Государственное бюджетное профессиональное образовательное учреждение «Дзержинский педагогический колледж»

**Методические разработки самостоятельных занятий по**

**ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ 09.**

***Проектирование, разработка и оптимизация веб-приложений***

**Дзержинск – 2017-2021**

|  |  |
| --- | --- |
| **Одобрено на заседании ПЦК преподавателей спец. Информатика****Протокол №\_\_\_\_\_\_\_от\_\_\_\_\_\_\_\_****Председатель ПЦК \_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_/** **Составитель: Пучкина Т.С.,  Руденко Н.А.** | **Методические разработки самостоятельных занятий составлены в соответствии с требованиями федеральных государственных образовательных стандартов среднего профессионального образования по спец. 09.02.07. Информационные системы и программирование**  |

Методические разработки самостоятельной работы занятий по ПМ 09. Проектирование, разработка и оптимизация веб-приложений специальность 09.02.07. Информационные системы в программировании. Представленные, в данных методических указаниях, самостоятельных задания направлены на формирование знаний и умений по дисциплине, а так же общих и профессиональных компетенций:

**ПК.9.1Разрабатывать техническое задание на веб-приложение в соответствии с требования-ми заказчика.**

***Иметь практический опыт:***

* Сбора предварительных данных для выявления требований к веб-приложению.
* Определения первоначальных требований заказчика к веб-приложению и возможности их реализации.
* Подбора оптимальных вариантов реализации задач и согласование их с заказчиком.
* Оформления технического задания.

***Уметь:***

* Проводить анкетирование.
* Проводить интервьюирование.
* Оформлять техническую документацию.
* Осуществлять выбор одного из типовых решений.
* Работать со специализированным программным обеспечением для планирования времени и организации работы с клиентами.

***Знать:***

* Инструменты и методы выявления требований.
* Типовые решения по разработке веб-приложений.
* Нормы и стандарты оформления технической документации.
* Принципы проектирования и разработки информационных систем

**ПК 9.2. Разрабатывать веб-приложение в соответствии с техническим заданием.**

**Иметь практический опыт:**

* Выполнять верстку страниц веб-приложений.
* Кодировать на языках веб-программирования.
* Разрабатывать базы данных.
* Использовать специальные готовые технические решения при разработке веб-приложений.
* Выполнять разработку и проектирование информационных систем.

**Уметь:**

* Разрабатывать программный код клиентской и серверной части веб-приложений.
* Использовать язык разметки страниц веб-приложения.
* Оформлять код программы в соответствии со стандартом кодирования.
* Использовать объектные модели Веб-приложений и браузера.
* Использовать открытые библиотеки (framework).
* Использовать выбранную среду программирования и средства системы управления базами данных.
* Осуществлять взаимодействие клиентской и серверной частей Веб-приложений.
* Разрабатывать и проектировать информационные системы

**Знать:**

* Языки программирования и разметки для разработки клиентской и серверной части веб-приложений.
* Принципы работы объектной модели веб-приложений и браузера.
* Основы технологии клиент-сервер.
* Особенности отображения веб-приложений в размерах рабочего пространства устройств.
* Особенности отображения элементов ИР в различных браузерах.
* Особенности выбранной среды программирования и системы управления базами данных.

**ПК.9.3 Разрабатывать интерфейс пользователя веб-приложений в соответствии с техническим заданием.**

**Иметь практический опыт:**

* Разрабатывать интерфейс пользователя.
* Разрабатывать анимационные эффекты

**Уметь:**

* Разрабатывать программный код клиентской части Веб-приложений.
* Оформлять код программы в соответствии со стандартом кодирования.
* Использовать объектные моделиВеб-приложений и браузера.
* Разрабатывать анимацию для Веб-приложений для повышения его доступности и визуальной привлекательности (Canvas).

**Знать:**

* Языки программирования и разметки для разработки клиентской части веб-приложений.
* Принципы работы объектной модели Веб-приложений и браузера.
* Технологии для разработки анимации.
* Способы манипуляции элементами страницы веб-приложения.
* Виды анимации и способы применения ее.

**ПК 9.4. Осуществлять техническое сопровождение и восстановление веб-приложений в соответствии с техническим заданием**

**Иметь практический опыт:**

* Устанавливать и настраивать веб-серверы, СУБД для организации работы веб-приложений.
* Использовать инструментальные средства контроля версий и баз данных.
* Проводить работы по резервному копированию веб-приложений.
* Выполнять регистрацию и обработку запросов Заказчика в службе технической поддержки.

**Уметь:**

* Подключать и настраивать системы мониторинга работы Веб-приложений и сбора статистики его использования.
* Устанавливать и настраивать вебсервера, СУБД для организации работы веб-приложений.
* Работать с системами Helpdesk.
* Выяснять из беседы с заказчиком и понимать причины возникших аварийных ситуаций с информационным ресурсом.
* Анализировать и решать типовые запросы заказчиков.
* Выполнять регламентные процедуры по резервированию данных.
* Устанавливать прикладное программное обеспечение для резервирования веб-приложений

**Знать:**

* Основные показатели использования Веб-приложений и способы их анализа.
* Регламенты работ по резервному копированию и развертыванию резервной копий веб-приложений.
* Способы и средства мониторинга работы веб-приложений.
* Методы развертывания веб-служб и серверов.
* Принципы организации работы службы технической поддержки.
* Общие основы решения практических задач по созданию резервных копий.

**ПК 9.5. Производить тестирование разработанного веб приложения**

**Иметь практический опыт:**

* Использования инструментальные средства контроля версий и баз данных, учета дефектов.
* Тестирования вебприложения с точки зрения логической целостности.
* Тестирования интеграции веб-приложения с внешними сервисами и учетными системами.

**Уметь:**

* Выполнять отладку и тестирование программного кода (в том числе с использованием инструментальных средств).
* Выполнять оптимизацию и рефакторинг программного кода.
* Кодировать на скриптовых языках программирования;
* Тестировать веб-приложения с использованием тест-планов.
* Применять инструменты подготовки тестовых данных.
* Выбирать и комбинировать техники тестирования вебприложений.
* Работать с системами контроля версий в соответствии с регламентом использования системы контроля версий.
* Выполнять проверку веб-приложения по техническому заданию.

**Знать:**

* Сетевые протоколы и основы webтехнологий.
* Современные методики тестирования эргономики пользовательских интерфейсов.
* Основные принципы отладки и тестирования программных продуктов.
* Методы организации работы при проведении процедур тестирования.
* Возможности используемой системы контроля версий и вспомогательных инструментальных программных средств для обработки исходного текста программного кода.
* Регламент использования системы контроля версий.
* Предметную область проекта для составления тест-планов.

**ПК 9.6. Размещать веб приложения в сети в соответствии с техническим заданием**

**Иметь практический опыт:**

* Публикации веб-приложения на базе хостинга в сети Интернет.

**Уметь:**

* Выбирать хостинг в соответствии с параметрами веб-приложения.
* Составлять сравнительную характеристику хостингов

**Знать:**

* Характеристики, типы и виды хостингов.
* Методы и способы передачи информации в сети Интернет.

Устройство и работу хостинг-систем.

**ПК 9.7. Осуществлять сбор статистической информации о работе веб-приложений для анализа эффективности его работы.**

***Иметь практический опыт:***

* Реализации мероприятий по продвижению веб-приложений в сети Интернет.
* Сбора и предварительного анализа статистической информации о работе веб-приложений

***Уметь:***

* Подключать и настраивать системы мониторинга работы Веб-приложений и сбора статистики его использования.
* Составлять отчет по основным показателям использования Веб-приложений (рейтинг, источники и поведение пользователей, конверсия и др.).

**Знать:**

* Основные показатели использования Веб-приложений и способы их анализа.
* Виды и методы расчета индексов цитируемости Веб-приложений (ТИЦ, ВИЦ).

**ПК 9.8. Осуществлять аудит безопасности веб-приложения в соответствии с регламентами по безопасности**

**Иметь практический опыт:**

* Обеспечения безопасной и бесперебойной работы

**Уметь:**

* Осуществлять аудит безопасности веб-приложений.
* Модифицировать веб-приложение с целью внедрения программного кода по обеспечению безопасности его работы

**Знать:**

* Источники угроз информационной безопасности и меры по их предотвращению
* Регламенты и методы разработки безопасных веб-приложений.

**ПК 9.9. Модернизировать веб-приложение с учетом правил и норм подготовки информации для поисковых систем**

**Иметь практический опыт:**

* Модернизации веб-приложения для обеспечения доступа к ним поисковых систем.

**Уметь:**

* Модифицировать код веб-приложения в соответствии с требованиями и регламентами поисковых систем.
* Размещать текстовую и графическую информацию на страницах веб-приложения.
* Редактировать HTML-код с использованием систем администрирования.
* Проверять HTMLкод на соответствие отраслевым стандартам.

**Знать:**

* Особенности работы систем управления сайтами.
* Принципы функционирования поисковых сервисов и особенности оптимизации Веб-приложений под них (SEO).
* Методы оптимизации веб-приложений под социальные медиа (SMO).

**ПК 9.10. Реализовывать мероприятия по продвижению веб-приложений в сети Интернет**

**Иметь практический опыт:**

* Реализации мероприятий по продвижению веб-приложений в сети Интернет.
* Сбора и предварительного анализа статистической информации о работе веб-приложений.

**Уметь:**

* Подключать и настраивать системы мониторинга работы Веб-приложений и сбора статистики его использования.
* Работать с системами продвижения вебприложений.
* Публиковать информации о вебприложении в специальных справочниках и каталогах.
* Осуществлять подбор и анализ ключевых слов и фраз для соответствующей предметной области с использованием специализированных программных средств.
* Составлять тексты, включающие ссылки на продвигаемый сайт, для размещения на сайтах партнеров
* Осуществлять оптимизацию веб-приложения с целью повышения его рейтинга в сети интернет.

