

**МИНИСТЕРСТВО ЭНЕРГЕТИКИ, ПРОМЫШЛЕННОСТИ И СВЯЗИ
СТАВРОПОЛЬСКОГО КРАЯ**

**Государственное бюджетное профессиональное образовательное
учреждение «Ставропольский колледж связи имени Героя Советского
Союза В.А. Петрова»**

Специальность: 09.02.07 Информационные системы и программирование

Допустить к защите

приказом № __ от «__» _____2023г.

Заместитель директора по учебно-
воспитательной работе

_____ Е.В. Анищенко

МП

ДИПЛОМНАЯ РАБОТА

на тему:

Разработка метапоисковой системы с использованием языка
программирования PHP для ГАОУ ДО «Центр для одаренных детей «Поиск»
г. Минеральные Воды

Дипломник: Евсееенко Илья Николаевич

Курс 4 группа ИП193

Руководитель ВКР _____/И.В. Ерёмина

Ставрополь, 2023

**МИНИСТЕРСТВО ЭНЕРГЕТИКИ, ПРОМЫШЛЕННОСТИ И СВЯЗИ
СТАВРОПОЛЬСКОГО КРАЯ**

**Государственное бюджетное профессиональное образовательное
учреждение «Ставропольский колледж связи имени Героя Советского
Союза В.А. Петрова»**

УТВЕРЖДАЮ

Председатель ЦК ВТ

_____/И.В. Ерёмина

«27» марта 2023 г.

ЗАДАНИЕ

на выпускную квалификационную работу

Студенту 4 курса, ИП193 группы, специальности 09.02.07

Информационные системы и программирование

Евсеевко Илье Николаевичу

Тема работы: Разработка метапоисковой системы с использованием языка
программирования PHP для ГАОУ ДО «Центр для одаренных детей «Поиск»
г. Минеральные Воды

Утверждена приказом по ГБПОУ СКС №193-уд от «16» декабря 2022 г.

Исходные данные к ВКР: основные данные для создания веб-сайта

Перечень подлежащих разработке в ВКР вопросов или краткое содержание
ВКР:

- 1) Введение;
- 2) Предпроектное исследование предметной области;
- 3) Выбор средств разработки;
- 4) Разработка сайта метапоисковой системы;
- 5) Размещение сайта в сети Интернет;
- 5) Заключение.

Перечень приложений: техническое задание, исходный код главной
страницы сайта.

Рассмотрено на заседании ЦК «Вычислительная техника»

Протокол № 7 от «17» марта 2023 г.

Срок сдачи законченной работы «14» июня 2023 г.

Руководитель _____

(подпись руководителя)

/

(инициалы, фамилия)

Задание принял к исполнению: _____ «27» марта 2023 г.

(подпись обучающегося)

КАЛЕНДАРНЫЙ ПЛАН

Работы над дипломной работой студента Евсеенко Ильи Николаевича

№ п/п	Наименование этапов выполнения выпускной квалификационной работы	Сроки	Примечание
1.	Утверждение тематики ВКР	8.12.2022	
2.	Закрепление тем ВКР, назначение руководителя	16.12.2022	
3.	Подбор и анализ исходной информации, подготовка и утверждение плана ВКР	16.01.2023- 3.02.2023	
4.	Выполнение теоретической части ВКР	6.02.2023- 24.03.2023	
5.	Выдача задания на ВКР и преддипломную практику	27.03.2023	
6.	Сбор информации и систематизация материала во время прохождения преддипломной практики	20.04.2023- 10.05.2023	
7.	Оформление практической части ВКР	11.05.2023- 17.05.2023	
8.	Оформление приложений	18.05.2023- 19.05.2023	
9.	Согласование содержания ВКР, устранение замечаний руководителя	22.05.2023- 24.05.2023	
10.	Выполнение презентационной части	25.05.2023- 26.05.2023	
11.	Проверка содержания и утверждение ВКР руководителем	31.05.2023	
12.	Получение отзыва руководителя ВКР	1.06.2023- 2.06.2023	
13.	Рецензирование ВКР	5.06.2023- 6.06.2023	
14.	Предварительная защита	9.06.2023	
15.	Предоставление ВКР в учебную часть, на получение допуска к защите	14.06.2023	
16.	Демонстрационный экзамен	29.05.2023- 30.05.2023	
17.	Защита ВКР	19.06.2023	

Студент _____ / _____
подпись инициалы, фамилия

Руководитель работы _____ / _____
подпись инициалы, фамилия

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ	8
1.1 Анализ деятельности предприятия.....	8
1.2 Основные требования к разработке веб-сайта	10
2 ВЫБОР СРЕДСТВ РАЗРАБОТКИ	11
2.1 Выбор программного средства для разработки сайта	11
2.2 Понятие, виды и структура сайтов	11
2.3 Разработка интерфейса сайта.....	19
3 ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ.....	21
3.1 Документ-концепция проекта.....	21
3.2 Проектирование функциональной модели	22
3.2.1 Контекстная диаграмма	22
3.2.2 Диаграмма декомпозиции первого уровня.....	23
3.2.3 Диаграмма декомпозиции второго уровня DFD	24
3.3 Объектно-ориентированное проектирование системы	26
3.3.1 Диаграмма вариантов использования	26
3.3.2 Диаграмма кооперации.....	27
3.3.3 Диаграмма деятельности	28
3.3.4 Диаграмма состояний	29
3.3.5 Диаграмма классов.....	30
4 РАЗРАБОТКА МЕТАПОИСКОВОЙ СИСТЕМЫ	31
4.1 Разработка сайта.....	31
4.2 Проектирование и написание модуля поискового движка.....	33
4.2.1 Алгоритм модуля поискового движка	34
4.2.2 Объединение поисковых запросов	35
4.3 Добавление функции настроек	38
4.4 Функция поиска через протокол API	42
4.5 Отображение поискового ответа в виде HTML страницы	43

4.6 Схема сайта	44
4.7 Требования к ПК для просмотра сайта	46
4.8 Размещение сайта в сети Интернет	47
4.9 Тестирование сайта	47
ЗАКЛЮЧЕНИЕ	51
ПРИЛОЖЕНИЕ А	56
ПРИЛОЖЕНИЕ Б.....	60

ВВЕДЕНИЕ

В настоящее время в сети Интернет находится много различной информации. Эта информация находится на сайтах, которые индексируют поисковые роботы поисковых систем.

На данный момент есть несколько крупных поисковых систем – Google, Yandex, Bing, Yahoo. Данные системы работают по-разному, у каждого свой алгоритм поисковых роботов.

Основная проблема крупных поисковых систем – ответы на запросы разнятся от одной системы к другой. Некоторые сайты не индексируются конкретными поисковыми системами, а какие-то удаляются. В большинство поисковых систем внедрены метрические системы, отслеживающие действия пользователей данных систем. Таким образом страдает конфиденциальность пользователя, появляются таргетированные рекламные предложения, «рекламные» ответы на запросы.

Метапоисковые системы – помогают сохранить конфиденциальность пользователя, объединяют поисковые ответы, чтобы ответ на запрос был полным и независимым.

Целью выпускной квалификационной работы является создание метапоисковой системы для ЦДО «Поиск». Данный веб-сайт позволит пользователям системы искать информацию сразу из нескольких поисковых систем без отправки конфиденциальных данных, сохраняя полноту ответа на запрос.

Цель исследования – на основе теоретического материала, приобретенного из различных источников, разработать метапоисковую систему для ЦДО «Поиск». с использованием языка программирования PHP.

Объект исследования – ЦДО «Поиск».

Предмет исследования – процесс создания метапоисковой системы для ЦДО «Поиск».

Исходя из цели и предмета исследования, были определены следующие задачи исследования:

1. Изучить специальную литературу, Интернет-источники по разработке сайтов на языке программирования PHP;
2. Выделить требования, предъявляемые к созданию сайтов;
3. Выделить виды и структуру веб-сайтов;
4. Изучить паттерны проектирования и создания сайтов на PHP;
5. Спроектировать структуру метапоисковой системы;
6. Создать модули для поиска в различных поисковых системах;
7. Опубликовать веб-сайт в сети Интернет.

Для решения поставленных задач были выбраны следующие методы исследования.

Теоретические: изучение, анализ и систематизация специальной литературы по проблеме исследования; анализ существующих поисковых систем.

Практические: интервью с заказчиком, разработка структуры поисковой системы, публикация сайта в сети Интернет.

Практическая значимость работы состоит в том, что разработан и опубликован информационный веб-сайт метапоисковой системы в сети Интернет по следующему адресу <https://trydex.org.ru>.

1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Анализ деятельности предприятия

Основная деятельность Минераловодский филиал Государственного автономного образовательного учреждения дополнительного образования «Центр для одаренных детей «Поиск» – предоставление дополнительного образования детям, а также реализация дополнительных образовательных программ и услуг в интересах личности, общества, государства.

Филиал был основан в 2002 году и на сегодняшний день реализует 12 программ по четырём направлениям. Ежегодно в филиале обучается до 1500 учащихся. Важным аспектом деятельности филиала является его специализация на обучении одаренных детей. Филиал «Минераловодский филиал Государственного автономного образовательного учреждения дополнительного образования «Центр для одаренных детей «Поиск» предлагает широкий спектр направлений работы, охватывающих учебную, воспитательную, психологическую, конкурсную и научно-методическую деятельность.

В учебной деятельности филиал предлагает общеразвивающие программы, специализированные программы, а также профессионально-ориентированные программы, которые помогают учащимся раскрыть и развить свои способности и приобрести необходимые знания и умения. Кроме того, филиал предоставляет возможность получения международных сертификатов, подтверждающих соответствие качества образования международным стандартам по иностранным языкам и информатике.

В воспитательной деятельности филиал предлагает программу «Человек успеха», организацию интеллектуальных и праздничных мероприятий, а также работу с родителями.

В психологической деятельности филиал предлагает диагностику индивидуальных способностей, групповое и индивидуальное

консультирование, коррекционную работу, психологические тренинги личностного развития, а также школу для родителей и клуб для подростков.

В конкурсной деятельности филиал предлагает программу участия в интеллектуальных соревнованиях всех рангов, формирование жизненного стиля, направленного на успех, а также продвижение учащихся через систему интеллектуальных мероприятий.

В научно-методической деятельности филиал предлагает разработку и внедрение в учебный процесс авторских учебных программ, разработку и издание учебно-методических пособий, совершенствование методик преподавания и воспитания, а также организацию и проведение экспериментальной работы, семинаров, тренингов, консультаций для педагогических работников и курсов повышения квалификации сотрудников Центра.

Руководитель и заместители филиала – Костыря Марина Ивановна.

Наименование: Минераловодский филиал Государственного автономного образовательного учреждения дополнительного образования «Центр для одаренных детей «Поиск»

ИНН 2635009017

КПП 263501001

ОГРН 1022601954540

ОКВЭД 85.41

ОКТМО 07701000001

Банковские реквизиты:

Расчетный счет: 40702810160150000337 в Ставропольском отделении №5230 ПАО СБЕРБАНК РОССИИ.

Кор.счет: 30101810907020000615

БИК 040702615

1.2 Основные требования к разработке веб-сайта

Разрабатываемый веб-сайт должен предоставлять пользователю следующие возможности:

- быстрый и точный поиск информации в сети Интернет;
- защита конфиденциальности данных;
- предоставление результатов поиска в удобном и понятном формате;
- доступность сайта для использования на различных устройствах и браузерах.

Веб-сайт должен иметь дружелюбный интерфейс, рассчитанный на пользователя начальной и средней квалификации [9].

2 ВЫБОР СРЕДСТВ РАЗРАБОТКИ

2.1 Выбор программного средства для разработки сайта

Чтобы реализовать поставленные задачи, необходимо сначала выбрать программное средство для разработки информационного сайта. Во время интервью с руководителем минераловодского филиала государственного автономного образовательного учреждения дополнительного образования «Центр для одаренных детей «Поиск» Костыря Мариной Ивановиной было решено, что самым оптимальным вариантом будет использовать язык программирования PHP.

2.2 Понятие, виды и структура сайтов

Веб-сайт представляет собой комплекс программных, информационных и мультимедийных средств, логически связанных между собой [2]. Адрес каждой страницы веб-сайта состоит из трех компонентов:

- первый компонент – приставка «<http://>», которая присутствует в адресе практически любого сайта, а также любой его страницы, и выполняет техническую функцию. Приставка «<https://>» использует SSL протокол для защиты данных;

- вторая часть – адрес сайта, которому принадлежит страница. Например, это «ru.wikipedia.org»;

- третья часть – адрес самой страницы, принадлежащий сайту. Например, это «wiki/Сайт». Если эта часть отсутствует – значит это главная страница сайта.

В сети Интернет существует множество терминов, связанных с веб-сайтами, которые начинаются с приставки «web». Например, отрасль дизайна, которая занимается созданием красивых и удобных сайтов, называется «web-дизайном», а отдельная страница сайта – «web-страницей».

В зависимости от конкретной задачи, которую должен решать сайт, он может выполнять множество функций: от простого предоставления информации до возможности совершения покупок или других сделок через Интернет (например, Интернет-магазины), а также обеспечения возможности общения пользователей посредством обмена сообщениями, комментирования разнообразной информации (например, социальные сети). Необходимая функциональность определяется для каждого сайта индивидуально.

С учетом последних тенденций стремительного развития Интернет-рынка, можно утверждать, что веб-сайт является неотъемлемой частью современного предприятия и предоставляет широкий спектр возможностей своему владельцу.

В общем, не существует стандартной четко обусловленной классификации веб-сайтов. Разделение сайтов на типы начали использовать сами разработчики и опытные пользователи. Однако, в настоящее время, такое разделение на типы теряет свою актуальность, поскольку очень сложно определить границы любого конкретного сайта: ресурс может одновременно быть информационным, Интернет-магазином и малой социальной сетью, или сочетать в себе несколько других типов.

Когда речь заходит о разновидностях сайтов, можно выделить две основные категории: одностраничные и многостраничные. Одностраничные сайты, также известные как лендинги, состоят из одной страницы и обычно посвящены одному конкретному товару или услуге. Такие сайты позволяют более точно и красочно представить преимущества товара или услуги, и мотивировать клиента к действию, например, оставить контактные данные или совершить покупку. Одностраничные сайты также могут использоваться для продвижения основного сайта компании.

Многостраничные сайты, в свою очередь, предназначены для предоставления комплексной информации. Они могут содержать несколько десятков и даже сотен страниц. Цель таких сайтов – предоставить посетителям максимально полную и точную информацию о компании, ее продуктах и

услугах, новостях и событиях. Многостраничные сайты могут быть использованы для различных целей, включая информационные сайты, интернет-магазины, корпоративные сайты, социальные сети, блоги и форумы. Каждый из этих типов сайтов имеет свои особенности и функциональные возможности, которые могут быть оптимально использованы в зависимости от конкретных потребностей и целей. На рисунке 1 изображена схема виды сайтов.



