

|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

**Институт Информационных технологий**

**Кафедра Инструментального и прикладного программного обеспечения**

ПРАКТИЧЕСКАЯ РАБОТА №4

по дисциплине «Разработка серверных частей интернет-ресурсов»

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил студент группы | ИКБО-01-20 |  | Клочек А.М. |
| Принял преподаватель |  |  | Волков М. Ю. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Практические работы выполнены « | | |  | » | 2022 г. |
|  |  |  |  |  | Подпись студента |
| «Зачтено» « |  | » 2022 г. |  |  |  |
|  |  |  |  |  | Подпись преподавателя |

Москва 2022

**Содержание**

[Цель работы 3](#_bookmark0)

[Задание на практическую работу 3](#_bookmark1)

[Выполнение практической работы 4](#_bookmark2)

[Вывод 8](#_bookmark3)

[Ответы на вопросы 9](#_bookmark4)

[Список информационных источников 11](#_bookmark5)

# Цель работы:

Получить навыки хранения сессионных данных, изменения способов их хранения и обработки загрузки и скачивания файлов.

# Задание на практическую работу:

Предполагается выполнить апгрейд разрабатываемого в процессе первых 4 практических работ интернет-ресурса механизмами обработки сессий и согласования контента. Предлагается добавить следующую функциональность:

1. Хранение данных сессий в БД Redis.
2. Использование данных для согласования контента на уровне сервера для формирования контента пользователя с помощью (выбор по варианту). Требуется использовать хотя бы 3 параметра для формирования индивидуального контента, например, логин пользователя, тема(темная, светлая или для людей с цветовой слепотой) и рекомендуемый язык.
   1. файлов cookie
   2. *файлов сессий*
3. Загрузка файлов в формате pdf на сервер и хранение их(выбор по варианту), а также их выдача обратно пользователю по запросу.
   1. в файловой системе сервера
   2. в реляционной базе данных (MySQL)
   3. *в не реляционной базе данных (MongoDB)*

Предполагается создание стабильной версии интернет-ресурса и сохранение предыдущей функциональности с практических работ 1-4.

Варианты:

* Кофейня,
* Строительный магазин,
* Автосервис,
* *Магазин электроники,*
* Портфолио,
* Библиотека,
* Ресторан,
* Погода,
* Ломбард*,*
* Магазин игрушек,
* Индивидуальная тема.

# Выполнение практической работы:

В ходе выполнения данной работы были использованы наработки прошлых работ. Сохранение данных сессий в БД Redis реализовано с помощью настроек соответствующих конфигурационных файлов. Согласование контента на уровне сервера (и не только) для формирования контента пользователя реализовано с помощью данных сессии, которые содержат следующие значения: тема – светлая false или темная true, логин пользователя строка и флаг включения тестовой страницы логический тип. Пользователь может выбрать параметры и задать их, сервер их запомнит в параметрах соответствующей сессии и отразит в виде измененной клиентской части. Загрузка и скачивание файлов реализовано с помощью не реляционной СУБД MongoDB, в которой хранится имя файла, идентификатор сессии пользователя, загрузившего файл и сам контент файла. При загрузке в БД сохраняются название файла, идентификатор сессии и контент файла. При скачивании пользователю отдается файл, который он загрузил в соответствующей сессии, при загрузке нового файла любые старые файлы, ассоциированные с данной сессией удаляются из БД. Также полсле загрузки в БД временный файл удаляется, также как и при скачивании, при котором из БД достается название файла и его контент, создается временный файл, в него записывается контент, файл отдается пользователю и затем файл удаляется.

Выполнение работы продемонстрировано на рисунках 1-.

Файлы проекта расположены в удаленном репозитории на платформе

GitHub.com, расположенном по ссылке: https://github.com/Agser228/RSCHIR/tree/PZ-5.

