МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий (ИТ)

Кафедра Инструментального и Прикладного Программного Обеспечения (ИиППО)

ПРАКТИЧЕСКАЯ РАБОТА №5

по дисциплине

«Разработка серверных частей интернет ресурсов»

Тема: **«Сессии. Файлы. БД»**

Студент группы ИКБО-13-20 Лянной А.П.

_____

(подпись студента)

Принял руководитель работы Волков М.Ю.

_____

(подпись руководителя)

Практическая работа выполнена «___»_____2022 г.

Зачтено «___»_____2022 г.

Москва 2022 г.

ОГЛАВЛЕНИЕ

1. Цель работы

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.

a. файлов cookie

b. файлов сессий

3. Загрузка файлов в формате pdf на сервер и хранение их (выбор по варианту), а также их выдача обратно пользователю по запросу.

a. в файловой системе сервера

b. в реляционной базе данных

c. в не реляционной базе данных

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

2. Ход работы

Для хранения данных сессии в Redis необходимо модифицировать файлы Docker и конфигурационные файлы (листинг 1).

Листинг 1 – Модификация файлов docker'а

```
//docker-compose.yml

services:
    redis:
        image: redis
        volumes:
        - redis_volume:/data
        ports:
        - 6379:6379

volumes:
    redis_volume:

//Dockerfile (apache)

RUN pecl install -o -f redis \
&&  rm -rf /tmp/pear \
&&  docker-php-ext-enable redis
COPY php.ini /usr/local/etc/php

//php.ini

[Session]

session.save_handler = redis
session.save_path = "tcp://redis"
```

Теперь сессия хранится в БД Redis (рис. 1).

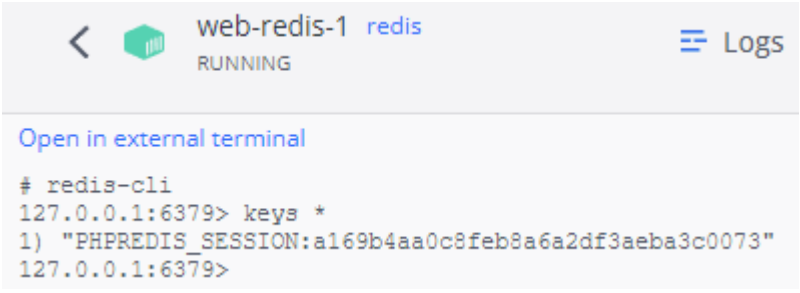

Рисунок 1 – Сессия в Redis

Реализуем создание, чтение, модификацию и удаление cookie для согласования контента (листинг 2).

Листинг 2 – Реализация согласования контента с помощью cookie

```
//cookieLogin.php (только PHP)

<?php
//Если пользователь недавно вошёл –
//пропустить вход
if (isset($_COOKIE['logged']))
{
     header('Location: cookieThemes.php');
     exit;
}
?>

//cookieLoginer.php

<?php
setcookie('logged', $_GET["username"], time() + 60, '/');
if (!isset($_COOKIE[$_GET["username"]]))
     setcookie($_GET["username"], '00', time() + 300, '/');
?>

//cookieThemes.php

//Если срок входа пользователя закончился
//или пользователь не вошёл в систему –
//отправить на страницу входа
<?php
if (!isset($_COOKIE['logged']))
{
     header('Location: cookieLogin.php');
     exit;
}

//Пример согласования контента при входе
function cook($options, $i)
{
     echo $options[$_COOKIE[$_COOKIE['logged']][$i]];
}
?>

<label for="theme1"><?php cook(array("Dark", "Тёмная"),
0);?></label>

//cookieLogouter.php

<?php
setcookie($_COOKIE['logged'], $_GET['lang'].$_GET['theme'],
time() + 300, '/');
setcookie('logged', "", time() - 1, '/');
?>
```

Выполним следующую последовательность действий – войдём, как user1, поставим тёмную тему и английский язык, выйдем, войдём, как user2, поставим светлую тему и русский язык, выйдём, затем зайдём на user1 и user2, чтобы удостовериться, что контент согласуется в соответствие с предпочтениями (рис. 2, 3).