Рисунок 1 – Виды сайтов

1) Корпоративные сайты.

Корпоративные сайты являются одним из самых распространенных видов сайтов. Они предназначены для предоставления наиболее полной информации о компании, ее продуктах и услугах, истории, преимуществах, ценах, контактах и т.д. Кроме того, на таких сайтах могут быть размещены примеры выполненных работ, отзывы клиентов, ближайшие мероприятия. На большинстве корпоративных сайтов есть возможность приобрести товары или услуги этой компании [3].

Одной из основных задач корпоративного сайта является коммуникация с потенциальными покупателями. Поэтому на таком сайте, как правило, можно задать свой вопрос, сделать заказ или оставить предложение по сотрудничеству. Основная задача корпоративного сайта – информировать

клиентов о сфере деятельности компании, формирование необходимого имиджа и помогать потенциальным клиентам сделать выбор в пользу компании.

Кроме корпоративных сайтов, существуют и другие разновидности. Например, сайты-визитки предназначены для представления сжатой информации о компании, ее краткой презентации, а более полную информацию клиент может узнать по телефону или в офисе. Другими разновидностями корпоративного сайта являются сайты-галереи и сайты-каталоги, где представлены либо результаты работы компании, либо каталог ее продукции. Каждый из этих типов сайтов имеет свои особенности и функциональные возможности, которые могут быть оптимально использованы в зависимости от конкретных потребностей и целей.

2) Информационные сайты.

Как следует из термина, информационные веб-ресурсы предназначены для предоставления посетителям полной и исчерпывающей информации по определенному вопросу. Модель заработка таких веб-сайтов основана на рекламе или предоставлении платного контента. Информационные веб-ресурсы могут включать следующие типы:

Новостные порталы – основная задача таких веб-ресурсов заключается в информировании пользователей о последних новостях и тенденциях в различных сферах жизни.

Тематические сайты – они могут быть посвящены какой-либо узкой тематике, так и охватывать широкий спектр вопросов. Тематические сайты пользуются популярностью, поскольку помогают пользователям быстро получить необходимую информацию по интересующему их вопросу. Однако создателям таких сайтов необходимо уделять внимание качеству контента и удобству использования сайта. Если сайт будет перегружен рекламой с минимальным количеством полезной информации, посетители вряд ли будут желать возвращаться на него повторно.

Тематические порталы – это большие интернет-ресурсы, которые предоставляют посетителям полную информацию по определенной тематике. Порталы имеют сходство с тематическими сайтами, но дополнительно предлагают средства взаимодействия для пользователей, такие как форумы и чаты, позволяющие пользователям общаться в пределах портала.

Блоги – данный тип веб-ресурса напоминает тематические порталы. Автор блога также стремится предоставить информацию своей аудитории по определенной тематике, за исключением одного: обычно автор описывает свою личную точку зрения.

Блоги могут быть корпоративными или персональными. Корпоративные блоги, как правило, создаются компаниями в качестве дополнительного ресурса для продвижения бренда. Их задача заключается в демонстрации экспертности компании в определенной области и предоставлении лучших решений для клиента [15].

Примером корпоративного блога может служить банк Тинькофф, который помимо своего основного корпоративного сайта, ведет несколько дополнительных веб-ресурсов, посвященных решению актуальных вопросов для предпринимателей.

Персональный блог, в свою очередь, обычно рассказывает о жизни конкретного человека. Формат такого блога может быть разнообразным. Создатель персонального блога может делиться своими личными и деловыми переживаниями, выражать свое мнение по различным вопросам, описывать путешествия, делать обзоры на автомобили и гаджеты, а также предоставлять множество другой информации.

3) Интернет-магазины.

Интернет-торговля представляет собой электронную платформу, где пользователи имеют возможность приобрести различные товары и услуги. Интернет-магазины могут быть как онлайн-версиями уже существующих офлайн-магазинов, так и самостоятельными электронными платформами.

Несколько примеров известных интернет-магазинов включают ozon.ru, wildberries.ru, dns-shop.ru и т.д.

4) Веб-сервисы.

Веб-сервисы представляют собой онлайн-ресурсы, созданные для выполнения различных задач, включая обмен информацией, общение и другие функции. В эту категорию веб-сайтов входит огромное количество разновидностей, и мы постараемся представить максимально полную классификацию:

Поисковые сервисы – помогают пользователю найти определенную информацию в интернете (Google, Яндекс, Duckduckgo).

Почтовые сервисы – Mail.ru, mail.yandex.ru.

Доска объявлений – представляет собой ресурс, на котором есть возможность размещения публичного объявления о продаже или покупке товаров и услуг. Например, Avito, auto.ru.

Сайт-агрегатор – представляет собой ресурс, дающий возможность свести продавцов каких-либо услуг и покупателей этих услуг. Сферы могут быть абсолютно различными: от путешествий до оптовых закупок. Например, MTC Travel, Aviasales и др.

Видеохостинг – RuTube, VK Video

Социальные сети – Вконтакте, Одноклассники, ЯRUS и др.

Перечисленные выше сервисы являются повседневными для большинства пользователей. Однако существуют и другие типы веб-сайтов, которые также играют важную роль в современном мире. К ним относятся конструкторы сайтов, которые позволяют пользователям создавать свои собственные веб-сайты без необходимости знания программирования. Также существуют файлообменные сервисы, которые обеспечивают возможность обмена файлами между пользователями. Сервисы облачного хранения данных предоставляют возможность хранить и доступ к файлам через сеть Интернет. Фотохостинги предназначены для загрузки и обмена фотографиями. Блоговые

сервисы позволяют пользователям создавать и вести свои собственные блоги, делиться своими мыслями, опытом и знаниями.

Все эти разнообразные веб-сервисы играют важную роль в сфере информационных технологий и обеспечивают пользователей возможность обмена информацией, коммуникации и выполнения различных задач в онлайн-среде.

5) Сайты-представительства.

Государственные сайты часто являются сайтами-представительствами, которые предоставляют гражданам информацию о правительственных службах, законах и регулированииях, а также о проводимых мероприятиях и услугах, доступных гражданам.

Социальные проекты также могут иметь сайты-представительства, где предоставляется информация о целях и задачах проекта, его инициаторах и способах участия. Такие сайты часто предлагают возможность обратной связи и поддержки проекта со стороны пользователей.

Сайты-представительства не имеют коммерческой направленности и служат в первую очередь для обеспечения доступа к информации и коммуникации с публикой. Их целью является предоставление информации и установление взаимодействия с пользователями для целей информирования, образования, социального участия и других общественных целей [14].

Веб-сайты обычно имеют иерархическую структуру, состоящую из страниц, содержащих тексты, изображения, мультимедийные элементы и другие объекты. Структура сайта играет важную роль в обеспечении комфортного пользовательского опыта и облегчении поиска необходимой информации.

Основными элементами структуры веб-сайта являются:

Главная страница (Homepage): Является первой страницей, на которую попадает посетитель сайта. Главная страница обычно содержит общую информацию о сайте, навигационные элементы и ссылки на основные разделы.

Разделы (Sections): Сайт может быть разделен на несколько основных разделов, которые отражают различные тематические категории или функциональные области. Каждый раздел содержит свои подразделы или страницы.

Подразделы (Subsections): Внутри каждого раздела могут быть созданы подразделы, чтобы дополнительно организовать и структурировать содержимое. Это помогает упорядочить информацию и облегчить поиск.

Страницы (Pages): Основные единицы содержимого на сайте представлены страницами. Каждая страница может содержать тексты, изображения, видео, формы, ссылки и другие элементы. Страницы могут быть непосредственно связаны с навигационными меню или ссылками на сайте [4].

Навигационное меню: Навигационное меню предоставляет посетителям возможность перемещаться по различным разделам и страницам сайта. Четкая и интуитивно понятная навигация помогает пользователям быстро находить нужную информацию.

Внутренние ссылки: Внутренние ссылки используются для связи страниц внутри сайта. Они позволяют пользователям переходить между различными страницами, что облегчает навигацию и обеспечивает связность контента.

Поиск: Наличие функции поиска на сайте позволяет пользователям быстро находить информацию, по ключевым словам, или фразам. Поиск облегчает доступ к нужному контенту, особенно на больших сайтах с обширной структурой.

Четкая и продуманная структура сайта способствует удобству использования и улучшает пользовательский опыт, позволяя посетителям быстро находить нужную информацию и проводить желаемые действия на сайте. Когда структура сайта хорошо организована, посетители могут легко найти интересующие их разделы, переходить между страницами, а также быстро понять логику сайта [13].

Важно, чтобы каждая страница сайта имела понятное название и соответствующее ее содержимому. Кроме того, ссылки и кнопки должны быть интуитивно понятными, чтобы пользователи могли легко осуществлять навигацию и переходить на нужные им страницы.

Хорошо структурированный сайт также облегчает работу по его обновлению и поддержке. При добавлении нового контента или изменении разделов сайта, структура должна быть гибкой и легко расширяемой, чтобы обеспечить удобство администрирования.

В конечном итоге, цель структуры веб-сайта заключается в том, чтобы предоставить посетителям удобство и удовлетворить их потребности в доступе к информации или выполнении определенных действий. Четкая и продуманная структура способствует достижению этой цели, делая сайт функциональным, удобным и привлекательным для пользователей.

2.3 Разработка интерфейса сайта

При разработке интерфейса сайта метапоисковой сети Trydex, основной акцент следует сделать на создании страниц, которые будут максимально легкими, красивыми и функциональными.

Легкость страниц означает оптимизацию загрузки контента, чтобы пользователи могли быстро получить доступ к информации без задержек. Это достигается использованием сжатых изображений, минимизацией кода и оптимизацией серверной стороны. Страницы должны быть быстрыми и отзывчивыми, что обеспечит позитивный пользовательский опыт и удовлетворение их потребностей [12].

Красота интерфейса включает в себя гармоничное сочетание цветов, выбор различных тем оформления, эстетичное размещение элементов, привлекательный дизайн и удобочитаемые шрифты.

Функциональность страниц – это важный аспект, который обеспечивает удобство использования сайта и позволяет пользователям эффективно

выполнять свои задачи. Это включает в себя интуитивно понятную навигацию и легкость поиска.

Разработка интерфейса сайта осуществлялась через онлайн-сервис для разработки интерфейсов и прототипирования. Одним из примеров таких сайтов является Figma (Рисунок 2 и 3).

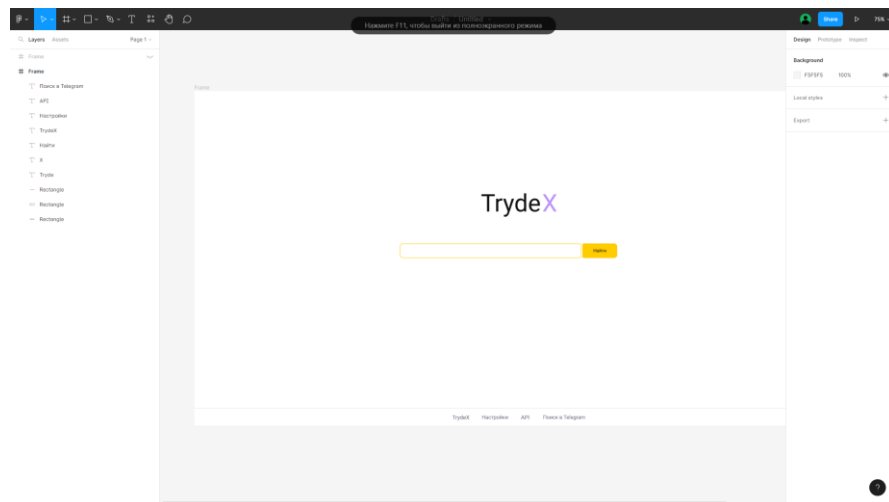


Рисунок 2 – UI главной страницы сайта

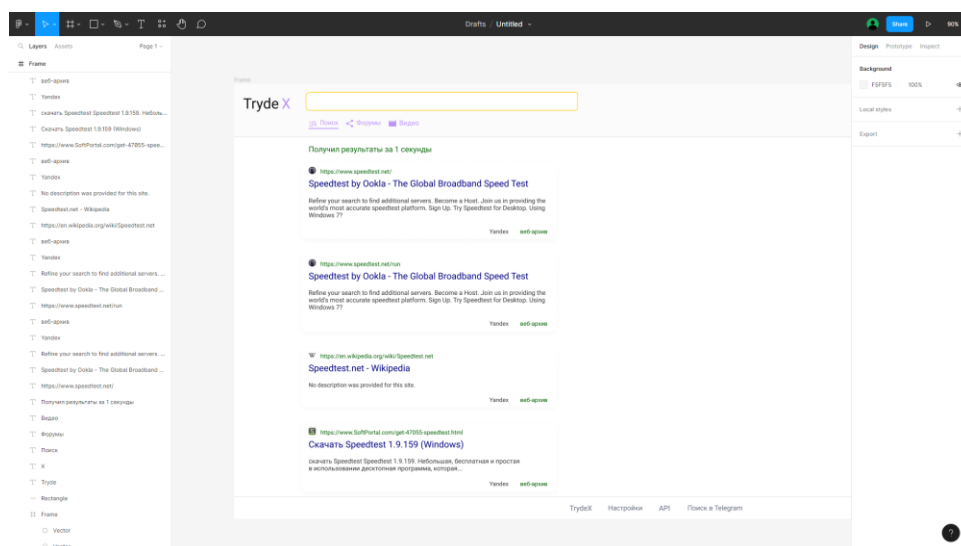


Рисунок 3 – UI страницы поисковой выдачи

3 ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ

3.1 Документ-концепция проекта

Концепция проекта, основанная на проведенных интервью с клиентом и документировании требований, включает следующие ключевые моменты:

1. Интуитивно понятный интерфейс: разработка интерфейса, который будет легко осваиваться пользователями, обеспечивая удобный ввод, просмотр и отслеживание данных, связанных с поисковым запросом.

2. Возможности персонализации: предоставление возможностей пользователям для постановки целей, напоминаний и получения персонализированных рекомендаций в соответствии с их потребностями и предпочтениями.

3. Наглядная визуализация данных и отчетность: создание визуальных инструментов и отчетов, позволяющих пользователям легко анализировать прогресс и результаты своих поисковых запросов.

4. Возможность подключения и интеграции: предусмотренные возможности интеграции сервиса с другими устройствами и приложениями, связанными с поисковым запросом, для более широкого функционала и удобства использования.

5. Меры конфиденциальности и безопасности: разработка надежных мер конфиденциальности и безопасности, чтобы защитить конфиденциальные данные пользователей от несанкционированного доступа и утечек.