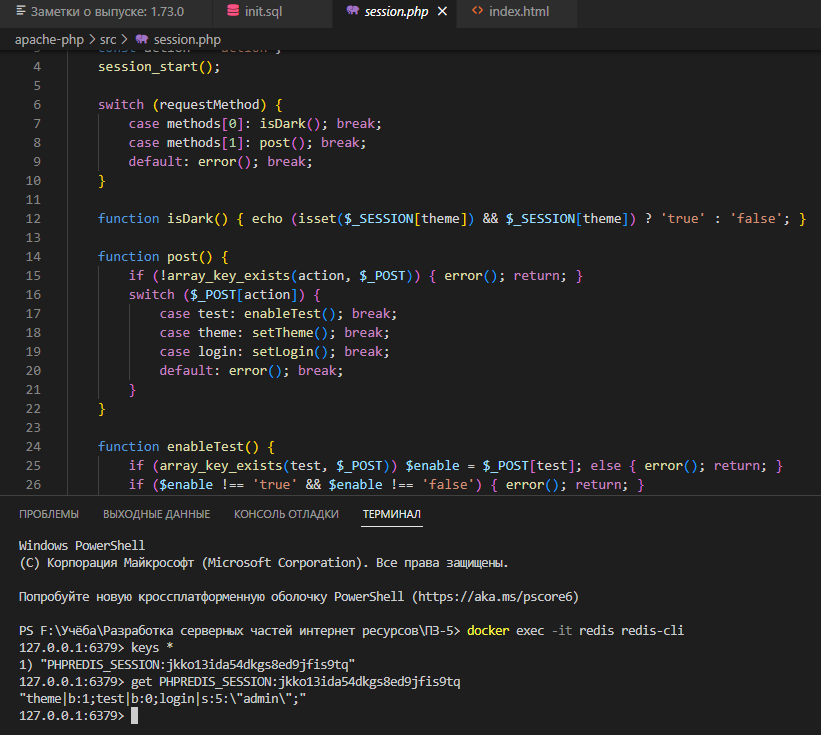


Рисунок 1 – Хранение данных сессии в БД Redis

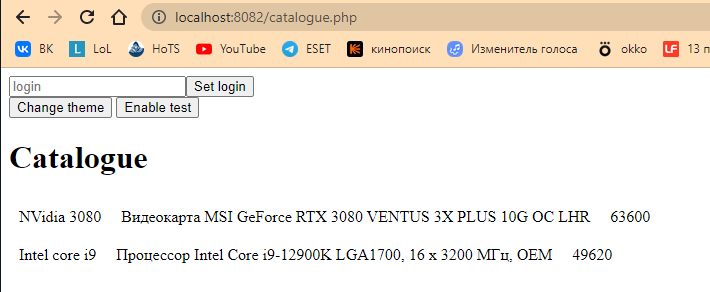


Рисунок 2 – Страница каталога (без сохраненных параметров сесии)

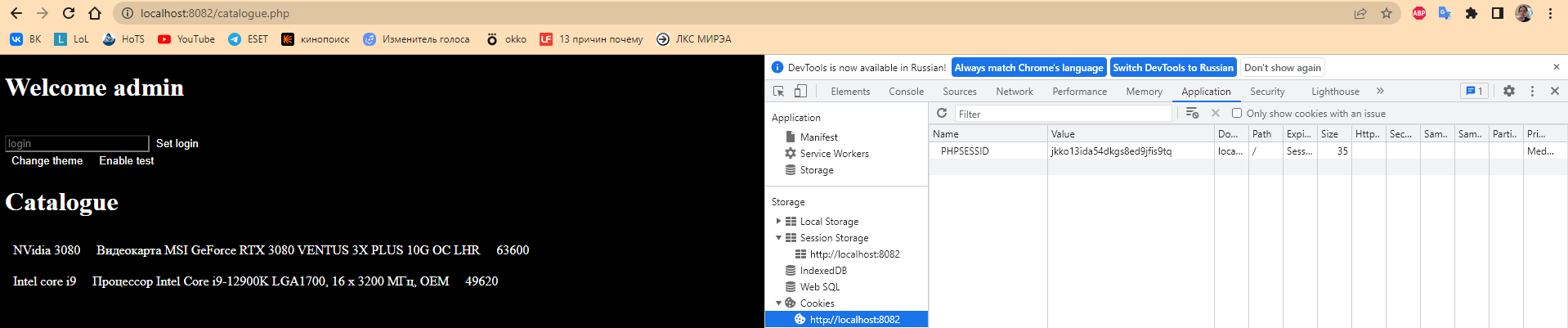


Рисунок 3 – Страница каталога (с сохраненными параметрами сесии – темной темой, логином и флагом включения теста)