**Знать:**

* Принципы функционирования поисковых сервисов.
* Виды и методы расчета индексов цитируемости вебприложений (ТИЦ, ВИЦ).
* Стратегии продвижения вебприложений в сети Интернет.
* Виды поисковых запросов пользователей в интернете.
* Программные средства и платформы для подбора ключевых словосочетаний, отражающих специфику сайта.
* Инструменты сбора и анализа поисковых запросов.

**Создание проекта «Интернет магазин»**

**Цель:** Создать «Интернет магазин», по критериям.

**Теоретическое обоснование**

Через сеть Интернет покупатель при помощи браузера заходит на web-сайт интернет-магазина. Web-сайт содержит электронную витрину, на которой представлены каталог товаров (с возможностью поиска) и необходимые интерфейсные элементы для ввода регистрационной информации, формирования заказа, проведения платежей через интернет, оформления доставки, получения информации о компании-продавце и online помощи.

Регистрация покупателя производится либо при оформлении заказа, либо при входе в магазин. После выбора товара от покупателя требуется заполнить форму, в которой указывается, каким образом будет осуществлена оплата и доставка. Для защиты персональной информации взаимодействие должно осуществляться по защищенному каналу. По завершению формирования заказа и регистрации вся собранная информация о покупателе поступает из электронной витрины в торговую систему интернет-магазина. В торговой системе осуществляется проверка наличия затребованного товара на складе, инициируется запрос к платежной системе. При отсутствии товара на складе направляется запрос поставщику, а покупателю сообщается о времени задержки.

В том случае, если оплата осуществляется при передаче товара покупателю: курьером или наложенным платежом - необходимо подтверждение факта заказа. Чаще всего это происходит посредством электронной почты или по телефону.

При возможности оплаты через интернет, подключается платежная система (PayPal, EasyPay, WebMoney и т.п.). После сообщения о проведении online платежа торговой системой формируется заказ для службы доставки.

Торговые системы электронных магазинов на практике редко бывают полностью автоматизированными. Полностью автоматизированными они бывают лишь в тех случаях, когда интернет-магазин сделан профессионально и когда в нём отсутствуют какие-либо ошибки. Лёгкость совершения покупки зачастую приводит к увеличению количества ошибок пользователя (особенно при низкокачественном дизайне электронного магазина и отсутствии online помощи) - а это существенные потери для магазина. Поэтому требуется проверка менеджером каждого отдельного факта заказа. Исключение составляет лишь продажа информационного продукта (файлы, архивы и т.п.), который можно доставить с минимальными издержками непосредственно через интернет.

Можно утверждать, что основные проблемы электронной коммерции лежат на стыках интернета и реальной деятельности. В обычной торговле покупатель привык к тому, что есть возможность оценить товар визуально, определить его качество и характеристики. В электронной торговле он такой возможности лишён. Максимум, на что он может рассчитывать, это фотография товара и перечисление его характеристик. Зачастую этой информации достаточно, но здесь вступают в действие эмоциональные и психологические факторы. Большинство электронных магазинов имеют проблемы с доставкой товаров, особенно если цена товара невелика. Проблемы также возникают при необходимости оплатить товар в электронном магазине. Тому есть множество причин: недоверие граждан по отношению к банковской системе в целом и безналичным платежам в частности, неурегулированность организационных и правовых вопросов электронных платежей, неуверенность в безопасности проведения транзакций через интернет.

В общем случае техническую сторону любого интернет-магазина можно рассматривать как совокупность электронной витрины и торговой системы.

Электронная витрина предназначена для выполнения следующих задач:

\* предоставление интерфейса к базе данных продаваемых товаров в виде каталога либо прайс-листа;

\* работа с электронной "корзиной" покупателя;

\* регистрация покупателей;

\* оформление заказов с выбором метода оплаты и доставки;

\* предоставление online помощи покупателю;

\* сбор маркетинговой информации;

\* обеспечение безопасности личной информации покупателей;

\* автоматическая передача информации в торговую систему.

Витрина электронного магазина располагается на интернет-сервере и представляет собой Web-сайт с активным содержанием.

Основа витрины электронного магазина - каталог товаров с указанием цен, который может быть структурирован различными способами (по категориям товаров, по производителям), содержать полную информацию о характеристиках каждого товара, а также его изображение. Выбрав понравившийся товар, пользователь помещает его в "корзину". В любой момент до окончательного оформления заказа покупатель может отредактировать содержимое корзины и количество товаров каждого вида.

Процесс регистрации может инициироваться системой до или после выбора товаров из каталога. Ввод регистрационных данных после выбора товаров позволяет покупателю сэкономить время в том случае, если он не принял решения что-либо купить в этом электронном магазине.

Зачастую электронная витрина и является собственно интернет-магази-ном, а вторая важная часть, электронная торговая система, просто отсутствует. Все запросы покупателей поступают не в автоматизированную систему обработки заказов, а к менеджерам по продажам. Далее бизнес-процессы электронного магазина полностью повторяют бизнес-процессы предприятия розничной торговли. Таким образом, интернет-витрина является инструментом привлечения покупателя, интерфейсом для взаимодействия с ним и проведения маркетинговых мероприятий.

Автоматизация торговли становится выгодной только с ростом масштабов бизнеса. До тех пор, пока несколько человек без особых усилий справляются с ручной обработкой заказов покупателей, особенно если цена каждого отдельного заказа относительно низка или количество покупателей невелико, проще всего организовать интернет-магазин именно таким образом, как это будет показано в данной дипломной работе. Но для фирм, проводящих сотни транзакций в день и ориентированных на бизнес в интернете это решение не подходит.

**Практическая часть:**

**Задание 1**

* Определение названия проекта
* «Создать документации к проекту по созданию и внедрению интернет магазина»

**Задание 2**

1. Определение целей проекта
2. Определение критериев достижения целей проекта

*Пример таблицы 2.1. Критерии достижения целей проекта*

|  |  |  |
| --- | --- | --- |
| № | **Цель** | **Критерии** |
| 1 | Вывод на окупаемость не менее, чем через год | Получение прибыли не раньше через 12 месяцев после запуска проекта |
| 2 | Достижения уровня продаж, обеспечивающих рентабельность | Достижение рентабельности не менее 15% |

1. **Определение ограничений проекта**

Инвестиции в размере 450 000 руб.

Срок реализации проекта 2 месяца

1. **Определение допущений проекта**

Браузеры Internet Explorer и Opera, Yandex, Google

Рост платежеспособности спроса

Покупатели предпочитают различные виды доставки и оплаты

**Задание № 3.**

**Определение конечного результата проекта**

* Корректно-функционирующий интернет магазин, который:
* Оптимизирован для поисковых машин Yandex, Google

**Оценка необходимости разбиения проекта на подпроекты**

**Конечный результат проекта**

Создание интернет магазина

**Высокоуровневые требования к конечному результату проекта**

|  |  |
| --- | --- |
| № | Наименование требования |
| 1 | Дизайн: визуальная платформа, создание структуры сайта,Создание логотипа интернет магазина |
| 2 | Программная часть: разработка, настройка CMS, разработка сервисов, установка оригинальных шаблонов |
| 3 | Наполнение: главная страница, каталог, новинки, контакты, о нас, карта сайта, разновидности товара, для корпоративных клиентов |
| 4 | Вывод на рынок интернет магазина: конкурентоспособность на рынке |

1. **Определение фаз проекта**

|  |  |
| --- | --- |
| № | **Наименование фазы проекта** |
| 1 | Инициация |
| 2 | Создание шаблонов и дизайна интернет магазина |
| 3 | Адаптация системы управления сайтом |
| 4 | Наполнение и запуск сайта |

**Разработка аркадной игры с применением сценария JavaScript**

**Цель:** разработать аркадную игру с помощью сценария JavaScript/

**Теоретическая часть:**

В последние годы Интернет стал неотъемлемой частью нашей повседневной жизни. Использование игр приобрело заметную роль в этой вселенной, позволяя миллионам людей получать доступ к развлечениям быстро и бесплатно.

Наш заказчик сделал срочный заказ: необходимо разработать игру и воплотить желание заказчика в реальность! Все что вам необходимо – это реализовать логику игры.

Название игры: Рыцарь.

Технологии этого модуля: HTML 5, CSS3, JavaScript, jQuery, Граф. дизайн, PHP

Вам необходимо реализовать функционал игры. Готовый шаблон и все необходимые файлы предоставлены. Использование шаблона обязательно.

## Практическое задание:

В игре используются элементы, описанные ниже:

1. Рыцарь: элемент, который контролируется игроком.
2. Монстры: элементы, которые необходимо уничтожить игроку.
3. Шкала жизней (HP): шкала, отражающая запас жизненной энергии игрока.
4. Шкала энергии (MP): шкала, отражающая запас магической энергии игрока.
5. Имя игрока: имя игрока, которое он ввел на стартовом экране.
6. Убийства: количество убитых монстров.
7. Панель умений: умения, которые может использовать игрок, для сражения с монстрами.

Игра должна начинаться со стартового экрана с инструкцией к игре, полем для ввода имени игрока и кнопкой "Начать", если поле имени пустое, то кнопка не активна. Инструкция к игре должна быть представлена анимировано.