6. Совместимость и бесперебойная работа: обеспечение совместимости сервиса с различными платформами и операционными системами, а также непрерывной работоспособности на целевой операционной системе.

7. Обработка ошибок и управление исключениями: реализация надлежащей обработки ошибок и управления исключениями, чтобы повысить надежность сервиса и предотвратить потерю данных или снижение производительности.

8. Документация и поддержка: разработка всесторонней документации и предоставление поддержки пользователям для эффективного использования сервиса и решения возникающих вопросов и проблем.

9. Масштабируемый и обслуживаемый дизайн: создание дизайна сервиса, который будет масштабируемым и легко поддерживаемым, чтобы облегчить будущие улучшения и модификации сервиса в соответствии с меняющимися потребностями пользователей.

3.2 Проектирование функциональной модели

3.2.1 Контекстная диаграмма

На рисунке 4 представлена схема, иллюстрирующая процесс создания контекстной диаграммы программной системы с применением нотации IDEF0.



Рисунок 4 – Контекстная диаграмма информационной системы

Контекстная диаграмма представляет собой графическое представление, которое позволяет визуализировать взаимодействие программной системы с ее окружением.

Нотация IDEF0, или функциональная 블록ная диаграмма IDEF0, является стандартизированным методом моделирования, используемым для анализа и проектирования систем. Она позволяет описать функциональные блоки и связи между ними в системе, а также определить входы и выходы каждого блока.

Контекстная диаграмма программной системы, созданная с использованием нотации IDEF0, помогает исследователям и разработчикам получить общее представление о системе и ее взаимодействии с внешней средой. Это основной этап проектирования, который затем может быть уточнен и дополнен более подробными диаграммами и моделями.

В контекстной диаграмме демонстрируется структура и взаимодействие программной системы с внешними компонентами. Она позволяет определить главные функции и роли системы, а также ее взаимосвязь с окружающей средой. Это важный инструмент для понимания системы и ее дальнейшего развития.

3.2.2 Диаграмма декомпозиции первого уровня

На рисунке 5 представлена первая декомпозиция диаграммы IDEF0, отображающая систему в виде основных функций.

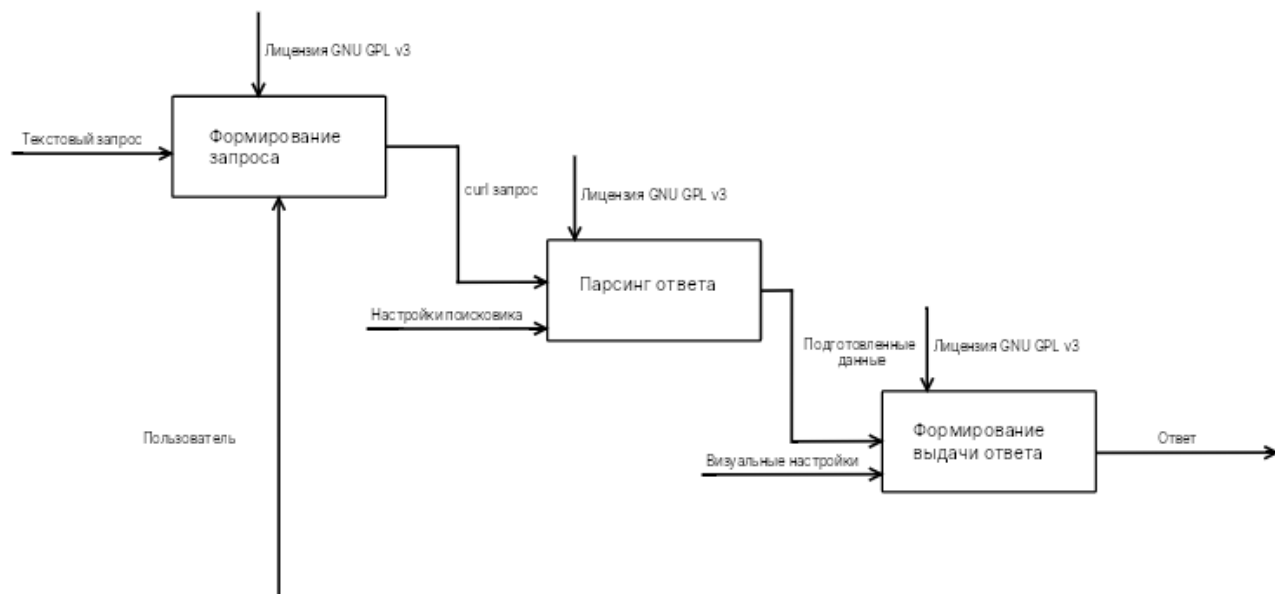


Рисунок 5 – IDEF0 – первое разбиение

Данная диаграмма служит важным инструментом для анализа и понимания работы системы, а также для определения ее ключевых функций и взаимосвязей между ними.

Декомпозиция диаграммы IDEF0 является этапом разработки, на котором осуществляется разбиение системы на более простые функциональные блоки. Это позволяет более детально изучить каждую функцию и выделить ключевые аспекты ее работы.

В первой декомпозиции, система разбита на основные функции, которые выполняются внутри нее. Это позволяет увидеть структуру и иерархию функций системы, а также их взаимосвязи. Данная диаграмма является отправной точкой для более детального анализа и разработки системы.

3.2.3 Диаграмма декомпозиции второго уровня DFD

На рисунке 6 представлена декомпозиция второго уровня системы, использующая нотацию DFD.

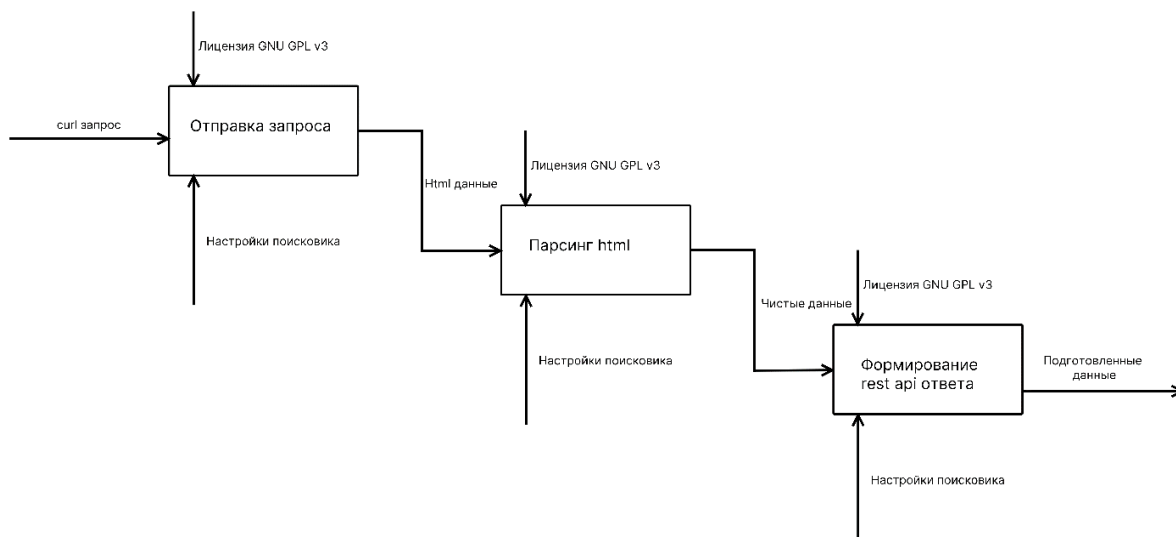


Рисунок 6 – DFD диаграмма информационной системы

Эта диаграмма позволяет более подробно изучить структуру и функционирование системы, разбивая ее на более мелкие компоненты и отображая потоки данных между ними.

Метод декомпозиции системы с использованием диаграмм потоков данных (DFD) является распространенным инструментом анализа и проектирования систем. Он позволяет увидеть, как данные проходят через систему и взаимодействуют между ее компонентами. Декомпозиция на второй уровень позволяет более детально рассмотреть каждый компонент системы и определить их функции и взаимосвязи.

Декомпозиция второго уровня является продолжением процесса декомпозиции системы. Она позволяет подробно изучить функциональность каждого компонента и понять, как они взаимодействуют друг с другом. Данная диаграмма предоставляет более глубокое представление о структуре системы и помогает разработчикам более эффективно анализировать и проектировать систему.

3.3 Объектно-ориентированное проектирование системы

3.3.1 Диаграмма вариантов использования

Диаграмма вариантов использования, изображенная на рисунке 7, представляет высокоуровневое представление системы и ее функциональности с точки зрения пользователей. Она служит важным инструментом для анализа и проектирования системы, а также для определения функций, которые ожидаются от системы со стороны пользователей.

Диаграмма вариантов использования помогает исследователям и разработчикам понять, как пользователи будут взаимодействовать с системой и какие функциональные возможности им будут доступны. Она отображает различные сценарии использования системы, которые могут включать в себя действия пользователей, ввод и вывод данных, а также взаимодействие с другими системами или компонентами.

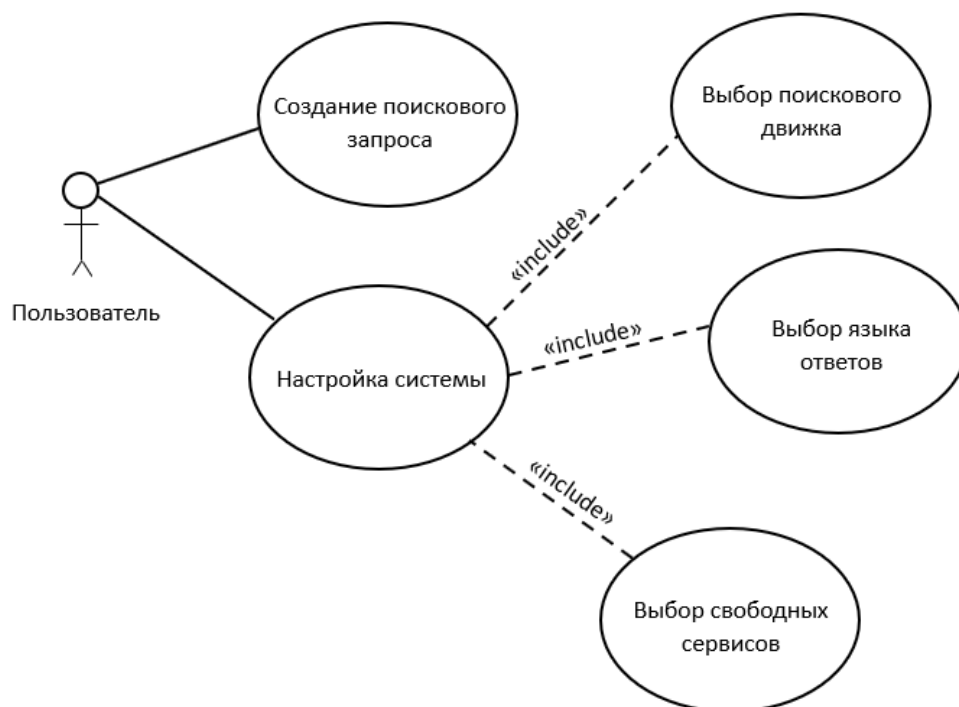


Рисунок 7 – Диаграмма вариантов использования для информационной системы

3.3.2 Диаграмма кооперации

На рисунке 8 представлена диаграмма кооперации, которая отображает взаимодействие объектов или компонентов системы через передачу сообщений или использование методов.

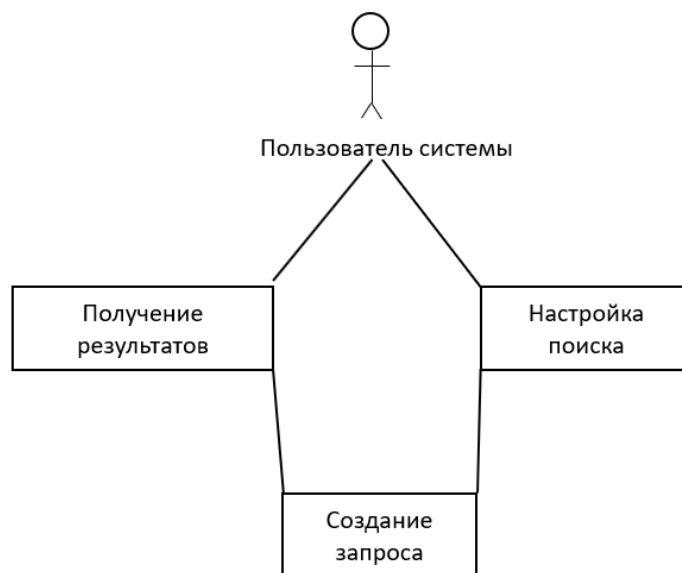


Рисунок 8 – Диаграмма кооперации

Данный графический инструмент позволяет проиллюстрировать, как различные элементы системы сотрудничают друг с другом для достижения определенного результата.

Анализ взаимодействия объектов или компонентов системы на диаграмме кооперации помогает разработчикам лучше понять, как каждый элемент взаимодействует с другими для выполнения определенных задач или функций системы. Важным аспектом диаграммы кооперации является отображение передаваемых сообщений или используемых методов, которые обеспечивают связь между элементами системы.

3.3.3 Диаграмма деятельности

Диаграмма деятельности, представленная на рисунке 9, является графическим инструментом, который демонстрирует поток работ или действий, осуществляемых в системе, процессе или варианте использования.

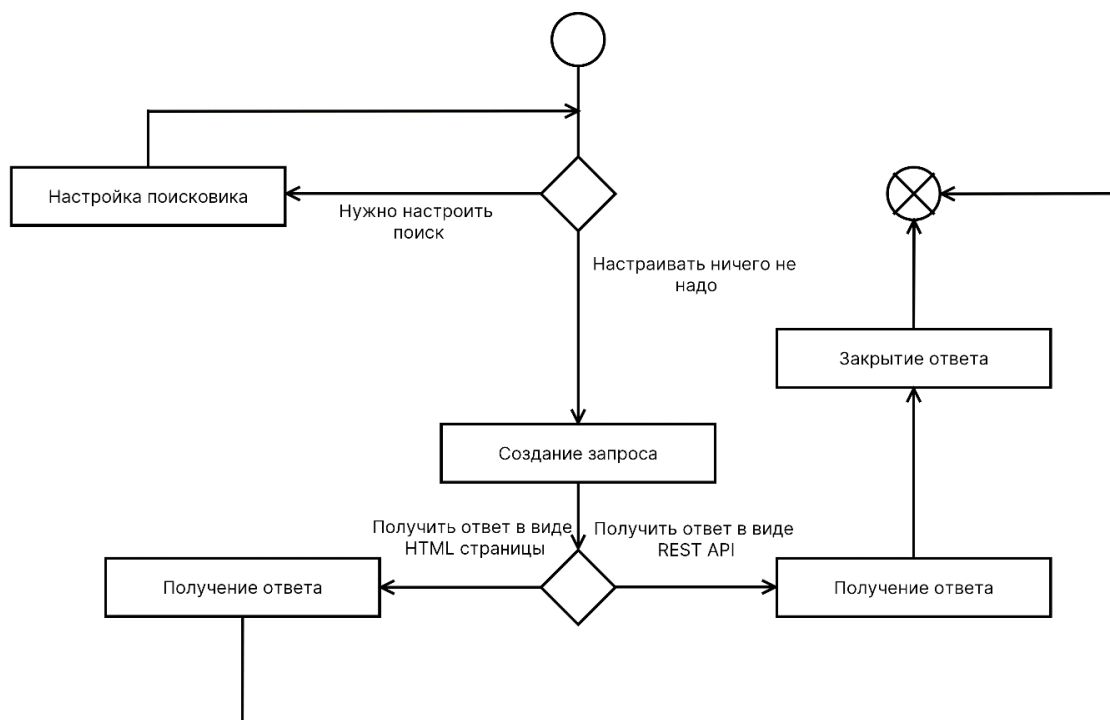


Рисунок 9 – Диаграмма деятельности

Этот тип диаграммы позволяет представить последовательность действий и связи между ними, образуя логическую структуру процесса.