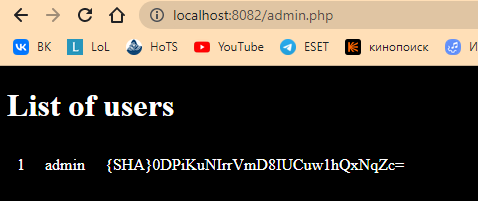


Рисунок 4 – темная тема применяется к каждой странице сайта

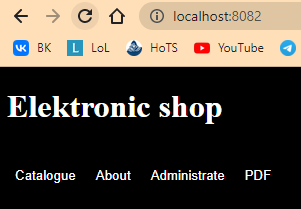


Рисунок 5 – темная тема применяется к каждой странице сайта

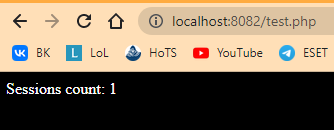


Рисунок 6 – Сраница теста сохранения параметра сессии в БД Redis

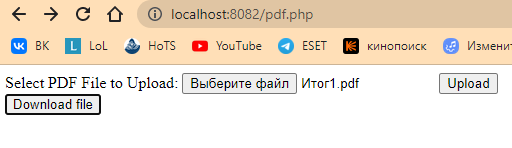


Рисунок 7 – Страница загрузки и скачивания файла

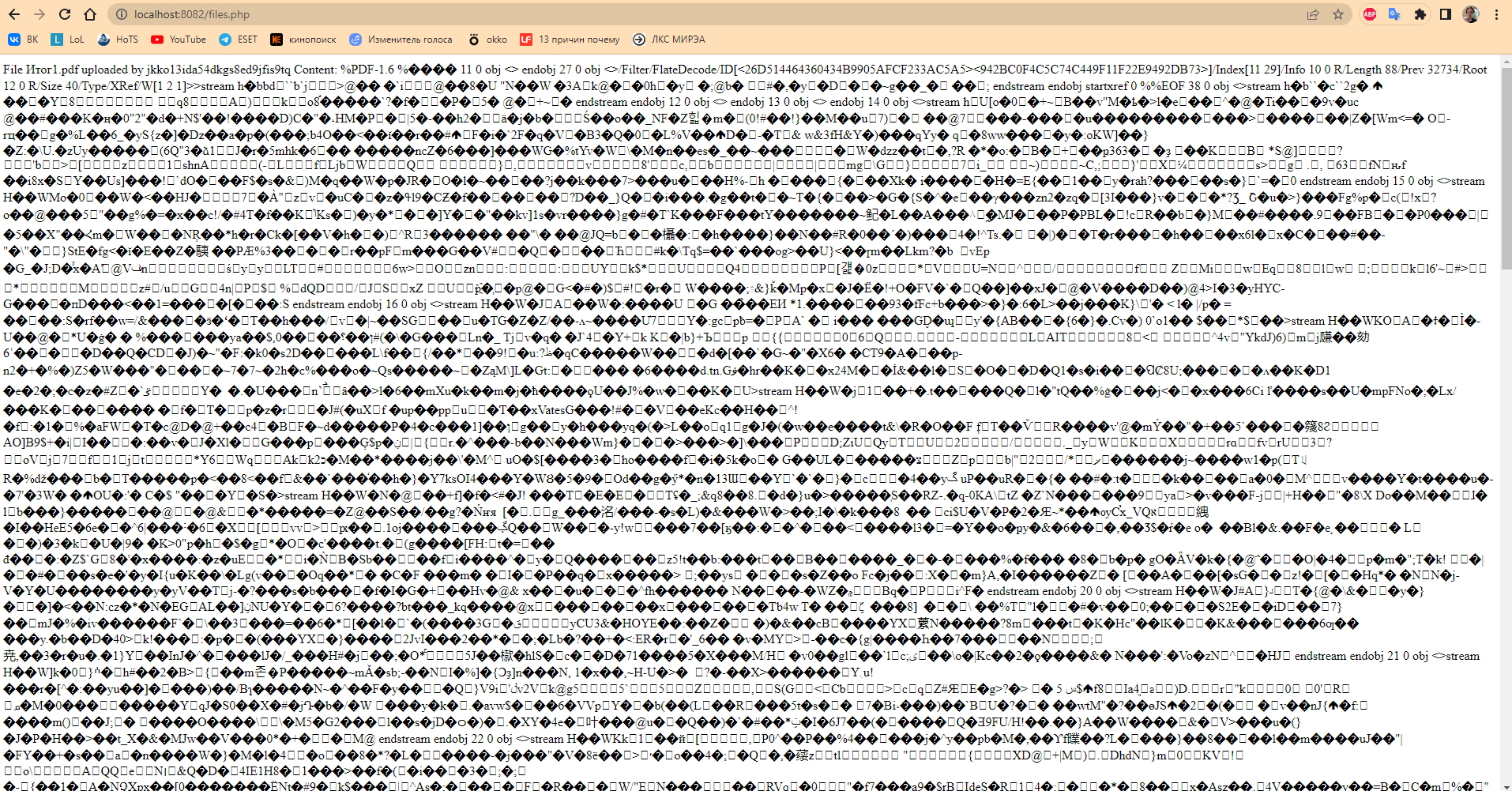


Рисунок 8 – Отладочная информация после загрузки файла (тема не применяется)



Рисунок 9 – Скачивание файлов (для тестирования использовались файлы- заглушки)

# Вывод:

В ходе выполнения данной работы была доработана предыдущая работа, добавлено сохранение пользовательских данных в сессиях, которые хранятся в БД Redis, обработка согласования контента на уровне сервера, загрузка, хранение в не реляционной СУБД и загрузка файлов.

**Ответы на вопросы:**

1. Что такое сессия в рамках веб-разработки?

Сессия в рамках интернет-ресурсов — это данные о пользовательской активности, сохраняемые между запусками сценария.

1. Что такое cookie в рамках веб-разработки?

Cookies — это данные от веб-сервера, хранимые на клиенте браузером. Чаще всего в cookies хранятся данные аутентификации пользователя, пользовательские предпочтения и настройки (например, темная тема интернет-ресурса вместо базовой светлой), данные состояния сеанса пользователя, а также пользовательская статистика.

1. Опишите механизм использования cookies.

1. Пользователь заходит на страницу авторизации на интернет-ресурсе и заполняет форму авторизации.

2. Нажатием кнопки “Войти” на сервер отправляется HTTP-запрос с данными для авторизации.

3. Сервер проверяет данные, отправленные пользователем:

a. Если данные авторизации верны, то сервер создает новую сес-сию: генерируется уникальный идентификатор сессии, а также файл на сервере, хранящий данные сессии.

b. Если данные неверны, то ошибка авторизации отправляется клиенту.

4. Клиенту отправляется идентификатор сессии в виде заголовка Set-cookie.

5. Клиент получает идентификатор сессии от сервера и сохраняет его в файле cookies.

6. Последующие запросы берут данные из файла cookies и отправляют их на сервер вместе с запросами. Сервер по этим данным идентифи-цирует пользователя.

1. Опишите простой пример работы сессий в PHP.

Если аутентификация пользователя проходит успешно, то вызывается функция начала сессии session\_start, в файл сессии сохраняется данные о том, что пользователь авторизован. Для этого в суперглобальный массив $\_SESSION записывается новое значение. Потом идет перенаправление на страницу меню интернет-ресурса.