Игровой функционал:

1. По нажатию на кнопку "Начать игру" игрок попадает на экран игры. Изначально у игрока 0 очков, 100 HP и 100 MP, таймер 00:00 (в формате mm:ss).
2. В начале игры запускается таймер, персонаж располагается в левой стороне игрового поля, что является начальной границей карты.
3. Игрок может передвигаться с помощью клавиш-стрелок: налево и направо. Когда игрок находиться в левой половине видимой части экрана фон не передвигается, а «замирает». Когда игрок доходит центральной точки он остается в центральной части экрана, происходит анимация его передвижения, а фон в свою очередь начинает «прокручиваться». При этом когда фон достигает конца, игрок должен передвигаться в правую часть экрана.
4. По мере прохождения игры на игрока нападают монстры, которые случайно генерируются в видимой части игрового поля и движутся справа налево. Игра заканчивается, как только игрок доходит до конца карты и достигает правого местоположения или происходит смерть рыцаря, а таймер останавливается.
5. За каждого убитого монстра дается 1 очко.
6. Если жизни игрока кончаются (0HP), то игра заканчивается.
7. Жизни игрока (HP) регенерируются со скоростью 2HP/с.
8. Энергия игрока (MP) регенерируется со скоростью 5MP/с.
9. Игрок может использовать умения. Каждое умение тратит какое-то количество MP и имеет время перезарядки. Вот список умений:

|  |  |
| --- | --- |
| **nc-game-media/layout/img/skill-swordУдар мечом.**Описание: Игрок делает удар мечом, нанося урон монстрам перед собой.Клавиша активации: 1.Время перезарядки: 0 сек.Количество потребляемой MP: 0.Урон: 15 ед. | **Блок.**Описание: Игрок выставляет щит перед собой. Щит блокирует весь урон. Монстры не могут пройти за щит.Клавиша активации: 2.Время перезарядки: 0 сек.Количество потребляемой MP: 5. |
| **nc-game-media/layout/img/skill-sword-3.pngТрио мечей.**Описание: Игрок выпускает три меча перед собой. Мечи летят, нанося 40 единиц урона всем монстрам на своем пути.Клавиша активации: 3.Время перезарядки: 3 сек.Количество потребляемой MP: 10.Урон: 40 ед. | **nc-game-media/layout/img/skill-sword-8.pngГрад мечей.**Описание: в радиусе нескольких метров от игрока на землю обрушивается град мечей уничтожая всех, кто был в данном радиусе.Клавиша активации: 4.Время перезарядки: 15 сек.Количество потребляемой MP: 30.Урон: 100 ед. |

1. При нажатии на кнопку ESC игра ставится на паузу, а при повторном нажатии - снимается с паузы. Во время паузы останавливаются все интерактивные действия, а также вся анимация, таймер тоже замирает, умениями пользоваться нельзя, игрок и монстры не двигаются.
2. При ударе мечом урон наносится один раз каждому монстру, который находится в зоне поражения мечом, т.е. если перед игроком находится один монстр, то он получит 15 ед. урока, а если 3 монстра, то каждый из них получит по 15 ед. урона.
3. Монстры должны следовать за игроком. Если игрок пробежал монстров, то они должны развернуться и следовать за ним.
4. Когда игрок идет обратно, он поворачивается в обратную сторону.
5. Каждый из монстров имеет свою скорость атаки и скорость движения, которая не превышает скорость движения игрока. На экране не должно находиться более 10 монстров. Вот описания монстров:

|  |  |  |
| --- | --- | --- |
| nc-game-media/enemies/dog/sprites/run/1.pngПёсНаносит 2ед уронаИмеет 15HP | nc-game-media/enemies/orc3/sprites/Attack2/Attack2_000.pngЭльфНаносит 5ед уронаИмеет 30HP | nc-game-media/enemies/orc1/sprites/Attack2/Attack2_000.pngГринчНаносит 10ед уронаИмеет 60HP |

1. Ваша игра должна работать без отображения JavaScript ошибок или сообщений в консоли браузера.
2. Ваш HTML/CSS и JavaScript код должен быть организован и понятным. Используйте корректные наименования переменных, методов и не забывайте оставлять комментарии для дальнейшей поддержки в будущем.
3. После завершения игры результаты должны быть сохранены на сервере и показана таблица с рейтингом.
4. Вам дан PHP файл, который сохраняет данные в базу данных и возвращает массив с результатами. Используйте этот файл для сохранения результатов и получения данных для рейтинга. В этом файле можно изменять только данные доступа к базе данных (логин, пароль, имя базы данных). Этот файл принимает следующий AJAX запрос:
* метод: POST
* username - имя пользователя
* score - количество убитых монстров
* time – время игры

В ответ возвращается массив с данными:

[ {"id":"1","username":"Player 1","score":"10","time":"20"},
{"id":"2","username":"Player 2","score":"8","time":"30"}]

1. После завершения игры открывается экран результатов, в котором формируется таблица с лучшими 10 игроками, если игрок не вошел в таблицу, то выводятся 9 лучших игроков, а в последней строке указывается место и результат игрока только что завершившего игру.
2. Данные в таблице должны быть отсортированы по количеству убитых монстров по убыванию. Если несколько строк имеют одинаковое кол-во убитых монстров, то они сортируются между собой по времени по возрастанию. Если несколько строк имеют одинаковое количество убитых монстров и одинаковое время, то они занимают одинаковую позицию в рейтинге.
3. На экране результатов должна быть кнопка «Играть сначала», которая позволяет начать игру с начала, первый экран не отображается.

Проявите все свои навыки для создания качественной игры, с анимацией и интерактивностью.

**СИСТЕМА ОЦЕНКИ**

|  |  |  |  |
| --- | --- | --- | --- |
| № | критерий | сумма |  |
|  | Организация работы и управление | 1.85 |  |
|  | Коммуникация и навыки межличностного общения | 2.00 |  |
|  | Дизайн | 5.00 |  |
|  | Верстка | 6.00 |  |
|  | Программирование на стороне клиента | 10.00 |  |
|  | Программирование на стороне сервера | 0.15 |  |
|  | CMS | 0.00 |  |
| **Всего** |  | **25.00** |  |

**Оптимизировать работу сайта**

**Цель работы*:*** Приобрести навыки поисковой оптимизации Web-ресур-

сов, провести внутреннюю SEO-оптимизацию страницы Web-ресурса.

# Теоретические сведения.

## Разработка красивого и функционального сайта зачастую не гаран- тирует его успех у потенциальных посетителей. Для их привлечения необходимо проделать объемную работу по продвижению Web-ресурса. На сегодня существует несколько методик продвижения сайта:

1. Контекстная реклама – размещение информации о ресурсе на специализированных рекламных площадках или в сервисах, демонстри- рующих ее целевой аудитории. Популярными площадками здесь явля- ются Яндекс.Директ, Google Adwords, Бегун.
2. Баннерная реклама – размещение мультимедийного рекламного блока (баннера), содержащего гиперссылку на продвигаемый ресурс, на сто- ронних тематически близких целевой аудитории продвигаемого ресурса сай- тах.
3. Вирусный маркетинг – распространение информации о ресурсе са- мими пользователями (посетителями) сайта. Основой продвижения здесь ста- новится информационный «вирус» – яркая, броская, необычная подача инфор- мации о продвигаемом ресурсе в виде необычного видеоролика или ориги- нально поданной новости, которыми пользователи начинают делиться друг с другом, тем самым быстро и бесплатно распространяя ссылки на ресурс в ре- жиме «сарафанного радио»
4. Использование социальных сетей позволяет быстро и бесплатно заявить о ресурсе на большую аудиторию, объединенную дружественными от- ношениями в популярных социальных сетях. Помимо непосредственного эф- фекта волнового распространения информации о ресурсе в виде всевозмож- ных «лайков», данный метод продвижения обладает потенциалом обратной связи, позволяющим узнать мнение пользователей относительно качества про- двигаемого ресурса.
5. Публикация в популярных блогах и на форумах также позволяет быстро заявить о продукте максимально широкой аудитории с учетом ее тема- тических предпочтений. Блоггер может посвятить ресурсу отдельную статью, оставить ссылку на ресурс в посте. Обсуждения в форуме позволяют оставлять скрытую рекламу в обсуждениях. Наличие собственного блога позволяет бо- лее целенаправленно вести рекламную кампанию ресурса, а также получать от читателей (посетителей) обратную связь.
6. Публикация новостей или пресс-релизов о продвигаемом ресурсе в электронных средствах массовой информации или на тематических порталах позволяют создать дополнительный информационный повод, адресованный заинтересованной (или интересующей владельцев ресурса) аудитории.
7. Регистрация ресурса в электронных каталогах позволяет увели- чить количество точек потенциального соприкосновения заинтересованных в конкретном типе услуг или товаров пользователей и продвигаемого ресурса.
8. SEO-продвижение ресурса позволяет заявить о ресурсе через ме- ханизм поиска популярных поисковых сервисов. Получение информации с ис- пользованием поисковых движков стал привычным и рутинным действием для подавляющего большинства пользователей сети Интернет, в связи с чем высокие позиции в выдаче по ключевым тематическим запросам в Google или Яndex стало приоритетом для владельцев Web-ресурсов. Правильно выбран- ная стратегия SEO-продвижения и умелое использование его инструмента- рием и методиками позволяет быстро, дешево и эффективно донести инфор- мацию о продвигаемом ресурсе желаемой аудитории, при этом эта информа- ция зачастую выглядит для нее как наиболее авторитетная и непредвзятая. Это делает SEO одним из самых часто используемых инструментов раскрутки Web-ресурса, а услуги SEO-оптимизаторов весьма востребованными на рынке, в связи с чем в данной лабораторной работе основное внимание будет уделено именно этому средству продвижения сайтов в сети Интернет.

## SEO (Search Engine Optimization) – процесс оптимизации Web- ресурса под поисковые системы для увеличения количества трафика пользователей на web-ресурс путем увеличения его ранга в результатах поиска в поисковых системах. Эмпирически понятно и статистически до- казано, что высокие позиции сайта в результатах выдачи поисковой си- стемы повышают вероятность того, что пользователь посетит данный сайт. Поисковая оптимизация предполагает изменение его внутренних факторов, влияющими на ранжирование в поисковых системах (струк- туры ресурса, качество контента, код HTML), а также его внешних фак- торов ранжирования (наличие внешних ссылок на сайт, качество этих ссылок, наличие сайта в тематических каталогах). Цель оптимизации – увеличение релевантности ресурса целевым ключевым словам, отражаю- щим содержимое ресурса, повышение ранга сайта по алгоритмам ранжи- рования поисковых машин, увеличение позиций в результатах поисковой выдачи поисковых машин для привлечения большего количества посети- телей на сайт.