Анализ деятельности на диаграмме позволяет визуализировать шаги, необходимые для достижения определенной цели или результатов. Диаграмма деятельности часто используется для описания бизнес-процессов, программного обеспечения или других комплексных систем, где важно понимать последовательность действий и взаимосвязи между ними.

3.3.4 Диаграмма состояний

Диаграмма состояний, представленная на рисунке 10, является графическим инструментом, который иллюстрирует различные состояния, переходы между ними и события, вызывающие эти переходы, для конкретной системы или объекта в определенных условиях.

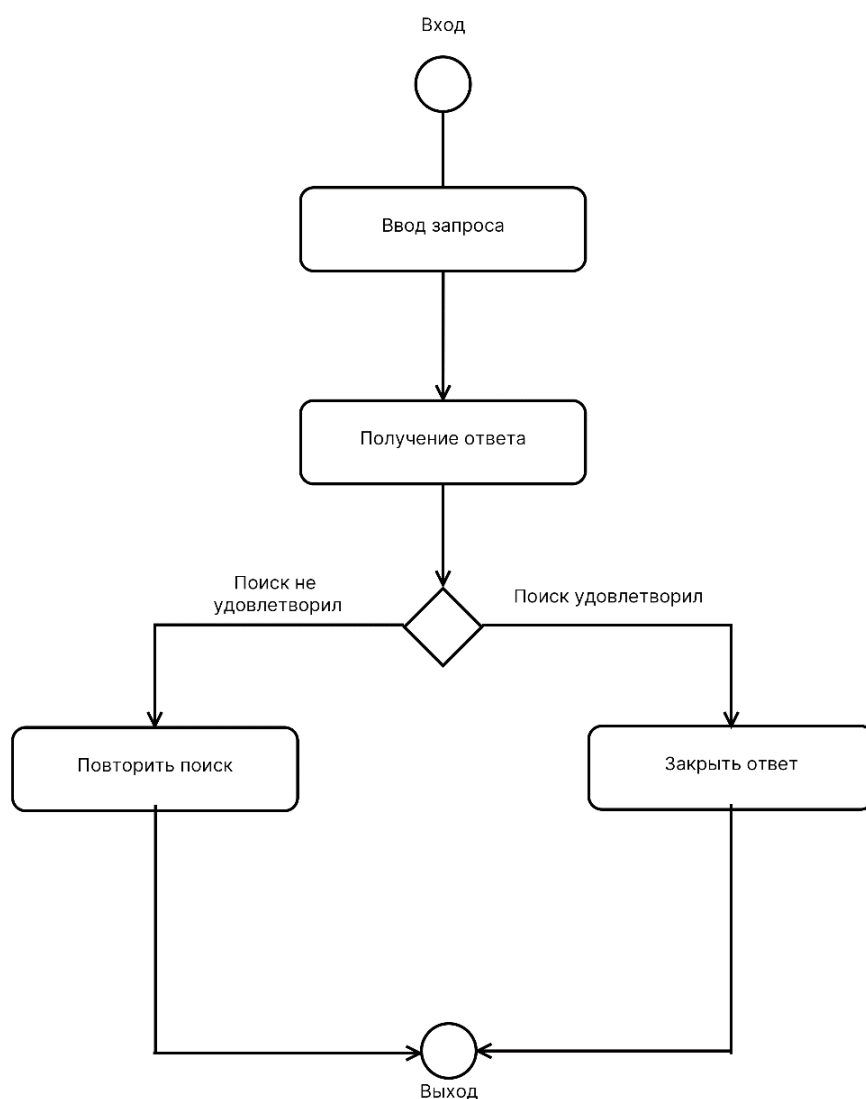


Рисунок 10 – Диаграмма состояний

Этот тип диаграммы позволяет моделировать и анализировать поведение системы и управление ее состояниями.

Анализ диаграммы состояний позволяет понять, как система или объект изменяет свое состояние в ответ на определенные события или условия.

Диаграмма состояний может включать различные состояния, такие как активное, ожидающее, завершенное и т.д., а также переходы между этими состояниями, которые могут быть вызваны определенными событиями.

3.3.5 Диаграмма классов

В данной системе, с целью сохранения конфиденциальности, не используется база данных. Однако, в системе присутствует класс REST API, который предоставляет интерфейс для взаимодействия с внешними системами или клиентами. Диаграмма классов на рисунке 11 наглядно демонстрирует структуру этого класса, его свойства и операции, которые могут быть выполнены с использованием REST API.

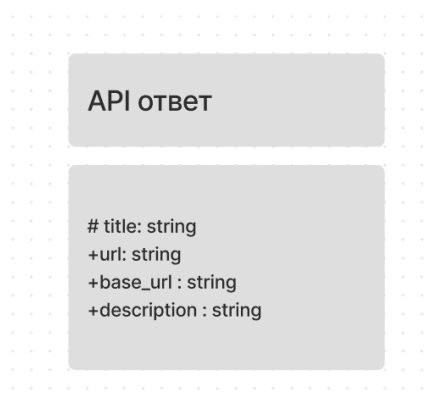


Рисунок 11 – Диаграмма классов

Анализ диаграммы классов позволяет лучше понять объекты в системе и их взаимодействие. Классы представляют собой шаблоны для создания экземпляров объектов, определяя их общие свойства и поведение. Операции, представленные на диаграмме классов, описывают функциональность, доступную для каждого класса.

4 РАЗРАБОТКА МЕТАПОИСКОВОЙ СИСТЕМЫ

4.1 Разработка сайта

Для разработки сайта метапоисковой системы был использован язык программирования PHP.

PHP – это язык программирования, предназначенный для генерации HTML страниц на веб-сервере и работы с базами данных. В настоящее время он поддерживается большинством хостинг-провайдеров. PHP входит в состав LAMP – стандартного набора для создания веб-сайтов (Linux, Apache, MySQL, PHP (Python или Perl)).

В области веб-программирования PHP является одним из самых популярных скриптовых языков (наряду с JSP, Perl и языками, используемыми в ASP.NET) благодаря своей простоте, скорости выполнения, богатым функциональным возможностям и распространению исходного кода на основе лицензии PHP [5].

PHP – модульный язык программирования. Из одного модуля можно подключить другой модуль. Тем самым можно разбить функциональность сайта на папки, содержащие конкретные модули. Структура сайта системы в виде «Дерева» каталогов показана на рисунке 12.

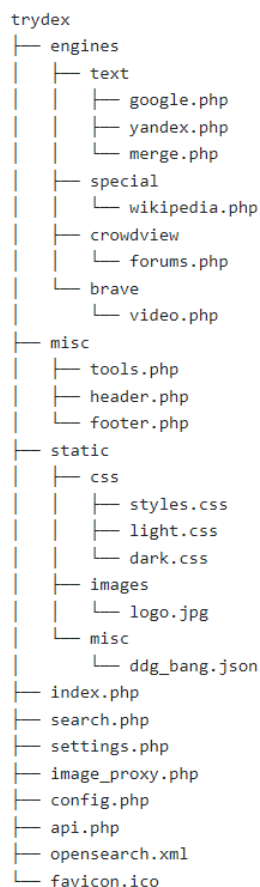


Рисунок 12 – Файловая структура сайта

В корневом каталоге содержатся запускаемые пользователем напрямую в браузере функциональные файлы: `index.php` – подключает модули `footer.php` и `header.php`, содержит форму для ввода запроса и кнопку поиска, направляющая на страницу `search.php`.

`image_proxy.php` – модуль для вывода иконок сайтов в результате выдаче. Модуль необходим, чтобы провайдеры поиска иконок не отслеживали IP адрес пользователя, если бы он напрямую получал иконки сайтов.

`settings.php` – страница настроек.

`config.php` – модуль для установки параметров настроек по умолчанию.

`api.php` – модуль для предоставления поисковой выдачи в виде API интерфейса. API выводится в виде JSON страницы.

`search.php` – страница, выводящая поисковые результаты в виде HTML файла.

В папке `statis` содержатся файлы стилей `CSS`, картинки и конфигурационный файл для поиска по различным сайтам.

В папке `misc` содержатся шапка и подвал сайта. Это нужно для того, чтобы не писать один и тот же код для файлов `index.php` и `search.php`, а подключать их модулем.

В папке `engines` находятся модули поисковых систем, таких как Яндекс, Google.

4.2 Проектирование и написание модуля поискового движка

У Яндекс и Google есть официальная документация для получения поисковых ответов в виде интерфейса прикладного программирования. У Яндекс это XML (<https://xml.yandex.ru/>), у Google – API (<https://developers.google.com/custom-search/v1/overview?hl=en>).

Основная проблема официального доступа к этим интерфейсам – наличие ограничения на количество запросов. Таким образом единственный доступный и неограниченный доступ к поисковым выдачам является парсинг страницы поискового ответа у каждой поисковой системы.

Легальность метода парсинга состоит в терпимом подходе к метапоисковым движкам, особенно если они не создают негативного воздействия на бизнес или функционирование поисковой системы, и, если использование результатов поиска является ограниченным и не конкурирует напрямую с официальным сервисом.

Метапоисковая система Trydex является Open Source продуктом и не преследует коммерческой выгоды. Результат поиска хоть и не является ограниченным, но ограничен методами использования, т.е. данный сайт в основном будет использоваться внутри филиала ЦДО Поиск как альтернатива существующим поисковым сервисам. Наличие лицензии GNU на код Trydex так же не позволяет использовать код в коммерческих целях, в отличие от

лицензии BSD, которая разрешает использование кода в продуктах с закрытым исходным кодом, который может использоваться с целью получения выгоды.

4.2.1 Алгоритм модуля поискового движка

В начале модуля иницируются переменные с доменом, языком поиска и URL запросом для получения результатов выдачи. Далее описывается функция, которая получает на вход текст поискового запроса и номер страницы. Номер страницы используется для получения большей информации, которая распределена по страницам у поисковой системы [6].

Далее, с помощью функции `get_xpath()` в переменную `xpath` вносится содержимое страницы поисковой системы в формате DOM, который, далее, через цикл парсит каждый блок с уникальным ответом и вносит в переменные `url`, `title`, `description` значения текста. Парсинг осуществляется через получение значений HTML тегов через XPath пути. У каждого текста на странице есть уникальный XPath путь. Указав путь, и прописав на PHP строчку «`$title->textContent`», можно получить текст контента. Далее, используя функцию `array_push`, переменная `results` наполняется массивом данных переменных `url`, `title` и `description` и возвращается методом `return`.

Данный модуль (Рисунок 13) универсален и подходит для любой поисковой системы, важно ли изменить XPath пути для получения необходимых текстовых данных.

```

1  <?php
2
3  $domain = $config->google_domain;
4  $language = isset($_COOKIE["google_language"]) ? htmlspecialchars($_COOKIE["google_language"]) : $config->google_language;
5  $google_url = "https://www.google.$domain/search?q=$_query&start=$page&hl=$language&lr=lang_$language&asearch=arc&async=use_ac:true,_fmt:prog";
6
7  if (isset($_COOKIE["safe_search"])) {
8      $google_url .= "&safe=medium";
9  }
10
11 function get_google_results($response, $page)
12 {
13     global $config;
14
15     $results = array();
16     $xpath = get_xpath($response);
17
18     foreach ($xpath->query("//div/div[contains(@class, 'g')]") as $result) {
19         $url = $xpath->evaluate("./div[@class='yuUbf']/a/@href", $result)[0];
20
21         if ($url == null)
22             continue;
23
24         if (!empty($results)) // filter duplicate results, ignore special result
25         {
26             if (!array_key_exists("special_response", end($results)))
27                 if (end($results)["url"] == $url->textContent)
28                     continue;
29         }
30
31         $url = mb_convert_encoding($url->textContent, "ISO-8859-1", "UTF-8");
32         $url = check_for_privacy_frontend($url);
33         $title = $xpath->evaluate("./h3", $result)[0];
34         $title = mb_convert_encoding($title->textContent, "ISO-8859-1", "UTF-8");
35         $description = $xpath->evaluate("./div[contains(@class, 'vwiC3b')]",$result)[0];
36         $description = mb_convert_encoding($description->textContent, "ISO-8859-1", "UTF-8");
37         array_push(
38             $results,
39             array(
40                 "title" => htmlspecialchars($title),
41                 "url" => htmlspecialchars($url),
42                 "base_url" => htmlspecialchars(get_base_url($url)),
43                 "description" => $description == null ?
44                     "No description was provided for this site." :
45                     htmlspecialchars($description),
46                 "source" => "Google"
47             )
48         );
49     }
50
51     return $results;
52 }
53 >>

```

Рисунок 13 – Код модуля поискового движка

4.2.2 Объединение поисковых запросов

Чтобы создать пункт меню необходимо авторизоваться в административной панели, затем нажать Меню – Менеджер меню – MainMenu. В загрузившемся окне необходимо нажать кнопку Создать и заполнить форму, которая представлена на рисунке 8. Таким образом, создаем пункты меню. Нажать кнопку «Сохранить» – пункт меню готов.

Важная функция метапоисковой системы – объединение ответов из разных поисковых систем. Основная цель метапоисковой системы состоит в том, чтобы предоставить пользователю единый доступ к результатам поиска, полученным из различных поисковых движков.

Метапоисковые системы работают путем отправки поисковых запросов к нескольким поисковым движкам одновременно и сбора ответов от каждого из них. Затем эти ответы объединяются и представляются пользователю в удобном формате, обычно в виде единого списка результатов поиска. Это позволяет пользователю получить более широкий обзор информации, собранной из различных источников.