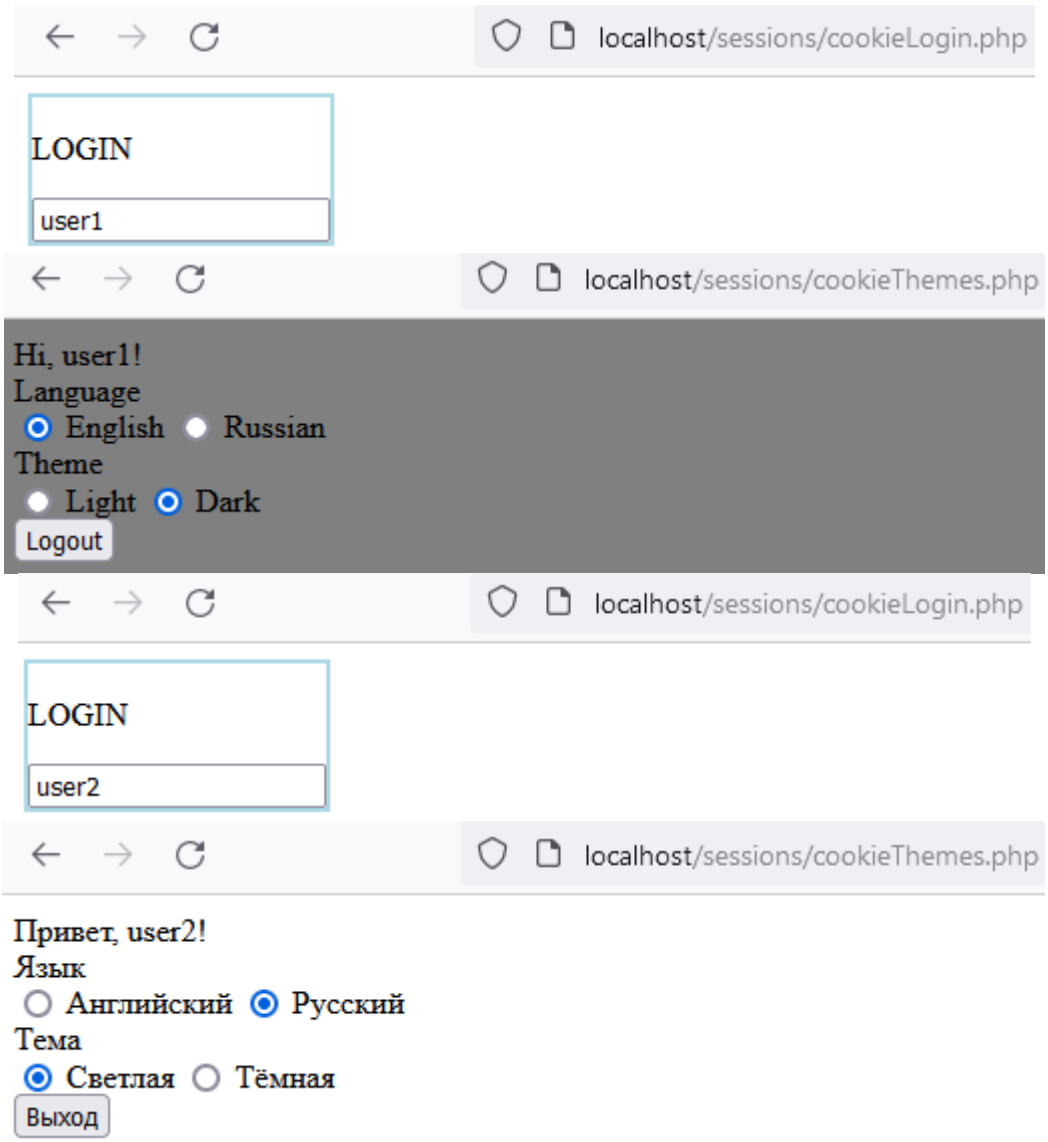

Рисунок 2 – Установка предпочтений двум пользователям

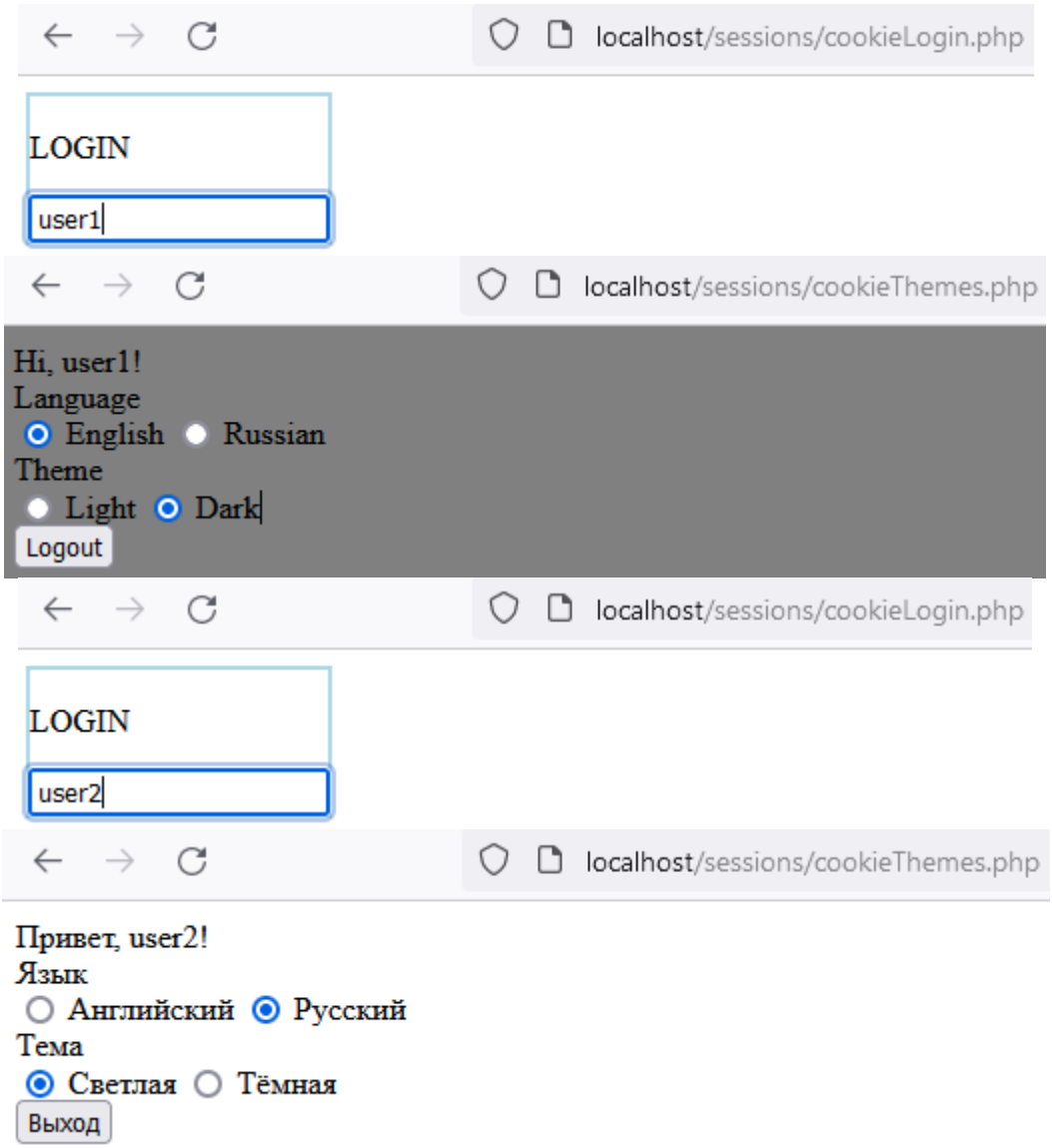

Рисунок 3 – Успешная проверка сохранения предпочтений

Реализуем загрузку и скачивание файлов с помощью файловой системы сервера (листинг 3).

Листинг 3 – Реализация загрузки и скачивания файлов

```
//index.html

//Форма, ответственная за загрузку файлов

<form action="upload.php" method="post"
enctype="multipart/form-data">
<input type="file" name="fileToUpload" id="fileToUpload">
<input type="submit" value="UPLOAD FILE" name="submit">
</form>

//Функция, ответственная за скачивание файлов

function DownloadEnter(e)
{
     if (e.keyCode == 13)
     {
          link.href = "http://localhost/files/" +
download.value;
          link.download = download.value;
          download.value = "";
          link.click();
     }
}

//upload.php

<?php
//Проверка, действительно ли файл - pdf (по содержимому)
if (file_get_contents($_FILES['fileToUpload']['tmp_name'],
false, null, 0, 5) == "%PDF-")
{
     move_uploaded_file($_FILES['fileToUpload']['tmp_name'],
"/var/www/html/files/".basename($_FILES["fileToUpload"]["name"
]));
     header('Location: index.html');
}
echo "ERROR: File is not PDF."
?>
```

Теперь попробуем загрузить ненастоящий и настоящий PDF файлы, а затем их скачать и посмотреть (рис. 4, 5).

