebird.de']llowed\_domains = ["bravebird.de"]

«Паук» скачает страницу и будет искать данные на странице, согласно правил. Для «паука» правила можно задавать различными способами, самым эффективным является xPath. Этот способ предполагает указание селекторов в DOM дереве странице, по которому нам нужно извлечь данные.

Для интересующих нас данных, правила будут выглядеть следующим образом:

**Название статьи:**

//div[@class="entry-header entry-header-top"]/h1//text()

**Дата публикации:**

//div[@class="entry-header entry-header-top"]/div[@class="entry-meta entry-meta-single"]/span[@class="post-info post-info-date"]/time[@class="entry-date published"]/text()

**Дата обновления статьи:**

//div[@class="entry-header entry-header-top"]/div[@class="entry-meta entry-meta-single"]/span[@class="post-info post-info-date"]/time[@class="updated"]/text()

**Количество комментариев к статье:**

//div[@class="entry-header entry-header-top"]/div[@class="entry-meta entry-meta-single"]/span[@class="post-info post-info-comment"]/a/text()

**Место, относящееся к статье:**

//div[@class="entry-header entry-header-top"]/div[@class="entry-meta entry-meta-single"]/span[@class="post-info post-info-location"]//text()'