В Trydex реализован поиск по тексту, форумам и видео. Для поиска по текстовым запросам на данный момент используются Яндекс и Google. Модуль объединения универсален и для других видах поиска. В начале мы импортируем поисковые движки, которые нужно объединить. (Рисунок 14).

```
require "engines/text/yandex.php";  
require "engines/text/google.php";
```

Рисунок 14 – Импортирование существующих поисковых движков

Далее, через цикл выполняются функции `get_yandex_results($response, $page)` и `get_google_results($response, $page)`. Эти функции возвращают массивы данных поисковых ответов. Далее, каждый массив данных проверяется на наличие текста, отвечающего за URL адрес одного поискового ответа. Если данные из нескольких поисковых систем одинаковы, то они объединяются в единый массив, а к полю массива `source`, приписывается название поисковой системы, которая объединила ответ с другой системой (Рисунок 15).

```
foreach ($temp_results as $result) {  
    $domain = $result["url"];  
  
    if (!isset($unique_results[$domain])) {  
        $unique_results[$domain] = $result;  
    } else {  
        $unique_results[$domain]["source"] .= ", " . $result["source"];  
    }  
}
```

Рисунок 15 – Код объединения одинаковых ссылок

Функция `print_merged_text_results` (Рисунок 16) сортирует и выводит массив ответов в виде HTML кода через цикл (Рисунок 17). Сортировка является методом ранжирования. Чем больше поисковых систем выдали один и тот же результат выдачи, чем выше этот результат будет.

```
usort($unique_results, function($a, $b) {  
    $a_engines = count(explode(",", $a["source"]));  
    $b_engines = count(explode(",", $b["source"]));  
  
    if ($a_engines == $b_engines) {  
        return 0;  
    }  
  
    return ($a_engines > $b_engines) ? -1 : 1;  
});
```

Рисунок 16 – Сортировка поисковой выдачи

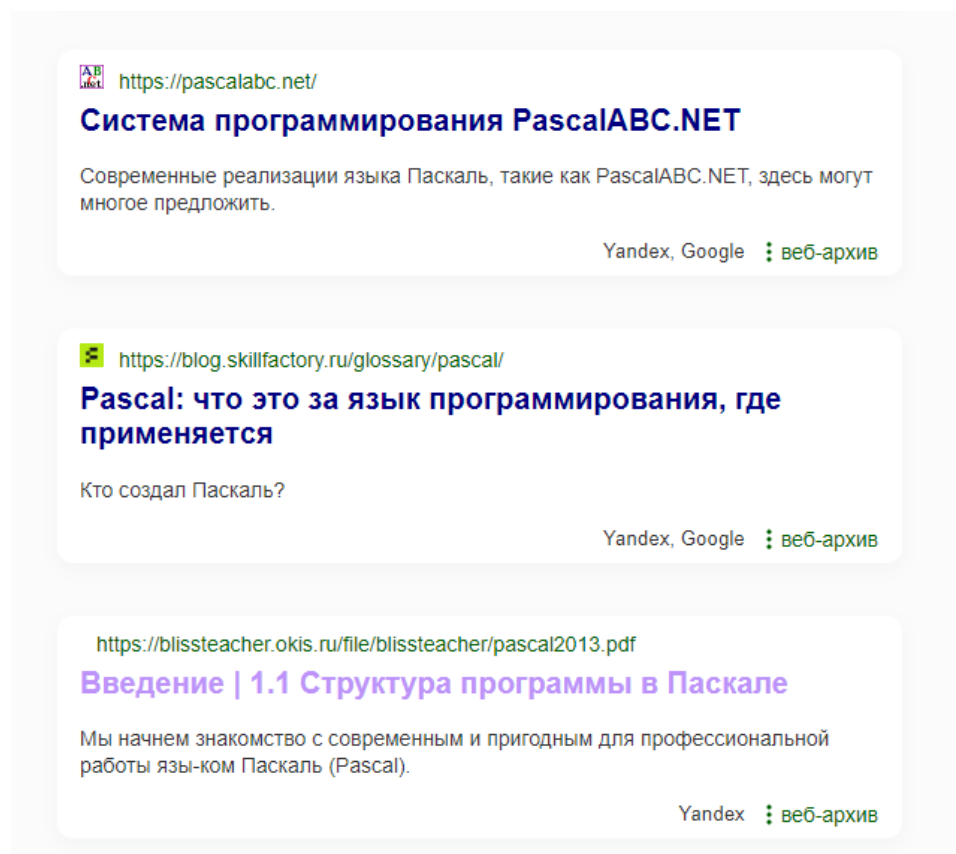
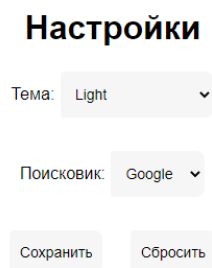


Рисунок 17 – Поисковые ответы из двух источников сверху

4.3 Добавление функции настроек

Настройки метапоисковой системы обычно содержат функции выбора поисковых систем, выбор темы оформления, настройки поиска. В Trydex к файлу настроек settings.php подключается конфигурационный файл config.php для установки настроек по-умолчанию. Таким образом, администратор развертывания продукта устанавливает собственные настройки под нужды своего экземпляра метапоисковой системы.

Важным пунктом в конфигурационном файле является переменная edit_settings, запрещающая (Рисунок 18) или разрешающая (Рисунок 19) менять настройки.



Настройки

Тема: Light ▼

Поисковик: Google ▼

Сохранить Сбросить

Рисунок 18 – Вид настроек с запрещенным изменением

Настройки

Тема: Light

Поисковик: Google

☐ Отключить специальные запросы (например: конвертация валюты)

Альтернативные варианты сервисов

Для примера, если вы хотите просматривать YouTube без слежки, нажмите на "Invidious", найдите наиболее подходящий для вас экземпляр и вставьте его в (правильный формат: `https://example.com`)

Invidious	Replace YouTube
Bibliogram	Replace Instagram
Rimoo	Replace Imgur
Scribe	Replace Medium
Gothub	Replace GitHub
Librarian	Replace Odysee
Nitter	Replace Twitter
Libreddit	Replace Reddit
Proxitok	Replace TikTok
Wikiless	Replace Wikipedia
Quetre	Replace Quora
Libremdb	Replace IMDb
Breezewiki	Replace Fandom
Anonymousoverflow	Replace StackOverflow

☐ Отключить сервисы

Настройки Google

Язык запросов в Google: ru

☐ Безопасный поиск

Рисунок 19 – Вид настроек с разрешенным изменением

На странице с настройками можно выбрать тему оформления и поисковой сервис (Рисунок 20).

Настройки

Тема: Light

Поисковик: Google

Рисунок 20 – Выбор темы оформления и поискового сервиса

Далее, идут параметры для замены популярных сервисов, таких как YouTube, Medium, Wikipedia, на альтернативные сервисы. Такие сервисы специально написаны для защиты конфиденциальности пользователей, отсутствия слежки, трекинга IP и Cookie файлов. Таким образом, в поисковой выдаче вместо youtube.com (Рисунок 21) будет отображен домен сервиса Invidious (Рисунок 22), который указан в настройках (Рисунок 23) [7].

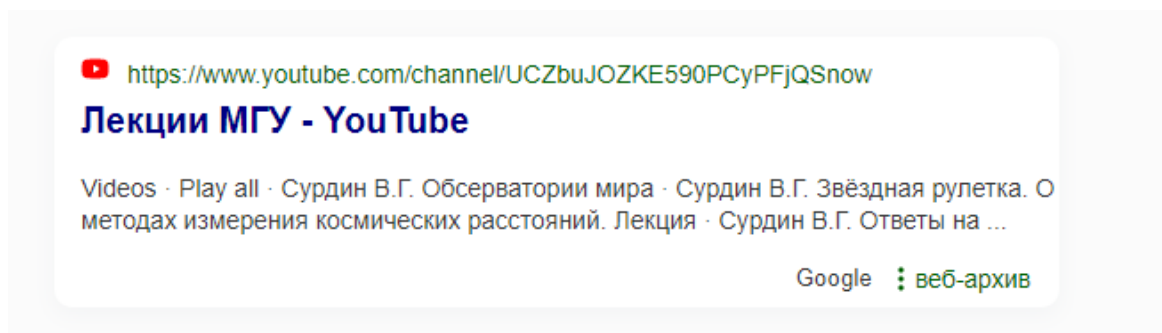


Рисунок 21 – До замены youtube на invidios

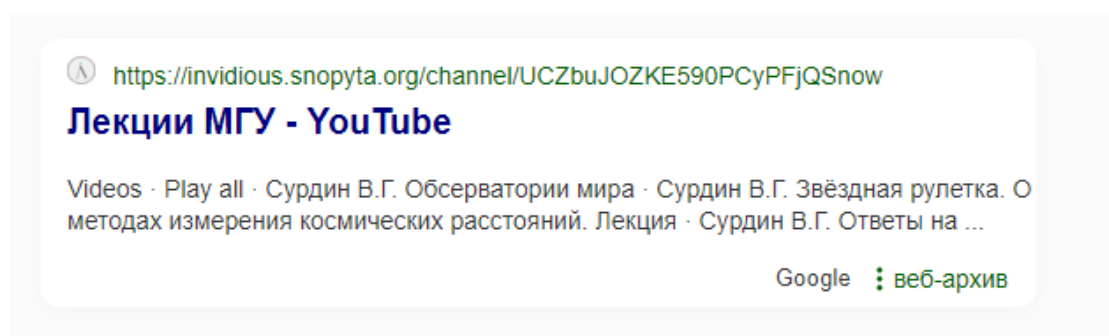


Рисунок 22 – После замены youtube на invidios

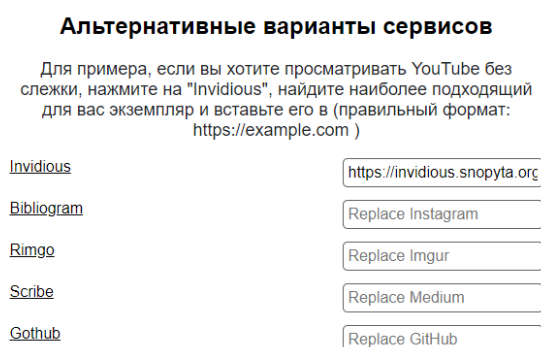


Рисунок 23 – Сайт YouTube заменен на свободный сервис Invidious

За изменение домена отвечает функция `try_replace_with_frontend`, которая из Cookie файлов заменяет домен на указанный из настроек (Рисунок 24).

```
function try_replace_with_frontend($url, $frontend, $original)
{
    $config = require "config.php";

    if (isset($_COOKIE[$frontend]) || isset($_REQUEST[$frontend]) || !empty($config->$frontend))
    {
        if (isset($_COOKIE[$frontend]))
            $frontend = $_COOKIE[$frontend];
        else if (isset($_REQUEST[$frontend]))
            $frontend = $_REQUEST[$frontend];
        else if (!empty($config->$frontend))
            $frontend = $config->$frontend;
    }
}
```

Рисунок 24 – Код замены сервисом на их свободные аналоги

Так же можно отключить альтернативные сервисы, активировав чек-бокс (Рисунок 25).

Bleeping Computer	Replace IMDb
Breezewiki	Replace Fandom
Anonymousoverflow	Replace StackOverflow
Отключить сервисы <input type="checkbox"/>	

Рисунок 25 – Чек-бокс быстрого отключения всех отключения сервисов

Последний пунктом настроек является язык поиска в Google и возможность выбрать безопасный поиск, чтобы исключить опасные сайты, нежелательные для просмотра на работе или детьми (Рисунок 26).

Настройки Google

Язык запросов в Google

ru

Безопасный поиск

☐

Сохранить

Сбросить

Рисунок 26 – Настройки поиска в Google

Таким образом в Cookie файлы содержат только параметры настроек (Рисунок 27).

```
cookie: theme=light; engines=google; invidious=https%3A%2F%2Finvidious.snopyta.org; google_language=ru; save=1
```

Рисунок 27 – Содержание Cookie на сайте Trydex после изменения настроек

4.4 Функция поиска через протокол API

Так как модули поисковых движком возвращают массив данных, с помощью функции `json_encode`, встроенную в PHP, можно с легкостью отобразить этот массив данных в JSON – текстовый формат обмена данными (Рисунок 28).

```
1 // 20230514132146
2 // https://trydex.tk/api.php?q=%D0%BF%D0%B0%D1%81%D0%BA%D0%B0%D0%BB%D1%8C&p=2&t=0
3
4 * [
5 *   {
6 *     "title": "Система программирования PascalABC.NET",
7 *     "url": "https://pascalabc.net/",
8 *     "base_url": "https://pascalabc.net/",
9 *     "description": "Современные реализации языка Паскаль, такие как PascalABC. ... Он создан в 2007 году, основан на языках C# и Delphi (Object Pascal), которые занимают ..."
10 *   },
11 *   {
12 *     "title": "Pascal: что это за язык программирования, где ...",
13 *     "url": "https://blog.skillfactory.ru/glossary/pascal/",
14 *     "base_url": "https://blog.skillfactory.ru/",
15 *     "description": "24 мар. 2023 г. - Pascal - это универсальный язык программирования, отличающийся строгой структурой и типизацией переменных, а также интуитивно понятным ..."
16 *   },
17 * ]
```

Рисунок 28 – Вывод JSON ответа через API Trydex

Данный модуль предназначен для интеграции с другими сервисами. К примеру, Телеграм бот для поиска внутри мессенджера (Рисунок 29).

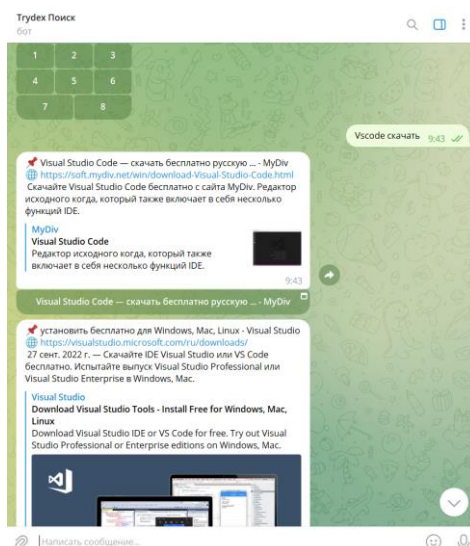


Рисунок 29 – Телеграм-бот, использующий API Trydex

4.5 Отображение поискового ответа в виде HTML страницы

За отображение выдачи ответов в виде HTML страницы используется модуль `search.php`. Данный модуль через GET запрос получает параметры поиска, такие как текстовый запрос, номер страницы, поисковой сервис (Рисунок 30).