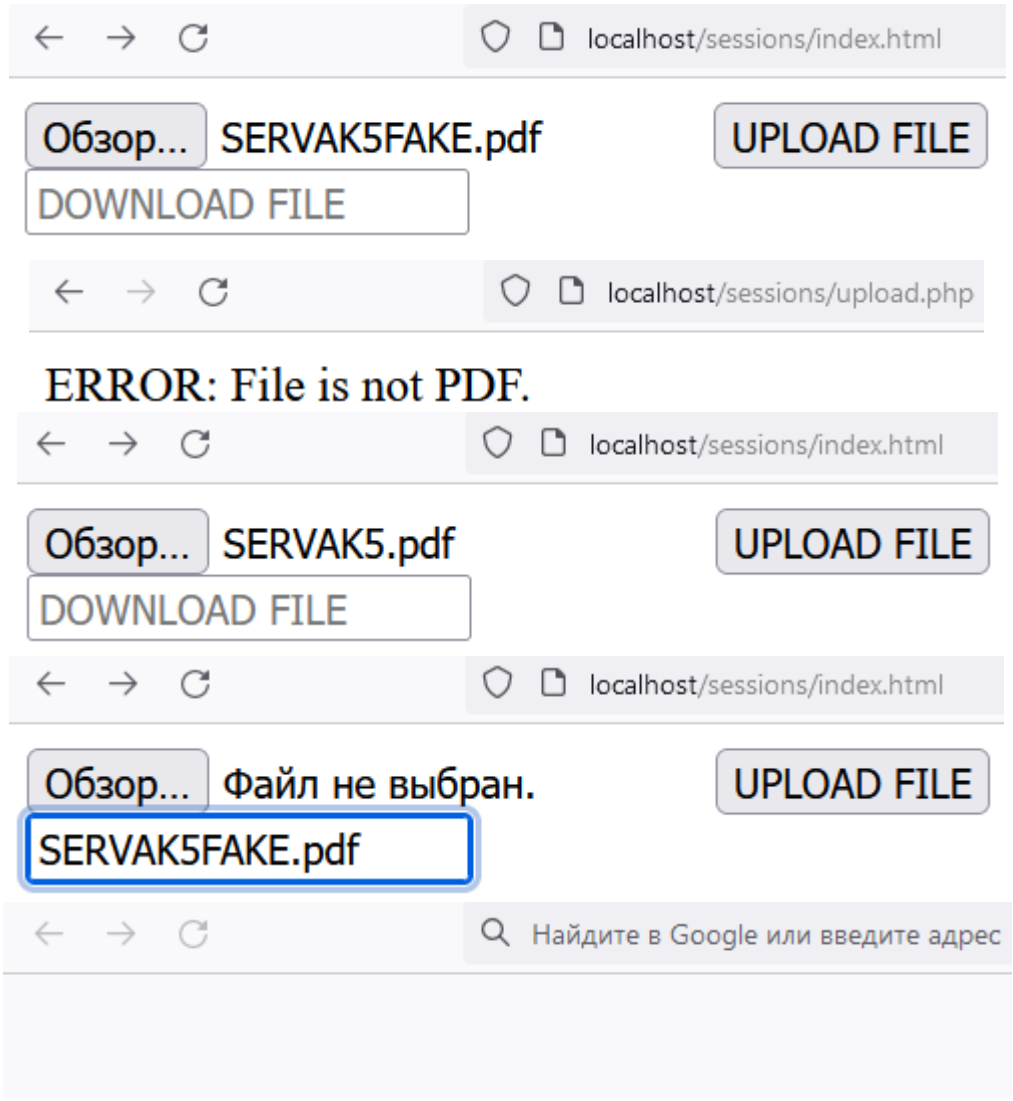

Рисунок 4 – Попытка загрузки файлов, попытка скачивания ненастоящего файла (получается пустота)