1. Опишите способы защиты сессии пользователя.

Одним из вариантов повышение безопасности веб-приложения является неадаптивное управление сессиями.

Вторым вариантом поддерживающим вариантом является пересоздание идентификаторов сессий.

Третьим методом защиты является удаление сессий. Данные истекших сессий должны быть недоступны и удалены.

1. Верно ли, что можно хранить данные сессии в БД?

В PHP существует возможность настройки хранения сессии не только в файловой системе, но и в базе данных, например. Или дополнительное шифрование. Базовым обработчиком сессий является обработчик files, но возможно создание пользовательских обработчиков на основе реализация класса наследника SessionHandler или имплементации интерфейса SessionHandlerInterface.

1. Опишите жизненный цикл сессии.

Когда стартует сессия, PHP внутренне вызовет обработчик open с после-дующим вызовом обработчика read, который должен вернуть закодированную строку — в точности такую, какая передавалась для сохранения. После возвра-щения обработчиком read закодированной строки, PHP декодирует её и запол-нит получившимся массивом суперглобальный массив $\_SESSION.

Когда PHP завершает исполнение скрипта (или когда вызвана функция session\_write\_close()), PHP внутренне закодирует суперглобальный массив $\_SESSION, и передаст эти данные с идентификатором сессии функции обрат-ного вызова write. После того, как отработает функция обратного вызова write, PHP внутренне вызовет обработчик функции обратного вызова close.

Когда сессия специально уничтожена, PHP вызовет обработчик destroy с идентификатором сессии.

1. Верно ли, что можно убрать механизм обработки сессий?

session.auto\_start указывает на то, будет ли модуль сессии запускать сес-сию автоматически при старте. По умолчанию данных механизм отключен.

1. Опишите примеры настройки сессии во время выполнения.

Также возможна настройка данной конфигурации во время исполнения

<?php

ini\_set("session.use\_cookies", 0);

ini\_set("session.use\_only\_cookies", 0);

ini\_set("session.use\_trans\_sid", 1);

ini\_set("session.cache\_limiter", "");

session\_start();

С помощью функции ini\_set возможна настройка во время исполнения, но важно ее проводить до выполнения какой-либо функциональности.

1. Опишите директивы конфигурации файловой системы и потоков в PHP.

allow\_url\_fopen = On

allow\_url\_include = Off

user\_agent="PHP"

default\_socket\_timeout = 60

from="john@doe.com"

auto\_detect\_line\_endings = Off

sys\_temp\_dir = "/tmp"

allow\_url\_fopen указывает на обработку объектов URL как обычных файлов. Обёртки, доступные по умолчанию, служат для работы с удалёнными файлами с использованием ftp или http протокола. Некоторые модули, например, zlib, могут регистрировать собственные обёртки.

allow\_url\_include указывает на возможность использования оберток fopen, которые поддерживают работу с URL, в функциях include, include\_once, require, require\_once. Данная директива объявлена устаревшей с версии 7.4.0.

user\_agent указывает на задаваемую PHP строку “User-agent”. По умолчанию данная директива не задается и поэтому значение данной строки равно пустой строке.

default\_socket\_timeout указывает на значение тайм-аута по умолчанию (в секундах) для потоков, использующих сокеты. Отрицательное значения означает бесконечное время ожидания. Бесконечные времена ожидания использовать не рекомендуется, чтобы не перегружать систему.

from указывает на адрес email, используемый в соединениях FTP без авторизации, а также в качестве значения заголовка From в HTTP соединениях при использовании ftp и http оберток, соответственно.

auto\_detect\_line\_endings указывает на включения функционала PHP для определения способа завершения строк.

sys\_temp\_dir указывает на путь к директории, в которой будут храниться временные файлы.

1. Какой тип ресурса использует файловая система. Опишите данный тип.