SEO-оптимизация популярна и эффективна благодаря:

* высокому доверию к выдаче популярных поисковых машин со стороны посетителей;
* оптимальному отношению «цена/качество» для процесса продвиже-

ния;

* долговременному эффекту от процесса продвижения;
* универсальности процесса продвижения для различных поисковых

движков – несмотря на уникальные алгоритмы индексирования и ранжирова- ния у разных поисковых машин, мероприятия по повышения позиции сайта в одной из них приводят к росту рейтинга и в других.

Есть у SEO-продвижения и проблемные места:

* Процесс продвижения носит вероятностный характер, нельзя га- рантировать 100%-й результат, например, попадания на первое место в выдаче по определенным запросам;
* Высокая инерционность работы поисковых систем приводит к временным задержкам при ожидании итогов работы.

## Прежде чем проводить непосредственно SEO-оптимизацию необ- ходимо понять принцип работы поисковых машин. Это позволит прово- дить работу осмысленно и направленно на положительный результат.

Для проведения состава работ по поисковой оптимизации сайта необхо- димо иметь представление о принципах работы поисковых систем. Современ- ная поисковая система является сложным программным комплексом, включа- ющим среди прочих:

**- Spider (**паук) - компонент, скачивающий страницы браузера с использо- ванием HTTP-запросов, как это делает браузер коечного пользователя. Од- нако, если целью браузера является визуальное представление страницы поль- зователю, то паук передает скачанную страницу на обработку другим компо- нентам системы (Crawler-паукам или индексаторам);

* **Crawler (**путешествующий паук**) –** компонент, обеспечивающий поиск новых страниц для скачивания и индексации. Он выделяет все ссылки, при- сутствующие на скачанной пауком-spider-ом странице. После этого строится план дальнейшего сканирования по найденным ссылкам. Таким образом поис- ковая система постоянно обновляет и пополняет список известных ей страниц.
* **Indexer (**индексатор) получает страницу от паука, парсит (разбирает) ее на составные части: текст, изображения, видео, ссылки, заголовки, специаль- ные теги и др. Для анализа индексатор использует оригинальные лексические и морфологические алгоритмы.
* **Database** (база данных) или индекс системы выступает как хранилище проиндексированных данных. В ней ставится соответствие элемента (слова, изображения, видеофрагмента) и Web-ресурсов, на которых он был найден па- уком. Данные в хранилище заносятся индексатором.
* **Search Engine Results Engine** (система выдачи результатов) работает на конечном этапе поискового цикла. Его назначение – ранжированием страниц по запросу пользователя. Именно этот компонент формирует поисковую вы- дачу – тот набор ссылок на Web-ресурсы, которые согласно алгоритмам этой конкретной поисковой системы в наибольшей степени соответствуют обраще- нию пользователя. Для отбора ссылок и их сортировки перед отправкой поль- зователю используются алгоритмы ранжирования (например, Google Pag- eRank или Яндекс ТИЦ). Обобщенная схема работы поисковой машины пред- ставлена на рис.1.

Пауки-Crawler’ы по указанию пауков-Spider’ов скачивают новые неизвестные страницы с сайта, они анализируются и индексируются, найденные на них ссылки служат путями для нового цикла поиска: команда Crawler’у инициирует поиск и скачивание, парсинг выделяет содержимое страниц и порожлдает новые пути поиска контента и т.д. В найденных пауками страницах анализи- руется содержание и они встраиваются в индекс поисковой системы, храня- щийся в общей базе.

Необходимо отметить, что поисковые боты не только ищут новые, ранее неизвестные ресурсы, но и периодически переиндексируют ранее посещенные страницы. Таким образом достигается актуальное состояние базы данных по- исковой системы.

Интернет

Поисковая система

Индексатор

Паук

База данных поисковой си- стемы

Путешествующий

паук

Поисковый запрос

Система выдачи результатов

Поисковая выдача

Рисунок 1 – Обобщенный алгоритм работы поисковой системы

С учетом того, что оптимизация поисковой выдачи (SEO) ориентируется именно на результаты работы системы выдачи поисковой машины, именно факторы ранжирования в первую очередь учитываются SEO-специалистом в его работе. Однако, нельзя не учитывать особенности работы поисковых ро- ботов-пауков при оптимизации Web-ресурса. Это позволит как снизить нагрузку на Web-сервер при обработке запросов от пауков, так и актуально обновлять индекс поисковой машины при обновлении данных на страницах сайта или появлении новых страниц на ресурсе.

## Разработчик (владелец) Web-ресурса может управлять процессом индексирования своего сайта со стороны поисковых роботов. Для этого используются:

* + метатеги заголовка HTML-документа;
	+ файл robots.txt.

Файл robots.txt представляет собой текстовый файл в кодировке UTF-8, который размещается в корневой директории сайта и может со- держать ряд инструкций для спайдеров поисковых систем по порядку ин- дексирования данного ресурса. Этот файл актуален для протоколов FTP, HTTP, HTTPS. Назначение и внутренняя структура файла задекларирована в документе под названием «Стандарт исключений для роботов, принятым кон- сорциумом W3C в 1994 года и использующимся большинством известных по- исковых машин.

Сессия работы поискового робота с сайтом начинается с загрузки файла robots.txt. Если файл HTTP-статус загрузки этого файла будет иметь ста- тус, отличный от 200 OK, робот будет считать, что доступ к документам ничем не ограничен. Иначе он прочитает строки конфигурационного файла и при ин- дексировании будет следовать указанным в них инструкциям. Необходимо по- нимать, что инструкции robots.txt носят рекомендательный характер, поиско- вые машины могут их игнорировать. Надежный способ защитить страницы и разделы сайта от индексирования поисковыми системами – поместить их в за- шифрованные разделы сайта.

Внутренняя структура robots.txt предельно проста: он должен содер- жать одну или несколько записей (records), разделенных одной или несколь- кими пустыми строками. Каждая запись должна содержать строки в форме:

<field>:<optional\_space><value><optional\_space>

Запись должна начинается с одной или нескольких строк User-Agent, за кото- рой следуют одна или несколько строк с директивами Allow или Disallow

Директива User-Agent указывает, к какому роботу обращены последую-

щие директивы. При том допускается сформулировать разные правила для различных роботов. Ниже приведен пример использования директив User- Agent для различных поисковых роботов:

# Директивы для всех роботов

User-agent: \*

#Обратите внимание на пустую строку после каждого блока User-Agent

# Директивы для всех роботов Яндекса

User-agent: Yandex

# Директивы только основному индексирующему роботу Яндекса

User-agent: YandexBot

# Не забываем про роботов Google User-agent: Googlebot

Ниже приведен список некоторых популярных роботов, на которые мо- жет ориентирован файл robots.txt для конфигурирования процесса индексиро- вания:

* **YandexBot** — основной поисковый робот Yandex
* **YandexBlogs** — робот поиск по блогам, индексирующий посты и ком- ментарии;
* **YandexMarket**— робот Яндекс.Маркета;
* **YandexNews** — робот Яндекс.Новостей;
* **YandexDirect** — скачивает информацию о контенте сайтов-партнеров Рекламной сети, чтобы уточнить их тематику для подбора релевантной рекламы;
* **YandexImages** — индексатор Яндекс.Картинок;
* **YandexCalendar** — робот Яндекс.Календаря.
* **YandexMetrika** — робот Яндекс.Метрики;
* **YandexMedia** — робот, индексирующий мультимедийные данные;
* **Googlebot-News** — робот Google для поиска новостей;
* **Mediapartners-Google** — робот Google для сервиса AdSense;
* **Googlebot-Image** — робот Google для картинок;
* **Googlebot-Video** — робот Google для видео;
* **Googlebot-Mobile** — робот Google для мобильных версий ресурсов;

Директива User-Agent только конкретизирует тип робота, для которого заданы следующие директивы в блоке. Без дополнительных директив его ис- пользование бессмысленно. Основными подобными директивами являются, как уже было упомянуто выше, Allow и Disallow.

**Disallow** —директива**,** запрещающая индексацию сайта или его части. Что именно запрещается сканировать/индексировать, уточняется в парамет- рах директивы.

Вот как, например, можно запретить в robots.txt индексацию сайта кон- кретным роботом:

User-agent: Yandex-News Disallow: /

Можно закрыть от робота (-ов) не весь сайт, а лишь конкретные раз- делы. Закрываем от всех роботов раздел docs сайта:

User-agent: \* Disallow: /docs

Для параметра директивы Disallow допустимо использовать символы- шаблонизаторы \* и ?. Например,

User-agent: \* Disallow: /page\*

запретит сканирование всех , начинающихся с page (/page\_old, /page-about,

/page/news). Знак $ указывает на необходимость точного соответствия:

User-agent: \* Disallow: /page$

скроет только папку /page, но откроет доступ к /page1, /page/news и др.

Если параметр директивы Disallow будет опущен, то это означает отсутствие запрещения.

Директива **Allow** играет роль альтернативы Disallow — она разрешает сканирование тех или иных разделов сайта и имеет синтаксис, аналогичный Disallow. Совместное использование двух рассмотренных директив позволяет комбинировать права роботов на доступ к тем или иным разделам:

User-agent: Googlebot Disallow: /

Allow: /docs

В приведенном примере мы запрещаем сканирование сайта за исключе- нием раздела /docs.