 <https://trydex.tk/search.php?q=паскаль&t=0&p=0>

Рисунок 30 – GET запрос на модуль `search.php`

Данный модуль через оператор `switch` получает значение название поискового сервиса и запускает модули поисковых движков (Рисунок 31).

```

$start_time = microtime(true);
switch ($type)
{
    case 0:
        require "engines/text/merge.php";
        $results = get_merged_text_results($query_encoded, $page);
        print_elapsed_time($start_time);
        print_merged_text_results($results);
        break;

    case 1:
        require "engines/crowdview/forums.php";
        $results = get_text_results($query_encoded, $page);
        print_elapsed_time($start_time);
        print_text_results($results);
        break;

    case 2:
        require "engines/brave/video.php";
        $results = get_video_results($query_encoded);
        print_elapsed_time($start_time);
        print_video_results($results);
        break;
}

```

Рисунок 31 – Код выбора типа поиска (текст, форумы, видео)

В данном примере из GET запроса получает значение $t=0$, которая относится к текстовому поиску. Если t будет равняться 1, то будет произведен поиск по форумам, а если $t=2$, то поиск по видео.

4.6 Схема сайта

На рисунке 32 схематически представлена структурная схема сайта.

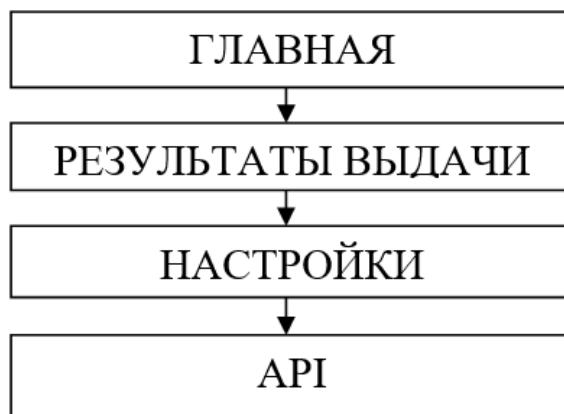


Рисунок 32 – Структурная схема сайта

Главная страница сайта – это самая первая страница, на которую попадает пользователь, введя в адресной строке браузера адрес метапоисковой сети <https://trydex.org.ru> [8]. Главная страница представлена на рисунке 33.

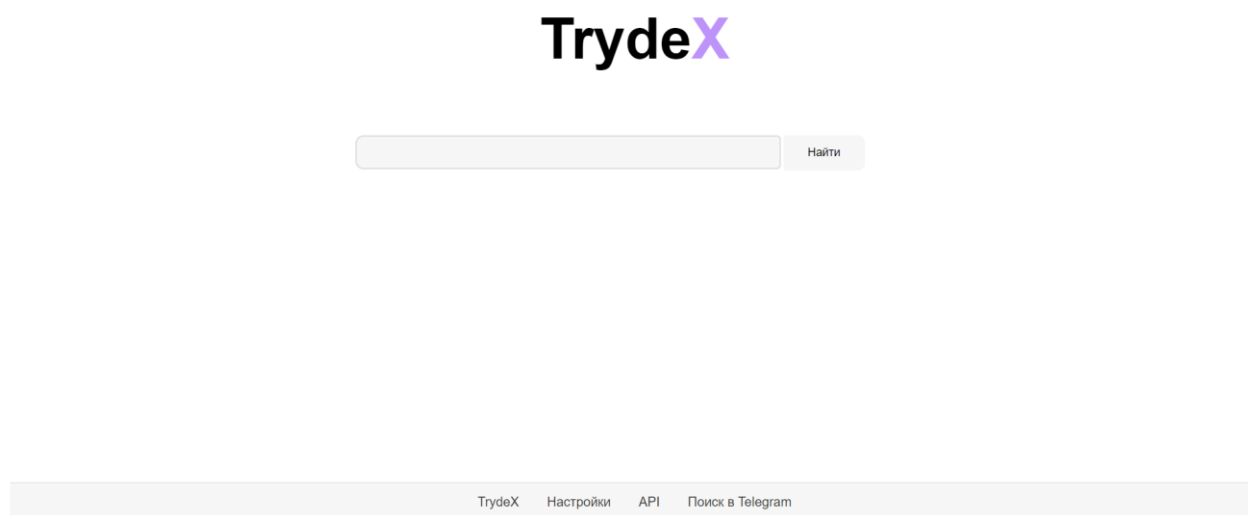


Рисунок 33 – Главная страница сайта Trydex

Страница сайта с поисковыми ответами перенаправляется из главной страницы по нажатию кнопки «Найти» (Рисунок 34).

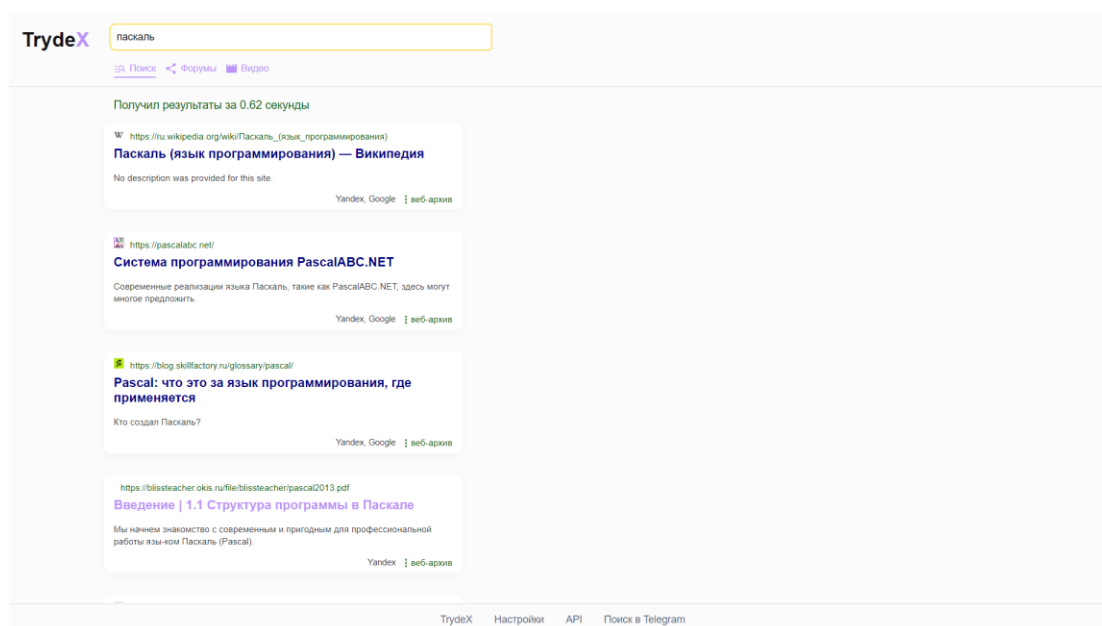


Рисунок 34 – Страница поисковой выдачи ответов

Данные страницы оптимизированы для просмотра на мобильных устройствах (Рисунок 35).

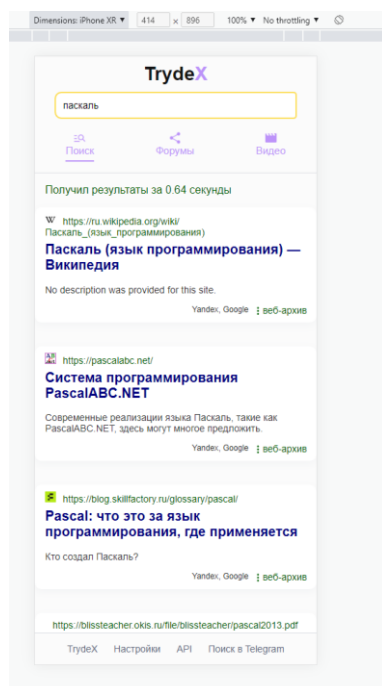


Рисунок 35 – Мобильная версия сайта

4.7 Требования к ПК для просмотра сайта

Для просмотра информационного сайта на ПК должна быть установлена любая операционная система. Для визуального отображения сайта, необходим браузер Internet Explorer, Yandex, Google Chrome или другой.

Рекомендуемое разрешение экрана – 1280x800px.

Так как сайт содержит мало графических элементов, загружается он довольно быстро, даже при медленном соединении с Интернетом, и не потребляет много ресурсов компьютера (Рисунок 36).

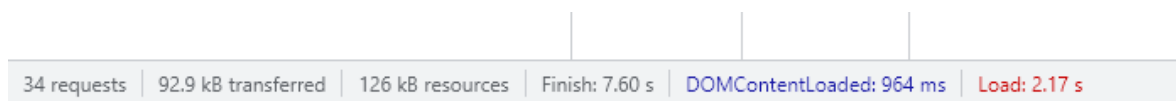


Рисунок 36 – Страница поисковой выдачи оптимизирована и имеет легкий вес

4.8 Размещение сайта в сети Интернет

У центра Поиск есть собственный домен <https://stavdeti.ru>. Но для размещения метапоисковой сети Trydex необходим отдельный домен. Для разработки и тестирования Trydex выбран домен trydex.tk. Данный домен бесплатно делегирован у провайдера freenom. Для рабочей версии продукта зарегистрирован домен trydex.org.ru у провайдера reg.ru [9].

Для размещения готовой версии продукта выбран хостинг-провайдер TimeWeb. Это популярный хостинг, имеющий высокий uptime, предлагающий выгодные тарифы. Данный хостинг работает на рынке 17 лет и имеет круглосуточную поддержку. Наличие собственной панели управления облегчает размещение сайта через встроенный FTP менеджер.

У компании имеется лицензия на предоставление телематических услуг связи № 142739. Хостинг предоставляется на платформе UNIX.

Загрузка данных сайта на сервер была загружена при помощи SSH-консоли. Введя команду `git clone https://github.com/LencoDigitexer/trydex`, на сайт загрузились файлы метапоисковой системы из системы управления версиями GitHub

Теперь отредактируем файл `config.php`, изменив в нем параметры настроек. Полная документация по размещению метапоисковой сети Trydex находится по адресу <https://github.com/LencoDigitexer/trydex/wiki/EN-Wiki>

4.9 Тестирование сайта

На данном этапе сайт полностью готов, осталось его протестировать на наличие ошибок [10].

Вначале проверим, как сайт отображается в разных браузерах. Возьмем три самых популярных браузера: Yandex, Google Chrome, Firefox. Результат представлен на рисунках 37, 38, 39. Дополнительно, проведено тестирование в браузере Safari (Рисунок 40).