Для работы с файлами файловая система использует потоки (streams) в качестве собственного типа ресурсов. Потоки были введены как инструмент для работы с файлами, сетевого обмена, сжатия данных и выполнения других операций с помощью одного общего набора функций.

1. Как открыть и закрыть файл с помощью PHP.

В данном примере идет открытие файла для чтения с помощью функции fopen и чтение всего содержимого файла в строку с помощью функции fread, первым параметром которой является дескриптор файла, а вторым количество байт для чтения. В данном случае мы считываем весь файл за счет получения размера файла с помощью функции filesize. После конца работы с файлом нужно обязательно закрыть его с помощью функции fclose.

1. Как производится чтение и запись файлов в PHP.

fread/fwrite

1. Опишите как считать только часть файла, как считывать файл последовательно и считать весь файл целиком.

Если нам надо прочитать файл полностью, то мы можем облегчить себе жизнь, применив функцию file\_get\_contents()

Также можно провести поблочное считывание, то есть считывать определенное количество байт из файла с помощью функции fread()

1. Как производится создание и удаление файлов с помощью PHP.

Функция file\_put\_contents() позволяет создать файл и заполнить его данными.

Для удаления файлов используется функция unlink()

1. С помощью каких функций и какую информацию о файле можно получить с помощью PHP?

file\_exists проверяет существование указанного файла или каталога.

fileatime возвращает время последнего обращения к файлу.

fileowner возвращает идентификатор владельца файла.

fileperms возвращает информацию о правах на файл. В примере идет отоб-ражение прав доступа в виде восьмеричного числа.

filesize возвращает размер файла в байтах.

filetype возвращает тип файла.

1. Что такое DOM?

Для работы с xml одной из крупных встроенных библиотек является DOM. Аббревиатура расшифровывается как Document Object Model. Данный набор функциональности поддерживает обработку как XML-документов, так и HTML-документов.

1. Как создать документ и работать с ним с помощью модуля DOM?

<?php

$doc = new DOMDocument('1.0');

// мы хотим красивый вывод

$doc->formatOutput = true;

$root = $doc->createElement('book');

$root = $doc->appendChild($root);

$title = $doc->createElement('title');

$title = $root->appendChild($title);

147

$text = $doc->createTextNode('Это заголовок');

$text = $title->appendChild($text);

echo "Сохранение всего документа:\n";

echo $doc->saveXML() . "\n";

1. Что такое JSON?

JavaScript Object Notation (JSON)

1. Как декодировать строку JSON и вернуть JSON-представление данных?

$arr = json\_decode($json);

1. Как проанализировать и выявить ошибки при кодировании и декодировании JSON?

$error = json\_last\_error();

1. Опишите создание, сохранение, парсинг XML-документа c помощью PHP.

$dom = new DOMDocument;

$dom->loadXML($xml);

$books = $dom->getElementsByTagName('book');

foreach ($books as $book) {

echo $book->nodeValue, PHP\_EOL;

}

1. Что такое драйвер в рамках взаимодействия с СУБД?

Определим драйвер — это специализированное ПО, созданное для взаимодействия с определенным сервером баз данных.

1. Опишите добавление записи в рамках использования модуля mysqli для взаимодействия с БД MYSQL.
2. Что такое постоянное соединение, опишите проблемы данного подхода и его решение в модуле mysqli.

Идея постоянных подключений состоит в том, чтобы соединение между клиентским процессом и базой данных можно было использовать повторно, особенно когда требуется создавать и закрывать соединения множество раз. Это бы позволило снизить накладные расходы на создание новых подключений каждый раз, когда они требуются, за счёт использования существующих кешированных подключений, свободных для повторного использования.

В отличие от модуля mysql, в mysqli нет отдельной функции для создания постоянных соединений. Чтобы открыть постоянное соединение, при создании подключения к имени хоста нужно добавить префикс p:.