Помимо уже рассмотренных, в файле robots.txt можно использовать еще ряд директив. Одна из них – директива **Host**. Она служит для указания роботу Яндекса (и поддерживается на сегодняшний день только этой поисковой си- стемой) главного зеркала Вашего сайта**..** Директива Host полезна в том случае, если ваш сайт доступен по нескольким адресам и вам необходимо установить приоритет для одного из них, чтобы именно он выдавался как результат поис- ковой выдачи. Например, для двух доменов mysite.ru и [www.mysite.ru](http://www.mysite.ru/) необхо- димо указать второй как основной. Тогда в файле robots.txt указываем:

User-agent: Yandex Disallow: /page Host: [www.mysite.ru](http://www.mysite.ru/)

Еще одна интересная и полезная директива – **Sitemap**. C ее помощью в robots.txt можно указать расположение на сайте файла карты сайта sitemap.xml. Указание адреса карты сайта через директиву Sitemap в robots.txt позволяет поисковому роботу узнать о наличии карты сайта и начать ее индексацию. Sitemaps — это XML-файл с информацией для поисковых си- стем о тех страницах веб-сайта, которые подлежат индексации. Sitemaps не только указывают поисковика те папки, которые подлежат индексированию, но и хранит информацию о времени их последнего обновления, планируемой частоте обновления и приоритете относительно других страниц сайта. Вся эта информация полезна поисковому роботу для повышения эффективности об- работки страниц в разделах Web-ресурса. Вот как можно задать местоположе- ния карты сайта:

User-agent: \* Disallow: /page

Sitemap: <http://www.some.com/sitemap.xml>

Внутренняя структура Sitemaps документирована упрощенный вариант подобного файла выглядит следующим образом:

<?xml version="1.0" encoding="UTF-8"?>

<urlset [xmlns="http://www.sitemaps.org/schemas/sitemap/0.9](http://www.sitemaps.org/schemas/sitemap/0.9)">

<url>

[<loc>http://www.some.com/](http://www.some.com/)</loc>

*<lastmod>2015-01-01</lastmod>*

*<changefreq>monthly</changefreq>*

*<priority>0.8</priority>*

</url>

</urlset>

Файл robots.txt – не единственная возможность управления процессом индексацию страницы. Для этих же целей можно воспользоваться специали- зированными html-тегами. В заголовке html-документа можно поместить теги meta со следующими атрибутами:

<meta name= “robots” content= “noindex”/> — не индексировать со- держимое страницы;

<meta name= “robots” content= “nofollow”/> — не переходить по ссылкам на странице;

<meta name= “robots” content= “none”/> — запрещено индексиро- вать содержимое и переходить по ссылкам на странице;

<meta name= “robots” content= “noindex, nofollow”/> — анало- гично content= “none”.

<meta name="Revisit-After" content="1 Days"> - приглашаем поиско- вого робота заходить на сайт с частотой 1 раз в день, поскольку именно с этой периодичностью на нем обновляется контент.

Описанные инструменты позволяют конфигурировать процесс индекса- ции поисковыми роботами содержимого сайта. Однако, главной целью SEO- оптимизации является продвижение страниц сайта на этапе ранжирования по запросу, когда поисковая система формирует список выдачи на сформулиро- ванный пользователем запрос. По результатам исследований [3] до 80% поль- зователей поисковых систем всегда или часто переходят по ссылкам, выдава- емым поисковой системой как наиболее подходящие поисковому запросу. Ал- горитмы ранжирования современных поисковых систем являются сложными многокритериальными, динамически изменяющимися моделями, учитываю- щими сотни статистических показателей работы ресурса. Разработчики поис- ковых систем должны отслеживать частоту обновления информации на ре- сурсе, его популярность у посетителей, авторитетность сайта у других ресур- сов, попытки владельцев ресурсов обманным путем искусственно повысить ранг своего сайта. При этом базовые принципы ранжирования web-ресурсов при поисковой выдаче известны, они опираются на ряд факторов ранжирова- ния. В связи с этим необходимо знать эти факторы и учитывать их как в про- цессе разработки web-сайта, так и при его продвижении. Общее количество факторов весьма велико (более 700 у Яндекс), поэтому рассмотреть их все не представляется возможным, заострим внимание на некоторых из них. Фак- торы ранжирования делятся на группы:

1. **Внутренние факторы.**

В эту группу входят факторы, которые определяют разработчики/вла- дельцы ресурса путем его внутреннего структурирования и наполнения. Ос- новными здесь можно назвать:

- **качество контента сайта**, определяемого по его актуальности, ориги- нальности, релевантности, «тошностности». Некоторые из этих характеристик не требуют особых пояснений – если информация на ресурсе актуальна, регу- лярно обновляется, не является копией контента с других сайтов, то такой ре- сурс будет иметь больший вес по алгоритмам ранжирования, привлекать вни- мание как посетителей, так и владельцев других ресурсов и, в конечном итоге, этот ресурс будет расти в позициях поисковой выдачи. Другие характеристики вступают во взаимный конфликт. Так, например, показатель релевантности учитывает процент вхождения слов поискового запроса в контенте сайта: чем он выше, тем более релевантны страницы сайта запросу. Однако, поисковики вводят термин «тошнотности» страниц сайта, которая определяется как пере- груженность текста целевыми ключевыми словами. Зачастую SEO- оптимизаторы перенасыщают страницы сайта, которые хотят поднять в поис- ковой выдаче, ключевыми словами тематике, которой посвящен ресурс. В этом случае читать человеку подобный текст становится неинтересно, от оби- лия целевых ключевых слов читателя начинает «тошнить». Показатель клас- сической тошноты – это квадратный корень из числа, обозначающего частоту употребления слова в тексте. Если вы используете слово на странице 25 раз, то его классическая тошнота будет равна 5 вне зависимости от объема текста. При этом показатель классической тошноты не может быть меньше 2,64: если слово использовано в тексте менее 7 раз, корень квадратный в этом случае из- влекается из числа 7.

Очевидно, что релевантность и тошнота – показатели-антиподы, выиг- рывая в одном мы проигрываем в другом и достижение их разумного баланса будет свидетельствовать о квалификации SEO-специалиста. На сегодня при- нято считать, что показатель частоты встречаемости ключевого слова в тексте в 3.5-5% является оптимальным.

Для того чтобы выше по рангу оказывались ресурсы с понятным и при- ятным для прочтения тестом поисковые системы постоянно усложняют свои алгоритмы ранжирования, отдавая предпочтения тем страницам, где исполь- зуются синонимы ключевых слов, семантически связанные конструкции. Так, например, Google сегодня использует при ранжировании такой фактор как ла- тентно-семантический индекс ключевых слов в контенте (LSI), который поз- воляет поисковикам бороться с ситуациями перегруженности контента од- ними и теми же ключевыми словами, которые говорят либо о неумелом насы- щении ключевиками при формировании контента, либо о плохом качестве тек- ста, что тоже должно отталкивать посетителей;

Повышает ранг сайта (страницы) также частое обновление контента, наличие мультимедийных материалов, наличие дополнительных полезных виджетов и сервисов.

* **соответствие** заголовка сайта (тег *title*), его описания (метатег *descrip- tion*), ключевых слов (метатег *keywords*) основному содержимому страницы;
* **насыщение целевыми ключевыми словами** значимых элементов страницы. Релевантность страницы запросу будет выше, если ключевые слова запроса будет встречаться в таких областях страницы, как заголовок (тег *title*),

описание (метатег *description*), ключевые слова (метатег *keywords*), заголовка в тексте страницы (теги *h1-h6*), текст с полужирным начертанием, атрибуты *alt* в изображениях;

* **грамотная внутренняя структура сайта:** если на сайте эффективно организована система навигации по страницам, то это также повысит его ранг. Если представить структуру сайта в виде дерева, то его последний уровень (листья), которые обычно представляют собой статьи, не должны находиться слишком глубоко (3-4 клика от главной страницы). Здесь также учитываются анкорный текст - содержимое тега *a* со ссылкой на другую страницу сайта должно содержать ключевые слова адресуемой страницы. Для поднятия ранга отдельных страниц и сайта в целом может понадобиться **внутренняя пере- линковка** (см. далее);
* **вид используемых URL**. Если страницы имеют постоянные уникаль- ные адреса, то лучше использовать ЧПУ (Человеко-Понятный-URL):

shop.ru?cat=50&firm=30&good=1345

хуже чем

shop.ru/notebook/asus

Базовый принцип – в строке адреса желательно иметь 2-3 ключевых слова (возможно – в транслите).

* **возраст и качество домена.** Данная группа факторов учитывает, как давно существует домен, как часто у него менялся владелец, своевременно ли продлялся домен. Наличие точного совпадения доменного имени с ключевым словом также учитывается и повышает ранг.

**- скорость загрузки и наличие ошибок на сайте** может косвенно по- низить ранг вашего ресурса;

* **количество и качество внешних ссылок на сайте**. Ранее считалось, что внешние ссылки снижают ранг сайта, поскольку отдают часть его веса сто- ронним ресурсам. Однако сегодня разработчики поисковых систем поощряют наличие ссылок на авторитетные и тематически связанные ресурсы, считая, что подобная связь между ресурсами свидетельствует о семантический близо- сти ресурсов и осмысленности формирования контента;
1. Внешние факторы.

Данная группа факторов зависит от отношения сторонних ресурсов к ва- шему и потому менее доступна к непосредственной модификации. Факторами этой группы являются:

* **количество и качество внешних ссылок на сайт**. Наличие ссылок на страницы продвигаемого ресурса с других (желательно близких по тема- тике сайтов) является важным фактором ранжирования для современных по- исковых систем. При этом алгоритм ранжирования учитывает не только коли- чество, но и качество внешних ссылок. Для этого разработаны специальные алгоритмы ранжирования (PageRank у Google или ТИЦ у Яndex). Согласно этим алгоритмам, посредством ссылки сайт-донор передает сайту – акцептору часть своего веса, авторитета и у поисковой системы. Чем больше ТИЦ или PR у сайта-донора ссылки, тем больше вес этой ссылки, и тем больше данная ссылка принесет пользы сайту-акцептору.

На ранжирование влияет не только удельный вес внешних ссылок по ТИЦ или PageRank, но и релевантность их анкорных текстов, чтобы тематика сайта совпадала по ключевым словам с текстом ссылки. Продвигая сайт за счет наращивания внешней ссылочной массы, можно следовать различным стратегиям: либо стремиться к увеличению любых ссылок, либо получить не- сколько высокоавторитетных ссылок на сайт. Так, разместив на сайте-доноре с ТИЦ 1000 статьи по тематике продвигаемого сайта и указав там пару ссы- лок на продвигаемый сайт, можно получить больший эффект в увеличении ранга сайта, чем при размещении ста ссылок на низкоавторитетных ссылок (на сайтах-донорах с ТИЦ<50).