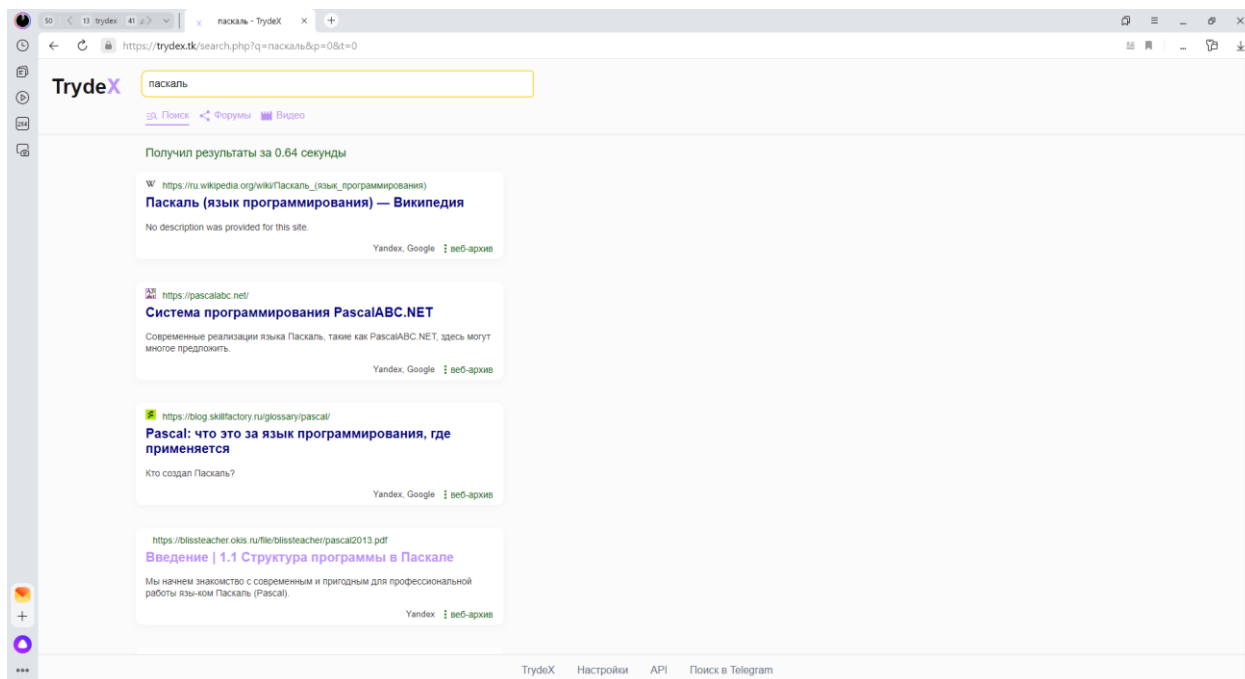


Рисунок 37 – Просмотр сайта через браузер Yandex

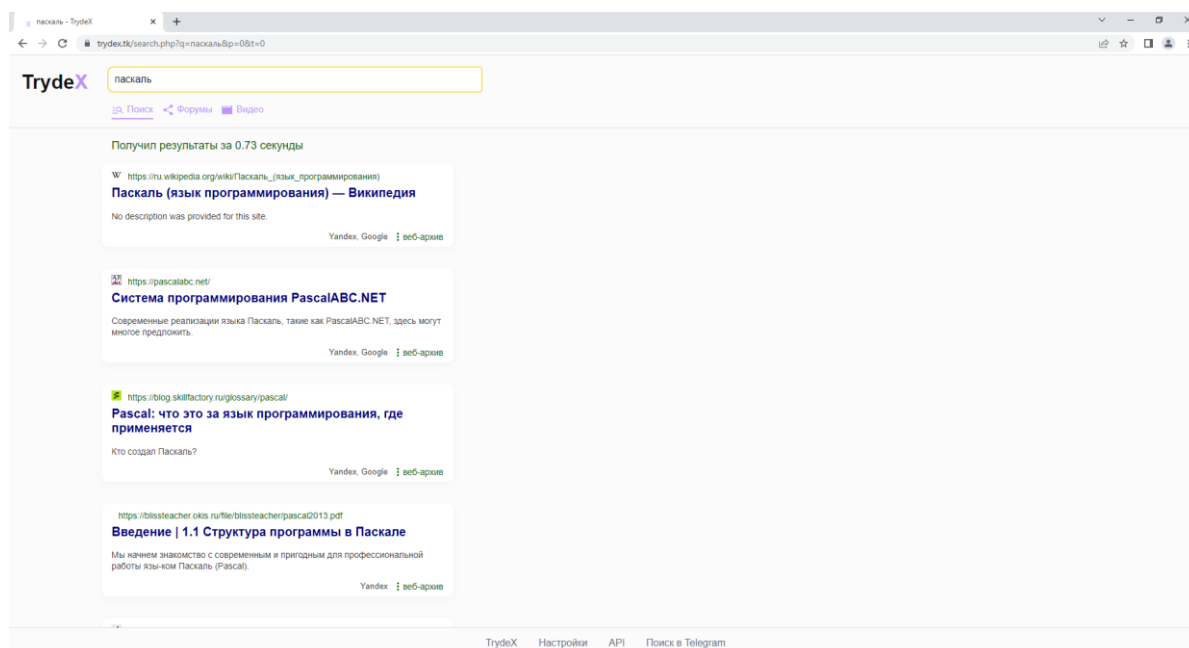


Рисунок 38 – Просмотр сайта через браузер Google Chrome

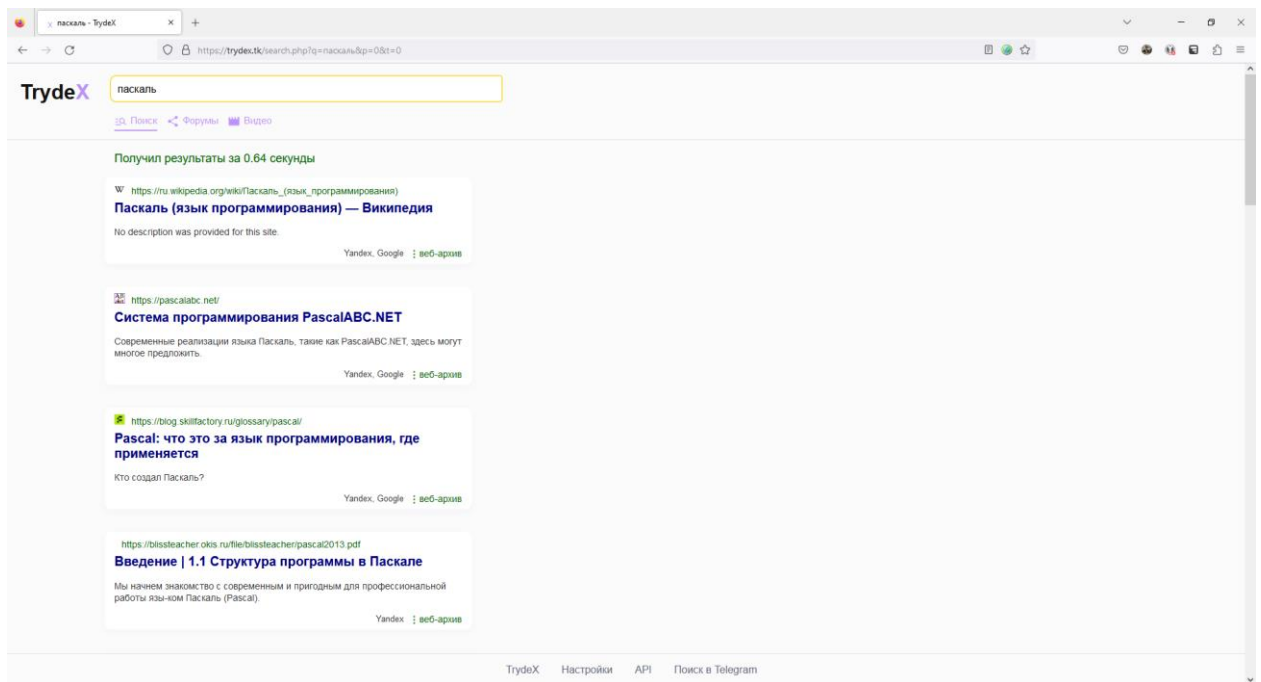


Рисунок 39 – Просмотр сайта через браузер Firefox

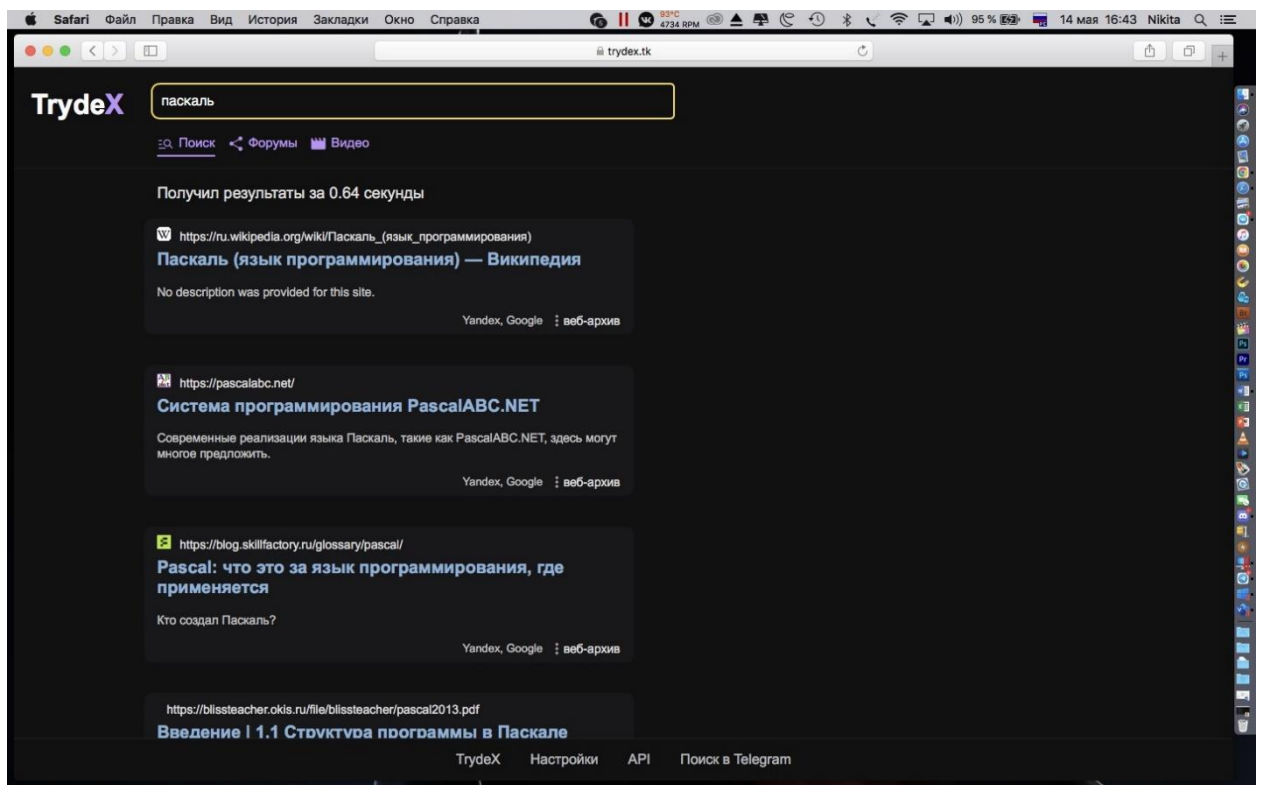


Рисунок 40 – Просмотр сайта через браузер Safari

При просмотре данного веб-сайта, во всех браузерах он отображается одинаково, без смещений. Это говорит о том, что он сделан качественно, с учетом всех стандартов веб-дизайна.

Далее, проверим сайт на наличие ошибок в исходном коде. PHP выводит ошибки на саму страницу, если они имеются. В процессе тестирования не отобразилось ни одной ошибки. Т.к. в процессе создания сайта не использовался JavaScript, то в консоли ошибок ничего не будет выведено [11].

При тестировании сайта было отмечено, что он довольно быстро загружается. Это положительно скажется на количестве постоянных посетителей сайта, так как на сайте, который быстро загружается, приятней находиться. На этом разработка сайта завершена.

ЗАКЛЮЧЕНИЕ

В заключение, выпускная квалификационная работа по метапоисковой системе Trydex показала, что эта система представляет собой инновационный подход к поиску информации в сети Интернет. Trydex объединяет несколько поисковых систем и позволяет получить более точные и полные результаты поиска.

В работе были рассмотрены основные принципы разработки и работы метапоисковой системы Trydex. Было установлено, что Trydex имеет ряд преимуществ перед традиционными поисковыми системами: более точные результаты поиска, возможность фильтрации результатов и удобный интерфейс.

Значимость дипломной работы по мета-поисковой системе Trydex проявляется в нескольких аспектах.

Во-первых, данная работа имеет актуальное значение в контексте современного информационного общества. С развитием Интернета и ростом объемов информации становится все сложнее искать нужные данные среди огромного количества различных поисковых систем. Метапоисковая система Trydex предлагает инновационный подход, объединяя несколько поисковых систем и предоставляя более точные и полные результаты поиска. Это существенно упрощает задачу пользователей в поиске нужной информации и повышает эффективность процесса.

Во-вторых, работа имеет практическую значимость для различных пользователей. Trydex может быть полезен для обычных пользователей, которым требуется быстрый и точный поиск информации в Интернете. Также, система может быть ценным инструментом для предприятий и бизнеса, которым необходимо проводить исследования рынка, анализировать конкурентов и находить актуальные данные. Более точные результаты поиска и удобный интерфейс Trydex делают его привлекательным решением для различных категорий пользователей.

В-третьих, данная работа может способствовать дальнейшему развитию и улучшению метапоисковой системы Trydex. Исследовательские выводы и рекомендации, сделанные на основе проведенных исследований, могут служить основой для дальнейшей оптимизации системы, расширения ее функциональности и улучшения пользовательского опыта. Развитие Trydex в соответствии с потребностями пользователей и требованиями рынка информационных технологий может привести к его дальнейшей успешной адаптации и применению в различных сферах исследования и бизнеса.

Дальнейшее развитие и совершенствование системы может привести к еще более точным и всеобъемлющим результатам поиска, что сделает Trydex еще более востребованным на рынке.

Удобный интерфейс является одним из факторов, способствующих успешной адаптации системы Trydex у широкого круга пользователей. Простота в использовании, интуитивно понятные элементы управления и логичная структура интерфейса делают процесс поиска информации более эффективным и комфортным.

Перспективы развития и совершенствования системы Trydex выглядят многообещающими. Учитывая динамичный характер Интернета и необходимость в постоянном обновлении и расширении системы, дальнейшее развитие Trydex может включать интеграцию новых поисковых источников, улучшение алгоритмов ранжирования результатов, оптимизацию производительности и расширение функциональности.

Предполагается, что с улучшением алгоритмов Trydex будет способна предоставлять еще более точные и всеобъемлющие результаты поиска. Это сделает систему более конкурентоспособной на рынке информационных технологий и привлекательной для широкого круга пользователей.

В целом, выпускная квалификационная работа подтверждает значимость и перспективность метапоисковой системы Trydex. Развитие системы и учет потребностей пользователей способны привести к еще более

успешной интеграции Trydex на рынке информационных технологий и повышению ее популярности.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. ГОСТ Р 51188-98 «Защита информации. Испытания программных средств на наличие компьютерных вирусов. Типовое руководство»
2. GitHub репозиторий trydex с исходными данными – URL: <https://github.com/lencodigitexer/trydex>
3. Белугина, С. В. Разработка программных модулей программного обеспечения для компьютерных систем. Прикладное программирование: учебное пособие для спо / С. В. Белугина. – 3-е изд., стер. – Санкт-Петербург: Лань, 2022. – 312 с. – ISBN 978-5-8114-9817-8. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/200390>
4. Зубкова, Т. М. Технология разработки программного обеспечения: учебное пособие для спо / Т. М. Зубкова. – 2-е изд., стер. – Санкт-Петербург: Лань, 2022. – 252 с. – ISBN 978-5-8114-9556-6. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/200462>
5. Макарова, Н.В. Основы программирования: учебник / Макарова Н.В., Нилова Ю.Н., Зеленина С.Б., Лебедева Е.В. – Москва: КноРус, 2021. – 451 с. – ISBN 978-5-406-03394-4. – URL: <https://book.ru/book/936582>
6. Мэтт Зандстра, Игорь Владимирович Красиков PHP 8: объекты, шаблоны и методики программирования [6-е издание] 2019. – 868 с.
7. Назаров, С. В. Архитектура и проектирование программных систем: монография / С.В. Назаров. – 2-е изд., перераб. и доп. – Москва: ИНФРА-М, 2023. – 374 с. – DOI 10.12737/18292. – ISBN 978-5-16-011753-9. – Текст: электронный. – URL: <https://znanium.com/catalog/product/1895672>
8. Нуретдинов Р.И. Разработка веб-приложений средствами языка программирования PHP / Учебное пособие // Санкт-Петербург – 2021. – 48 с.
9. Попов, А.А. Эргономика пользовательских интерфейсов в информационных системах: учебное пособие / Попов А.А. – Москва: КноРус, 2021. – 304 с. – ISBN 978-5-406-07634-7. – URL: <https://book.ru/book/938669>

10. Проскуряков, А. В. Качество и тестирование программного обеспечения. Метрология программного обеспечения: учебное пособие / А. В. Проскуряков; Южный федеральный университет. – Ростов-на-Дону; Таганрог: Издательство Южного федерального университета, 2022. - 197 с. - ISBN 978-5-9275-4044-0. – Текст: электронный. – URL: <https://znanium.com/catalog/product/2057599>

11. Руденко, О.Г. Системный анализ и проектирование информационных систем: учебное пособие / Руденко О.Г., Кожухова С.В. – Москва: ИНФРА-М, 2021. – 216 с.

12. Ступина М.В. Введение в веб-разработку на языке PHP // Учебное пособие / Ростов-на-Дону. гос. ун-т. – 2022. – 95 с.

13. Уильям Стейнмец, Брайан Вард, PHP: 75 готовых решений для вашего web-сайта. Самоучитель 2021.

14. Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем. Учебник, – М.: Издательский центр «Академия», 2019. – 384 с.

15. Шикина, В. Е. Техническая документация информационных систем: учебное пособие / В.Е. Шикина. – Ульяновск: УлГТУ, 2018. – 92 с. – ISBN 978-5-9795-1852-7. – URL: <http://lib.ulstu.ru/venec/disk/2017/460.pdf>

ПРИЛОЖЕНИЕ А

Техническое задание на разработку метапоисковой системы для ЦДО «Поиск»

1. Общие сведения

1.1. Доменное имя

trydex.org.ru

1.2. Название сайта

Trydex

1.3. Назначение сайта

Назначение сайта Trydex – предоставление пользователям возможности осуществлять поиск информации в нескольких поисковых системах одновременно. Сайт является метапоисковой системой, которая позволяет собирать результаты поиска из различных источников и предоставлять их пользователю в удобном виде. Основная цель сайта – облегчить процесс поиска информации в Интернете, сократить время, затрачиваемое на поиск и улучшить качество результатов.