При использовании постоянных соединений можно столкнуться с проблемой, которая заключается в том, что клиенты могут оставлять такие подключения в непредсказуемом состоянии. Например, клиент ставит блокировку на таблицу, а затем аварийно завершает работу. То есть блокировка снята не будет. Новый клиентский процесс, использующий это подключение повторно, получит его как есть, и вынужден будет провести какую-то очистку подключения, прежде чем начать его использовать. Соответственно, в задачи программиста входит ещё и проверка подобных ситуаций и внедрение кода, осуществляющего такую очистку.

Тем не менее, в mysqli эта проблема решена. В модуле есть встроенный функционал, осуществляющий очистку соединений и переводящий их в состояние пригодное для использования. Код очистки, реализованный в mysqli включает следующие операции:

* Откат активных транзакций
* Закрытие и удаление временных таблиц
* Снятие блокировки с таблиц
* Сброс переменных сессии
* Закрытие подготовленных запросов (всегда происходит в PHP)
* Закрытие обработчиков
* Снятие блокировок, установленных функцией GET\_LOCK()

Это позволяет быть уверенным в том, что возвращённые из пула соединения готовы к использованию в клиентских процессах.

1. Опишите основные особенности БД MongoDB.

Главные особенности MongoDB:

Это кроссплатформенная документоориентированная база данных NoSQL с открытым исходным кодом

Она не требует описания схемы таблиц, как в реляционных БД. Данные хранятся в виде коллекций и документов

Между коллекциями нет сложных соединений типа JOIN, как между таблицами реляционных БД. Обычно соединение производится при сохранении данных путем объединения документов

Данные хранятся в формате BSON (бинарные JSON-подобные документы)

У коллекций не обязательно должна быть схожая структура. У одного документа может быть один набор полей, в то время как у другого документа — совершенно другой (как тип, так и количество полей)

1. Опишите процесс добавления новой записи в СУБД MongoDB с помощью соответствующего драйвера.

<?php

require 'vendor/autoload.php'; // подключаем автоподгрузчик классов Composer

$client = new MongoDB\Client("mongodb://localhost:27017");

$collection = $client->demo->beers;

$result = $collection->insertOne( [ 'name' => 'Hinterland', 'brewery' => 'BrewDog' ] );

echo "Идентификатор вставленного документа '{$result->getInsertedId()}'";

?>

1. Опишите процесс получения и обработки записей с помощью драйвера MongoDB.

$item = $collection->findOne(array(

'\_id' => new MongoId('4e49fd8269fd873c0a000000')));

1. Опишите получение записей в рамках использования модуля mysqli для взаимодействия с БД MYSQL.

<?php

$mysqli = mysqli\_connect("example.com", "user", "password", "database");

$result = mysqli\_query($mysqli, "SELECT 'Пожалуйста, не ис-пользуйте устаревший модуль mysql в новых проектах.' AS \_msg FROM DUAL");

$row = mysqli\_fetch\_assoc($result);

echo $row['\_msg'];

1. Опишите поиск записей, подсчет и ограничение выборки с помощью драйвера MongoDB.

$mongo = new MongoDB\Driver\Manager("mongodb://localhost:27017");

$regex1 = new MongoDB\BSON\Regex("^[A-z]","i");

$pipeline = [

[ '$match' => ['searchcontent.name' => $regex1] ],

[ '$group' => ['\_id' => '$searchcontent.name'] ],

[ '$limit' => 50 ],

[ '$skip' => 10 ],

];

$aggregate = new \MongoDB\Driver\Command([

'aggregate' => 'search',

'pipeline' => $pipeline

]);

$cursor = $mongo->executeCommand('webdb', $aggregate);

foreach($cursor as $key => $document) {

var\_dump($document);

}

# Список информационных источников:

1. Документация по языку PHP, электронный ресурс URL: [https://php.](https://ktor.io/)net (дата обращения: 11.09.2022) – Текст: электронный;
2. Документация по системе Docker, электронный ресурс URL: https://docs.docker.com (дата обращения: 11.09.2022) – Текст: электронный.
3. Документация по HTML, CSS, JavaScript, электронный ресурс URL: https://developer.mozilla.org (дата обращения 11.09.2022) – Текст: электронный.