При анализе внешних ссылок учитываются также такие факторы как: возраст домена-донора ссылки, наличие ссылок с сайтов-конкурентов (сосе- дей по результатам поисковой выдаче), анкорный текст внешних ссылок.

- **упоминание сайта и ссылки на него в социальных сетях.** Трудно пе- реоценить роль социальных сетей в современном глобальном информацион- ном пространстве и поисковые системы отслеживают посты, комментарии, от- зывы в социальных сетях на предмет наличия в них упоминаний ссылок на те или иные страницы. наличие подобных ссылок идет в плюс рангу страницы и на этом строятся целые стратегии продвижения.

* **присутствие сайта в тематических каталогах поисковых систем** (таких как Яndex-Каталог, DMOZ, Rambler-Top100 и др.) повышает его пози- ции в итоговой выдаче.
1. Поведенческие факторы.

Данная группа факторов иллюстрирует, как ведут себя пользователи на страницах ресурсов и являются косвенными показателями его удовлетворен- ности от информации, предлагаемой на ресурсе. Показатели этой группы:

* **среднее время, проведенное посетителем на сайте**. Поисковые си- стемы засекают время, которое посетители проводят на сайте после перехода со страницы поисковой выдачи. Чем выше этот показатель, тем белее реле- вантным ожиданиям пользователя считается сайт, что повышает его ранг в си- стеме;
* **показатель отказов** подсчитывает, сколько раз пользователь сразу (не позже 15 секунд после перехода на сайт) покидает ресурс. Отказ означает априорное несоответствие контента ресурса ожиданиям клиента и снижает рейтинг сайта для поисковика;
* **повторное посещение сайта** учитывает тот факт, что посетитель впо- следствии повторно заходит на сайт, что означает для поисковой системы про- явление интереса к контенту в контексте поискового запроса. Прочими свиде- телями интереса пользователя к ресурсу могут стать занесение его адреса в закладки, дальнейшее использование прямой ссылки на ресурс в адресной строке.
* **количество комментариев на сайте** также является косвенным кри- терием интереса посетителей и дать поисковой систем повод повысить его ранг.

Для мониторинга поведенческих факторов поисковые системы исполь-

зуют:

* Специализированные сервисы (Яндекс.Метрика , Google.Analytics,

рис.2);

* Встроенные функции браузеров;
* Браузерные плагины;

Рисунок 2 – Статистика сайта в окне Яndex-метрики.

Узнав особенности работы поисковых систем, можно начинать работу по SEO-оптимизации сайта. Прежде всего необходимо определиться с ключе- выми словами в запросах пользователей, по которым вы хотели бы, чтобы ваш ресурс оказывался как можно выше в поисковой выдаче, чтобы насытить ими важные элементы структуры сайта. Помимо непосредственно ключевых слов, необходимо понимать контекст поиска информации со стороны клиенты, ко- торому вы хотели бы, чтобы ваш ресурс в наибольшей степени соответствовал. На сегодня выделяют несколько типов поисковых запросов:

1. **Навигационные запросы** осуществляются с целью попасть на конкретный сайт. В этом случае поисковый движок выполняет функцию теле- фонного справочника для пользователя, интересующегося конкретной компа- нией, персоной, событием. Такого типа запросы очень эффективны для вла- дельца сайта, поскольку клиента интересует именно этот бренд, но они не очень хорошо ориентированы на новых клиентов. Для владельцев сторонних ресурсов ориентация на подобные запросы интересна для маркетинговых ис- следований, а также для отвлечения трафика от конкурента. Примеры навига- ционных запросов: «Ближайший банкомат банка XXX», «Зарегистрироваться на рейс авиакомпании YYY», «Купить абонемент в фитнесцентр ZZZ».
2. **Информационные запросы** говорят об интересе пользователя к определенной тематике, но не обещают дальнейших транзакций. С учетом того, что интернет на сегодня для многих людей заменяет книги и энциклопедии, информационные запросы очень часто служат для удовлетворения любопытства или освежения в памяти подзабытых сведений. Примерами таких запросов могут быть: «Столица Бангладеш», «Кто исполнил главную роль в се- риале …», “Годы жизни Ф.М. Достоевского”. Несмотря на то, что подобные запросы не сулят сиюминутные выгоде в продаже или услуге, ориентация на них тоже может быть полезна, поскольку они позволяют привлечь посетите- лей на сайт, привлечь входящие ссылки, в долгосрочной перспективе могут привести к транзакциям: если вы создаете сайт книжного магазина, то запрос о годах жизни писателя можно привязать к информации о продаже книг его авторства.
3. **Транзакционные запросы** говорят об интересе пользователя в ка- кой-то услуге или товаре – он уже знает, что хочет, но не знает, где это сможет приобрести. Транзакционные запросы наиболее интересны для SEO- продвижения, поскольку их проще всего связать с услугой или товаром, пред- ставленном на сайте. Примерами подобных запросов могут быть: «Где купить пиццу в городе ….», «Стоимость тура в Египет», «Продажа смартфонов».

Выше были перечислены основные типы запросов, но ими классификация не ограничивается. Запросы бывают также мультимедийными, геозависи- мыми, нечеткими, адаптивными. Подробнее о поисковых запросах можно про- читать в.

По частоте использования запросы делятся на:

1. **Высокочастотные** (более 5 000 показов в месяц). Подобные за- просы обычно носят общий характер, не уточняют намерения посетителя. Например, по запросу «автомобиль» (почти 9 млн. просмотров в месяц по дан- ным wordstat.yandex.ru) непонятно, хочет ли ищущий узнать устройство авто- мобиля, приобрести или продать его, ознакомится с историей автомобилестро- ения. В связи с этим продвижение по подобным запросам весьма затруднено, он используется большим количеством компаний. Брать высокочастотные за- просы в семантическое ядро сайта имеет смысл лишь для продвижения ком- пании, вывода новой услуги (товара) на рынок.
2. **Среднечастотные** (от 1500 до 5000 показов в месяц). В таких за- просах посетитель уточняет свой интерес, запрос становится более конкрет- ным, сужается и его конкурентность. Например, запрос «прокат легкового ав- томобиля» дает статистику уже в 2000 показов в месяц, такой запрос отсекает компании, занимающиеся ремонтом, продажей, страхованием автомобиля. Но все равно запрос довольно популярен и подвинуть компании, давно занимаю- щиеся этим видом деятельности, будет непросто.
3. **Низкочастотные** (до 1500 показов в месяц). Ориентируясь на та- кие типы запросов, вы точно найдете своего клиента, он ищет именно то, о чем написано на вашем ресурсе. Здесь высока вероятность конверсии запроса. Од- нако подобные запросы не позволяют привлекать пользователей, не обладаю- щих еще сложившимся представлением о желаемом товаре (услуге). Пример низкочастотного запроса: «почасовой прокат автомобиля» - менее 100 показов в месяц. Если сайт рекламирует среди прочего эту услугу, то использование этих ключевых слов позволит вывести сайт в топ выдаче по указанному запросу.

Ориентация на каждый из перечисленных выше типов запросов имеет свои задачи в продвижении, и опытные SEO-специалисты рекомендуют ком- поновать поисковые запросы следующим образом:

* **ВЧ-запросы:** 10-15% от общего числа поисковых запросов, по которым продвигается сайт,
* **СЧ-запросы:** 20-40% от общего числа запросов,
* **НЧ-запросы:** 45-70% от общего числа запросов.

За счет ВЧ-запросов можно охватывать общий спрос тематики ресурса, СЧ-запросы будут привлекать посетителей, интересующихся более конкрет- ными направлениями, представленными на сайте, а за счет ВЧ-запросов можно приводить целевую аудиторию, уже готовую к покупке именно ваших товаров/услуг.

Определить частотность запросов помогают специализированные сер- висы и программы. Одним из самых популярных инструментов (и при этом бесплатным) здесь является сервис «Подбор слов» от Яндекс (wordstat.yandex.ru)



Рисунок 3 – Сервис «Подбор слов» от Яндекс.

## Данный сервис позволяет оценить частоту того или иного запроса за последний месяц, а также подобрать схожие по тематике запросы. При этом можно получать статистику с региональным фильтром, а также учи- тывать типы устройств, с которых были поданы запросы. Аналогичные сервисы есть и у Google: Google AdWords (https://adwords.google.com/) и Google-тренды (https://trends.google.com/trends).

Еще одной важной характеристикой запроса является его конку- рентность. Этот показатель можно оценить по количеству строк в поис- ковой выдаче, который выдает поисковый движок в ответ на запрос. Фак- тически, этот показатель определяет, насколько ключевые слова из этого запроса представлены на страницах ресурсов-конкурентов. Найти высо- кочастотный, но низкоконкурентный запрос, сориентировать на него SEO-продвижение и быстро выйти к вершинам выдачи – мечта любого SEO-специалиста.

Рассмотрим алгоритм SEO-оптимизации web-ресурса для повыше- ния его позиций в поисковой выдаче.