2. Назначение и цели создания сайта

2.1. Целевая аудитория

Пользователи Интернет – ресурса (от 16 лет и старше).

2.2. Внутренние интересы

Заказчик должен получить динамичный, красивый сайт со всеми работающими функциями.

2.3. Внешние интересы

Пользователи сайта получают возможность осуществлять поиск информации в нескольких поисковых системах одновременно

3. Требования к сайту

3.1. Сайт должен содержать следующие html страницы:

– главная;

– настройки;

- API;
- страница результатов.

Сайт должен отображаться во всех популярных браузерах.

3.2. Кнопки управления (навигация сайта)

1. С каждой страницы сайта должен быть обеспечен переход (установлена гиперссылка) на главную страницу сайта.

2. Язык сайта – русский.

3. Содержимое сайта будет вводиться через модули поисковых движков.

4. Состав и содержание работ по созданию сайта

4.1. Структура сайта

1. Главная страница:

- поле для ввода поискового запроса;
- кнопка «Поиск»;
- меню навигации.

2. Страница результатов поиска:

- поле для ввода поискового запроса;
- список результатов поиска из различных поисковых систем.

3. Страница настроек:

- список доступных поисковых систем и возможность выбора тех, которые будут использоваться при поиске;
- выбор темы оформления;
- таблица для замены сервисов на их свободные аналоги.

5. Требования к программной совместимости

5.1. Совместимость с ранними версиями программных продуктов

Текущая версия РНР 5.0 или выше.

6. Специальные требования

6.1. Требования к надежности

Для обеспечения надежного и устойчивого функционирования программы необходимо выполнение Заказчиком совокупности организационно-технических мероприятий, включающих в себя:

- организацию бесперебойного питания технических средств, что обеспечит непрерывность работы программы в случае отключения электроэнергии;

- использование лицензионного программного обеспечения, что гарантирует соответствие программы стандартам и нормативным документам;

- регулярное выполнение требований ГОСТ 51188-98 «Защита информации. Испытания программных средств на наличие компьютерных вирусов», что обеспечит защиту программы от вредоносных программных воздействий; [1]

- обеспечение возможности восстановления сайта после сбоя и других факторов в течение 3-5 минут, что обеспечит быстрое восстановление работы программы в случае возникновения сбоев.

Кроме того, необходимо обеспечить удобство использования технических средств и обеспечить интуитивно понятные интерфейсы и механизмы управления. Для этого должны быть использованы современные технологии, обеспечивающие удобство использования и удобный интерфейс для пользователей. Также необходимо обеспечить возможность удаленного управления и мониторинга, что позволит оперативно реагировать на возможные проблемы и сбои в работе программы.

6.2. Требования к защите информации от несанкционированного доступа

Защита информации

Обеспечение надежной и стабильной работы сайта, исключающей возможность сбоев и прерываний в работе.

Гарантирование бесперебойной работы всех функций сайта, что обеспечит его эффективное использование.

Обеспечение защиты сайта от вредоносных программных воздействий, таких как вирусы и другие факторы, что гарантирует сохранность информации и надежность работы сайта.

6.3. Требования по сохранности информации при авариях

При авариях различных видов: взломы, сбои в электроэнергии и различные другие факторы – данные должны остаться не тронутыми!

7. Требования к программной документации

7.1. Требования к документации по администрированию системы

Все данные должны быть представлены в виде README файлов, поставляемых вместе с дистрибутивом продукта

8. Порядок сдачи-приемки работ

- срок разработки сайта: до 1 июня 2023 года;
- программист передает сайт заказчику;
- заказчик обязан проверить наличие грамматических, орфографических и смысловых ошибок на сайте в течение трёх рабочих дней. Обнаруженные ошибки программист обязан устранить в течение трёх рабочих дней;
- программист не несет ответственности за несоответствие сайта эстетическим ожиданиям заказчика при условии выполнения технического задания на сайт.

ПРИЛОЖЕНИЕ Б

Исходный код сайта метапоисковой системы

index.php:

```
<?php require «misc/header.php»; ?>
<title>Trydex</title>
</head>
<body>
<form class=«search-container» action=«search.php» method=«get»
autocomplete=«off»>
<h1>Tryde<span class=«X»>X</span></h1>
<div class=«search-button-wrapper»>
<input type=«text» name=«q» autofocus/>
<input type=«hidden» name=«p» value=«0»/>
<input type=«hidden» name=«t» value=«0»/>
<input type=«submit» class=«hide»/>
<button name=«t» value=«0» type=«submit»>Найти</button>
</div>
</form>
<?php require «misc/footer.php»; ?>
```

config.php:

```
<?php
$domain = $config->google_domain;
$language = isset($_COOKIE[«google_language»]) ?
htmlspecialchars($_COOKIE[«google_language»]) : $config->google_language;
$google_url =
«https://www.google.$domain/search?q=$query&start=$page&hl=$language&lr
```

```

=lang_$language&asearch=arc&async=use_ac:true,_fmt:prog»;
if (isset($_COOKIE[«safe_search»])) {
$google_url .= «&safe=medium»;}
function get_google_results($response, $page)
{ global $config;
$results = array();
$xpath = get_xpath($response);
foreach ($xpath->query(«//div//div[contains(@class, 'g')]») as $result) {
$url = $xpath->evaluate(«//div[@class='yuRUbf']//a/@href», $result)[0];
if ($url == null)
continue;
if (!empty($results)) // filter duplicate results, ignore special result
{ if (!array_key_exists(«special_response», end($results)))
if (end($results)[«url»] == $url->textContent)
continue;}
$url = mb_convert_encoding($url->textContent, «ISO-8859-1», «UTF-8»);
$url = check_for_privacy_frontend($url);
$title = $xpath->evaluate(«//h3», $result)[0];
$title = mb_convert_encoding($title->textContent, «ISO-8859-1», «UTF-8»);
$description = $xpath->evaluate(«//div[contains(@class, 'VwiC3b')]», $result)[0];
$description = mb_convert_encoding($description->textContent, «ISO-8859-1»,
«UTF-8»);
array_push(
$results,
array(
«title» => htmlspecialchars($title),
«url» => htmlspecialchars($url),
«base_url» => htmlspecialchars(get_base_url($url)),
«description» => $description == null ?
«No description was provided for this site.» :

```

```

htmlspecialchars($description),
«source» => «Google»
));}
return $results;}?>

```

text.php:

```

<?php
function get_merged_torrent_results($query, $page)
{ global $config;
require «engines/text/yandex.php»;
require «engines/text/google.php»;
//$query = urlencode($query);
$query_encoded = urlencode($query);
$torrent_urls = array(
$yandex_url,
$google_url);
$mh = curl_multi_init();
$chs = $results = array();
foreach ($torrent_urls as $url)
{ $ch = curl_init($url);
curl_setopt_array($ch, $config->curl_settings);
array_push($chs, $ch);
curl_multi_add_handle($mh, $ch);  }
$running = null;
do {
curl_multi_exec($mh, $running);
} while ($running);
for ($i=0; count($chs)>$i; $i++)
{ $response = curl_multi_getcontent($chs[$i]);

```

```

switch ($i)
{ case 0:
$temp_results = get_yandex_results($response, $page);
break;
case 1:
$temp_results = get_google_results($response, $page);
break;}
foreach ($temp_results as $result) {
$domain = $result[«url»];
if (!isset($unique_results[$domain])) {
$unique_results[$domain] = $result;
} else {
$unique_results[$domain][«source»] .= «, « . $result[«source»];
}}}
$special_search = $page ? 0 : check_for_special_search($query);
if ($special_search != 0) {
switch ($special_search) {
case 1:
$url = «https://cdn.moneyconvert.net/api/latest.json»;
break;
case 2:
$split_query = explode(« », $query);
$reversed_split_q = array_reverse($split_query);
$word_to_define = $reversed_split_q[1];
$url = «https://api.dictionaryapi.dev/api/v2/entries/en/$word_to_define»;
break;
case 5:
$url = «https://wttr.in/@». $_SERVER[«REMOTE_ADDR»]. «?format=j1»;
break;
case 6:

```

```

$url = «https://check.torproject.org/torbulkexitlist»;
break;
case 7:
$url =
«https://ru.wikipedia.org/w/api.php?format=json&action=query&prop=extracts%7
Cpageimages&exintro&explaintext&redirects=1&pithumbsize=500&titles=$query
_encoded»;
break;
}
if ($url != NULL) {
$special_ch = curl_init($url);
curl_setopt_array($special_ch, $config->curl_settings);
curl_multi_add_handle($mh, $special_ch);} }
print_r($special_search);
echo $special_ch;
if ($special_search != 0) {
$special_result = null;
switch ($special_search) {
case 1:
require «engines/special/currency.php»;
$special_result = currency_results($query, curl_multi_getcontent($special_ch));
break;
case 2:
require «engines/special/definition.php»;
$special_result = definition_results($query, curl_multi_getcontent($special_ch));
break;
case 3:
require «engines/special/ip.php»;
$special_result = ip_result();
break;

```



```

case 4:
require «engines/special/user_agent.php»;
$special_result = user_agent_result();
break;

case 5:
require «engines/special/weather.php»;
$special_result = weather_results(curl_multi_getcontent($special_ch));
break;

case 6:
require «engines/special/tor.php»;
$special_result = tor_result(curl_multi_getcontent($special_ch));
break;

case 7:
require «engines/special/wikipedia.php»;
$special_result = wikipedia_results($query, curl_multi_getcontent($special_ch));
break;}

if ($special_result != null)
array_push($unique_results, $special_result);}

return $unique_results; }

function print_merged_torrent_results($unique_results)
{ $special = end($unique_results);
if (array_key_exists(«special_response», $special)) {
$response = $special[«special_response»][«response»];
$source = $special[«special_response»][«source»];
echo «<p class=\»special-result-container\»>«;
if (array_key_exists(«image», $special[«special_response»])) {
$image_url = $special[«special_response»][«image»];
echo «<img src=\»image_proxy.php?url=$image_url\»>«;}
echo $response;
if ($source)

```

```

echo «<a href=\»$source\» target=\»_blank\»>$source</a>«;
echo «</p>«;
echo «<div class=\»text-result-container\»>«;
if (!empty($unique_results))
{
    usort($unique_results, function($a, $b) {
        $a_engines = count(explode(«,», $a[«source»]));
        $b_engines = count(explode(«,», $b[«source»]));
        if ($a_engines == $b_engines) {
            return 0;
        }
        return ($a_engines > $b_engines) ? -1 : 1;
    });
    foreach($unique_results as $result)
    {
        $title = $result[«title»];
        $url = $result[«url»];
        $base_url = $result[«base_url»];
        $description = $result[«description»];
        $domain = parse_url($base_url)[«host»];
        $source = $result[«source»];
        echo «<div class=\»text-result-wrapper\»>«;
        echo «<img class=\»favicon-wrapper\»
            src=\»image_proxy.php?url=https://favicon.yandex.net/favicon/$domain\»>«;
        echo «<a href=\»$url\»>«;
        echo urldecode($url);
        echo «<h2>$title</h2>«;
        echo «</a>«;
        echo «<span>$description</span><br>«;
        echo '<div class=«engines»> <span>' . $source . '</span>
            <a href=«https://web.archive.org/web/» class=«cache_link» rel=«noreferrer»
            <svg class=«ion-icon-small» viewBox=«0 0 512 512» aria-hidden=«true»
            <circle cx=«256» cy=«256» r=«32» fill=«none» stroke=«currentColor» stroke-
            miterlimit=«10» stroke-width=«32»></circle>

```

```

<circle cx=«256» cy=«416» r=«32» fill=«none» stroke=«currentColor» stroke-
miterlimit=«10» stroke-width=«32»></circle>
<circle cx=«256» cy=«96» r=«32» fill=«none» stroke=«currentColor» stroke-
miterlimit=«10» stroke-width=«32»></circle>
</svg>веб-архив</a>&lrn;
</div>';
echo «</div>«; } }
else
echo «<p>Ничего не найдено, попробуйте изменить поисковой запрос</p>«;
echo «</div>«; } ?>

```

Файл search.php:

```

<?php
require «misc/header.php»;
$config = require «config.php»;
require «misc/tools.php»;?>
<title>
<?php
$query = htmlspecialchars(trim($_REQUEST[«q»]));
echo $query;
?> - TrydeX</title>
</head>
<body>
<form class=«sub-search-container» method=«get» autocomplete=«off»>
<h1 class=«logomobile»><a class=«no-decoration» href=«./»>Tryde<span
class=«X»>X</span></a></h1>
<input type=«text» name=«q»
<?php
$query_encoded = urlencode($query);

```

```

if (1 > strlen($query) || strlen($query) > 256)
{header(«Location: ./»);
die();}
echo «value=\»$query\»«;?>
>
<br>
<?php
foreach ($_REQUEST as $key => $value)
{if ($key != «q» && $key != «p» && $key != «t»)
{echo «<input type=\»hidden\» name=\»« . htmlspecialchars($key) . «\» value=\»«
. htmlspecialchars($value) . «\»/>«;}}
$type = isset($_REQUEST[«t»]) ? (int)$_REQUEST[«t»] : 0;
echo «<button class=\»hide\» name=\»t\» value=\»$type\»/></button>«;
?>
<button type=«submit» class=«hide»></button>
<input type=«hidden» name=«p» value=«0»>
<div class=«sub-search-button-wrapper»>
<?php
$categories = array(
«Поиск»,
«Форумы»,
«Видео»);
foreach ($categories as $category)
{$category_index = array_search($category, $categories);
if (($config->disable_bittorent_search && $category_index == 3) || ($config-
>disable_hidden_service_search && $category_index == 4))
{continue;}
echo «<a « . (($category_index == $type) ? «class=\»active\»« : «») .
«href=\»/search.php?q=« . $query . «&p=0&t=« . $category_index . «\»><img
src=\»static/images/\» . $category . «_result.png\» alt=\»« . $category . « result\»

```

```

/>« . ucfirst($category) . «</a>«; }?>
</div>
<hr>
</form>
<?php
$page = isset($_REQUEST[«p»]) ? (int)$_REQUEST[«p»] : 0;
$start_time = microtime(true);
switch ($type)
{ case 0:
require «engines/text/merge.php»;
$results = get_merged_text_results($query_encoded, $page);
print_elapsed_time($start_time);
print_merged_text_results($results);
break;
case 1:
require «engines/crowdview/forums.php»;
$results = get_text_results($query_encoded, $page);
print_elapsed_time($start_time);
print_text_results($results);
break;
case 2:
require «engines/brave/video.php»;
$results = get_video_results($query_encoded);
print_elapsed_time($start_time);
print_video_results($results);
break;}?>
<?php require «misc/footer.php»; ?>

```