* 1. Подобрать семантическое ядро. На этом этапе необходимо проанализировать тематику ресурса, определить те ключевые слова, по которым вы хотели бы, чтобы ваш сайт появлялся как можно выше в вы- даче поисковых движков. Выбранные ключевые слова необходимо про- верить на частотность и конкурентность с использованием уже упоми- навшихся инструментов «Подбор слов» от Яндекс и Google-тренды. Для каждого выбранного запроса можно подобрать похожие запросы, оце- нить их частотность и остановиться на пропорциях 10-15% ВЧ, 20-40% СЧ, 45-70% НЧ.
	2. Выбор страниц для продвижения сайта, которые в дальнейшем планируется насыщать ключевыми словами, внутренними и внешними ссыл- ками. На этом этапе можно выполнить кластеризацию запросов, выделив от- дельные группы ключевых слов и связав те или иные страницы с отдельной группой ключей, оптимизировать их для обеспечения релевантности по этой группе.
	3. SEO-копирайтинг. Этот этап заключается в написании текстового контента, в котором процент вхождения слов из семантического ядра обеспе- чивает релевантность целевым запросам и при этом не перегружен ключевыми словами [5]. Проверить качество текста можно различными сервисами, напри- мер, https://text.ru/seo, https://istio.com или https://advego.ru/text/seo.
	4. Проведение технического аудита (анализа) сайта. На данном этапе проверяется корректность html-разметки, наличие и правильность файлов ro- bots.txt и sitemap.xml, выявляются ошибки с кодом 404, битые ссылки, дубли- рование контента, проверяется корректность работы навигации, правильность функционирования отдельных блоков, анализируется скорость работы сер- вера. Периодически полезно возвращаться к данному шагу, так как изменения, вносимые в структуру, код или контент сайт могут отрицательно сказаться на его технических параметрах.
	5. SЕО-оптимизация интернет-страниц под поисковые запросы. На этом этапе внедряются ключевые слова в значимые для продвижения позиции и структурные элементы ресурса (теги и метатеги title, description, keywords, заголовки h1-h6), внедрение ЧПУ-адресации для страниц сайта, выполняется

внутренняя перелинковка сайта. Перелинковкой сайта обеспечивает связыва- ние страниц одного сайта или разных ресурсов гиперссылками. Она применя- ется для повышения релевантности страниц сайта запросу в поисковых систе- мах, увеличения статического веса страниц за счет количества входящих ссы- лок на страницу, ускорения индексации вновь появившихся страниц сайта за счет перехода робота по ссылкам на новые страницы, предотвращения дубли- рования контента, улучшения юзабилити сайта за счет большей связности страниц. Если связывание осуществляется между страницами одного ресурса, то имеет место внутренняя перелинковка, если ссылки ведут на сторонние ре- сурсы – перелинковка будет внешней. Постраничная перелинковка (каждая страница ссылается на все остальные) подходит для низкочастотных запросов (редко используемых) и среднечастотных. Она позволяет оптимально распре- делить вес сайта – рассредоточить его в нужных пропорциях и на нужных стра- ницах. Иерархическая перелинковка (главная страница ссылается на страницы разделов или категорий, а уже те ведут на конкретные статьи) приводит к тому, что конечные узлы-страницы дерева иерархий ссылок получают очень малый вес и плохо продвигаются. При проведении внутренней перелинковки важно наполнить продвигаемую по определенной группе запросов страницу входя- щими ссылками с других страниц сайта, причем тексты ссылок должны быть релевантны запросу, то есть содержать анкорный текст со словами из запроса. Если в качестве ссылки используется изображение, ключевиками имеет смысл наполнить атрибут *alt,* в соответствии с алгоритмами ранжирования поиско- вых систем лучше организовывать ссылки со страниц, отвечающих за НЧ за- просы (которые сами по себе должны получить высокий ранг при правильном контентном наполнении), на страницы с ВЧ и СЧ страницами, увеличивая тем самым их ранг.

* 1. Внешняя SEO-оптимизация ресурса. Для ее проведения необхо- димо провести внешнюю перелинковку сайта, обеспечить его продвигаемые страницы качественными входящими ссылками путем закупки ссылок на бирже, баннерообмену, продвижению в социальных сетях, на форумах и бло- гах. Необходимо зарегистрировать ресурс в каталогах поисковых систем (та- ких как Яндекс-Каталог, DMOZ, Rambler-Top100 и др.). Не стоит сильно увле- каться закупкой ссылок и следить за качеством сайтов-доноров, иначе внеш- ние ссылки могут сыграть в минус общему рангу сайта.
	2. Улучшение юзабилити сайта. Такие общие для разработки web- ресурса пожелания, как обеспечение кроссбраузерности, эффективности ис- пользования ресурсов, работоспособности функционала, удобства навигация, приятного дизайна косвенно скажутся и на поисковом продвижении (хотя бы через поведенческие факторы ранжирования).

## Рассмотренный алгоритм использует лишь самые базовые подходы к SEO-оптимизации сайта, профессионалы оперируют гораздо более мощными инструментами и методиками. Однако, выполнив даже эти шаги, можно добиться роста позиций сайта в выдаче поисковый систем. К тому же это будут реальные шаги в этом перспективном на рынке труда

направлении, они позволят сформировать фундамент будущих опытов и

открытий в области поискового продвижения web-ресурсов.

В процессе освоения SEO-методик начинающие сталкиваются с соблаз- нами использования черного и серого SEO – ряда методик и приемов, которые кажутся перспективными и на каком-то этапе даже являются эффективными в результатах продвижении, но по сути совей являющимися попыткой обмануть поисковый движок и искусственно увеличить вес своего ресурса. Разработ- чики поисковых систем борются с подобными методами и накладывают санк- ции на сайты, замеченные в нечестном продвижении вплоть до исключения из индекса. К черному SEO относят:

* использование скрытого текста (цвет шрифта и фона текста совпадают, текст расположен в блоке, который весь или частично невидим, используется очень малый шрифт);
* поисковым роботам и посетителям сайта демонстрируются различные версии ресурса (клоакинг);
* автоматическая регистрация (прогон) по множеству каталогов;
* автоматическая генерация текстов, автоматический обмен ссылками;
* осуществлять спам;
* создание дорвеев – небольших сайтов, наполненных бессмысленным контентом с высокой плотностью ВЧ- и СЧ-запросов и последующее перена- правление трафика с него на реальный ресурс;
* создание сети сателлитов – групп связанных web-серверов, которые совместно продвигают один целевой ресурс.

Серое SEO занимает промежуточное место между полностью разрешен- ным белым и запрещенным черным. К серому SEO-относится в том числе за- купка внешних ссылок, которая не приветствуется поисковыми машинами, но без которой на практике, с учетом ее повсеместного использования, очень за- труднительно будет обойти конкурентов. Практически все меры внешнего SEO-продвижения можно, когда репутацию своему сайту в виде статей, ссы- лок, отзывов, владельцы инициируют сами, можно отнести к серому SEO, раз- работчики поисковых движков стараются пресекать подобные методы накру- чивания рейтинга и их использование должно быть умеренным.

# Порядок выполнения работы.

Основная задача лабораторной работы – провести SEO-оптимизацию web-сайта, разработанного в рамках предыдущих лабораторных работ. С уче- том учебного характера разработанного ресурса необходимо сосредоточиться на внутренних факторах. Для этого необходимо выполнить следующие шаги:

1. Выбрать семантическое ядро сайта. С использованием одного из специализированных сервисов (например, «Подбор слов» от Яндекс) выбрать несколько ключевых слов по тематике сайта. Свести выделенные слова в таб- лицу:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ключевые слова | Тип(НЧ, СЧ, ВЧ) | Прогнозируемая частота появления (по данным сер-виса) | Конкурент- ность (по вы- даче поиско-вой системы) |
|  |  |  |  |  |

В семантическое ядро включить 1-2 ВЧ запроса, 3-4 СЧ запроса и 5-6 НЧ

запросов. Таблицу включить в отчет по лабораторной работе.

1. Выбрать страницу сайта, которая будет основной продвигаемой (можно ограничиться одной). Провести SEO-копирайтинг – составить текст страницы таким образом, чтобы вхождение ключевых слов в текст соответ- ствовало требованиям по показателям релевантности и тошнотности. Про- верку осуществлять с использованием специализированных сервисов (напри- мер, https://text.ru/seo, https://istio.com или https://advego.ru/text/seo). Результат проверки включить в отчет по лабораторной работе.
2. Насытить ключевыми словами значимые для продвижения пози- ции и структурные элементы страницы (теги и метатеги title, description, key- words, заголовки h1-h6). Выполнить внутреннюю перелинковку сайта с целью продвижения выбранной страницы.
3. Добавить на продвигаемую страницу несколько ссылок на автори- тетные ресурсы схожей тематики.
4. Создать файл robots.txt, в котором для основных поисковых робо- тов Яндекс и Google запретить индексацию страниц регистрации и аутентифи- кации пользователей.
5. Рекомендовать поисковым движкам повторно посещать сайт не чаще одного раза в неделю, разрешить им переход по ссылкам на страницах сайта.
6. Показать преподавателю результат работы в виде SEO-

оптимизированного сайта

1. Предоставить преподавателю отчет по лабораторной работе, от- читать лабораторную работу.

# Содержимое пояснительной записки.

1. Титульный лист
2. Результаты выбора семантического ядра в виде таблицы
3. Результаты анализа частоты вхождения ключевых слов в текст целе- вой страницы
4. html-код продвигаемой страницы
5. Содержимое файла robots.txt

**Повысить безопасность веб-приложения**

**Цель:** найти максимальное количество уязвимостей различных типов.

**Теоретический материал:**

### Используйте инструменты для анализа защищенности

Прежде чем искать уязвимости вручную проверьте приложение автоматизированными средствами. Они выполняют тесты на проникновение, пытаются его взломать, например, при помощи SQL-инъекции.

Ниже приведена подборка бесплатных инструментов.

**Приложения и фреймворки**

* [OpenVAS](http://www.openvas.org/) сканирует узлы сети на наличие уязвимостей и позволяет управлять уязвимостями.
* [OWASP Xenotix XSS Exploit Framework](https://www.owasp.org/index.php/OWASP_Xenotix_XSS_Exploit_Framework) сканирует ресурс на возможность эксплуатации XSS-уязвимостей.
* [Approof от Positive Technologies](http://approof.ptsecurity.ru/) проверяет конфигурацию веб-приложения, сканирует на наличие уязвимых компонентов, незащищенных чувствительных данных и вредоносного кода.

**Онлайн-сервисы**

* [SecurityHeaders.io](https://securityheaders.io/) проверяет на наличие и корректность заголовков ответа сервера, отвечающих за безопасность веб-приложения.
* [Observatory by Mozilla](https://observatory.mozilla.org/) сканирует ресурс на наличие проблем безопасности. Кроме своих результатов, при выборе соответствующей опции, собирает и добавляет к отчету аналитику со сторонних сервисов анализа защищённости.
* [One button scan](https://sergeybelove.ru/one-button-scan/) сканирует на наличие уязвимостей компоненты ресурса: DNS, HTTP-заголовки, SSL, чувствительные данные, используемые сервисы.
* [CSP Evaluator](https://csp-evaluator.withgoogle.com/) проверяет правильность составления политики безопасности содержимого (CSP) и устойчивость к XSS.
* [SSL Server Test](https://www.ssllabs.com/ssltest/) выполняет анализ SSL-конфигурации веб-сервера.
* [ASafaWeb](https://asafaweb.com/) проверяет на наличие распространённых уязвимостей конфигурации сайтов, написанных на ASP.NET.
* [Snyk](https://snyk.io/) сканирует JavaScript, Ruby и Java-приложения на наличие уязвимостей и, при необходимости, исправляет проблемы безопасности. Интегрируется с GitHub репозиторием для проведения автоматической проверки и оповещает о найденных уязвимостях.

Перед сканированием веб-приложения онлайн-сервисами обратите внимание на условия использования. Некоторые из них публикуют отчеты о проверенных сайтах в открытом виде.

Результаты автоматизированных тестов приводят в замешательство, так как показывают всевозможные разновидности потенциальных угроз. Но к каждой выявленной проблеме прилагается объяснение. В первую очередь проанализируйте и исправьте критические замечания.

После того как в приложение внесены рекомендуемые изменения по безопасности, просканируйте повторно, чтобы убедиться в правильности принятых мер.

Если автоматической проверки мало, попытайтесь вручную взломать свой ресурс, изменяя значения POST и GET запросов. Здесь может помочь отладочный прокси-сервер (например, [Fiddler](http://www.telerik.com/fiddler%22%20%5Ct%20%22_blank)), так как он перехватывает значения HTTP запросов между браузером и сервером. Уделите отдельное внимание формам — попробуйте обойти валидацию, чтобы внедрить XSS-инъекцию.

Если на сайте есть страницы, которые доступны только после аутентификации, попробуйте выдать себя за другого пользователя. Для этого измените параметры URL (например, ID пользователя) или значения Cookie.

### Защитите пользовательские данные с помощью HTTPS

HyperText Transfer Protocol Secure (HTTPS) — расширение HTTP, которое поддерживает шифрование и защищает данные пользователей при передаче в Интернете. HTTPS гарантирует целостность и конфиденциальность взаимодействия с сервером. В марте этого года популярность HTTPS [достигла переломного момента](https://tproger.ru/translations/https-tipping-point/), и вскоре его использование станет «нормой», а не исключением, как это было раньше.

Используйте HTTPS, если пользователи передают на сервер личные данные: информацию о кредитной карте, персональные данные и даже адреса посещённых страниц. Если при отправке данных с формы авторизации устанавливаются cookie-файлы, которые потом отправляются при каждом запросе к серверу, злоумышленник может получить их и подделать запрос к серверу. В результате он перехватит сессию пользователя. Чтобы предотвратить это, используйте HTTPS на всех страницах сайта.

Это просто: SSL-сертификат генерируется бесплатно (например, на [Let’s Encrypt](https://letsencrypt.org/%22%20%5Ct%20%22_blank)), для большинства платформ созданы инструменты автоматического получения и установки сертификата. Остаётся только включить на сервере поддержку HTTPS.

Примечательно, что компания Google объявила планы о предоставлении сайтам, использующим защищённое соединение, преимущества в результатах поиска. Это пряник. Кнутом станут предупреждения о небезопасном соединении в браузерах, количество которых будет расти. HTTP уходит в прошлое, поэтому самое время перейти на HTTPS.

Если HTTPS уже настроен, хорошей практикой считается использование HTTP Strict Transport Security (HSTS) — заголовка ответа сервера, который запрещает для домена использование незащищённого соединения.

### Обновляйте программное обеспечение

Это жизненно важно для безопасности веб-приложения. Хакеры регулярно обнаруживают и сразу же применяют новые уязвимости операционных систем и другого программного обеспечения: HTTP-серверов или систем управления контентом (CMS).

Если ресурс расположен на сервере хостинг-провайдера, то установка обновлений для операционной системы входит в комплекс услуг. В противном случае нужно самостоятельно обновлять операционную систему.

Если ресурс работает на базе движка стороннего производителя (CMS или форума), устанавливайте исправления безопасности сразу после выпуска. Большинство разработчиков информируют об обновлениях через рассылку или RSS-канал с описанием исправленных проблем. WordPress и Umbraco, кроме того, уведомляют о доступных обновлениях при входе в панель управления.

Многие разработчики используют менеджеры пакетов (например, Composer, NPM или RubyGems), чтобы устанавливать зависимые компоненты для приложений. В этих пакетах также обнаруживают уязвимости, поэтому следите за их обновлением. Чтобы автоматически получать уведомления о проблемах безопасности пакетов проекта, используйте инструменты вроде [Gemnasium](https://gemnasium.com/%22%20%5Ct%20%22_blank).

 Вот некоторые из уязвимостей веб приложений, который содержит DVWA:

* Брутфорс: Брутфорс HTTP формы страницы входа - используется для тестирования инструментов по атаке на пароль методом грубой силы и показывает небезопасность слабых паролей.
* Исполнение (внедрение) команд: Выполнение команд уровня операционной системы.
* Межсайтовая подделка запроса (CSRF): Позволяет «атакующему» изменить пароль администратора приложений.
* Внедрение (инклуд) файлов: Позволяет «атакующему» присоединить удалённые/локальные файлы в веб приложение.
* SQL внедрение: Позволяет «атакующему» внедрить SQL выражения в HTTP из поля ввода, DVWA включает слепое и основанное на ошибке SQL внедрение.
* Небезопасная выгрузка файлов: Позволяет «атакующему» выгрузить вредоносные файлы на веб сервер.
* Межсайтовый скриптинг (XSS): «Атакующий» может внедрить свои скрипты в веб приложение/базу данных. DVWA включает отражённую и хранимую XSS.
* Пасхальные яйца: раскрытие полных путей, обход аутентификации и некоторые другие.

Уровни безопасности

DVWA имеет три уровня безопасности, они меняют уровень безопасности каждого веб приложения в DVWA.

* Невозможный – этот уровень должен быть безопасным от всех уязвимостей. Он используется для сравнения уязвимого исходного кода с безопасным исходным кодом. До DVWA v1.9 этот уровень был известен как «высокий».
* Высокий - Это расширение среднего уровня сложности, со смесью более сложных или альтернативных плохих практик в попытке обезопасить код. Уязвимости не позволяют такой простор эксплуатации как на других уровнях.
* Средний - Этот уровень безопасности предназначен главным образом для того, чтобы дать пользователю пример плохих практик безопасности, где разработчик попытался сделать приложение безопасным, но потерпел неудачу. Он также выступает вызовом для пользователей, чтобы они усовершенствовали свои техники эксплуатации.
* Низкий - Этот уровень безопасности совершенно уязвим и совсем не имеет защиты. Его предназначение быть передовым примером среди уязвимых веб приложений, примером плохих практик программирования и служить платформой обучения базовым техникам эксплуатации.

Каждая уязвимая страница в DVWA имеет кнопку 'view source' (показать исходный код), эта кнопка используется для просмотра и сравнения исходного кода каждой уязвимости по сравнению с её безопасным уровнем. Это позволяет пользователю с лёгкостью получить доступ к исходному коду для сравнения безопасных и небезопасных практик написания кода.

Также в DVWA вы можете включить или отключить PHP-IDS. PHP-IDS - это популярная PHP система обнаружения вторжений (IDS), которые также известны как файерволы веб приложений (Web Application Firewall (WAF)). PHP-IDS работает посредством фильтрации по чёрному списку потенциально уязвимого кода любого пользовательского ввода. PHP-IDS используется в DVWA для служения живым примером, как WAF может помочь улучшить безопасность в веб приложениях, а в Damn Vulnerable Web Application (DVWA) - случае, как WAF можно обойти. PHP-IDS может быть включена или отключена одним кликом по соответствующей кнопке.

Инструменты

Для тестирования должен использоваться дистрибутив Kali Linux релиза не ниже 2016.2. Можно пользоваться любыми иструментами дистрибутва, включая, но не ограничиваясь [nikto](https://jehy.github.io/mami/infosec-lab8-nikto.html), [SQLMap](https://jehy.github.io/mami/infosec-lab8-sqlmap.html), [Hydra](https://jehy.github.io/mami/infosec-lab8-hydra.html).

**Практическое задание**

* Установить Kali Linux (в качестве виртуальной машины) или воспользоваться дистрибутивом в классе. Полное описание установки можно прочитать [здесь](https://jehy.github.io/mami/infosec-lab8-kali.html). При необходимости можно взять образ у других студентов группы.
* Установить DVWA из форка <https://github.com/jehy-security/DVWA> согласно инструкциям из репозитория, поставить "средний" уровнь безопасности. При неоходимости можно воспользоваться [подробной инструкцией по установке](https://jehy.github.io/mami/infosec-lab8-dvwa.html). Так же можно взять образ с утановленным приложением у других студентов группы;
* Реализовать успешную эксплуатацию каждой из 10 уявимостей (по числу пунктов меню приложения). Некоторые задачи можно выполнить несколькими методами, но достаточно описать хотя бы один.
* Написать отчёт в свободной форме, в котором показать успешное решение задач из предыдущего пункта.

Вопросы

1. Перечислите основные уязвимости, касающиеся веб приложений.
2. Можно ли ставить DVWA на свой публичный веб сервер?
3. Чем отличается тестирование DVWA от тестирования публичного сервера?
4. Что представляет собой дистрибутив Kali Linux?
5. Перечислите и опишите вкратце несколько компонентов, входящих в состав Kali Linux;