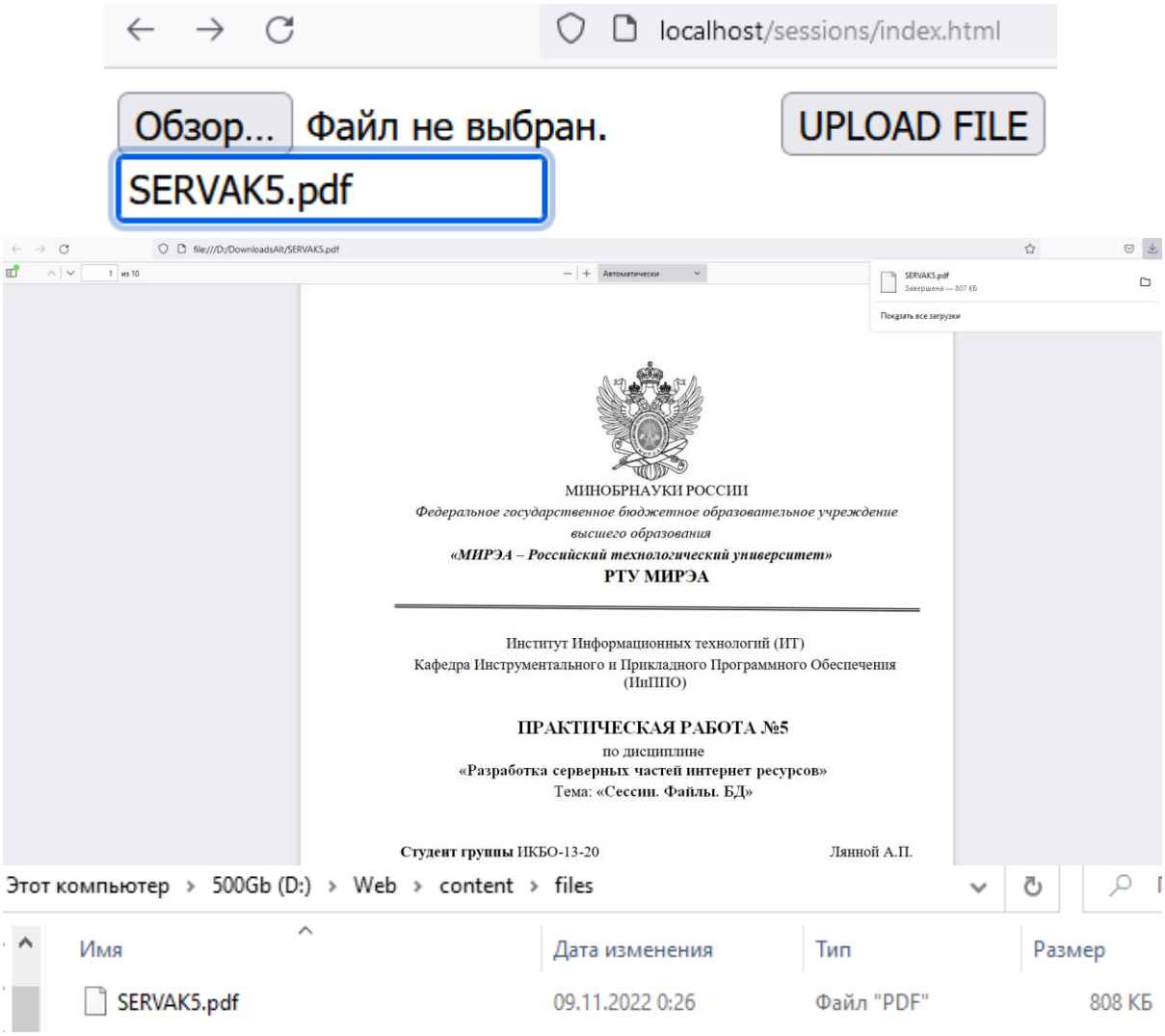

Рисунок 4 – Результат скачивания существующего файла, итоговое состояние файловой системы сервера

3. Вывод

Реализовано хранение данных сессий в БД Redis, использование данных cookie для согласования контента на уровне сервера, загрузка файлов в формате pdf на сервер и хранение их в файловой системе сервера, а также их выдача обратно пользователю по запросу.

4. Ответы на вопросы к практической работе

1. Что такое сессия в рамках веб-разработки?

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

2. 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.

3. Опишите механизм использования cookies.

Для взаимодействия с файлами Cookies в PHP существует массив $_COOKIE, содержащий данные в cookie файле. Для конфигурации самих cookie файлов необходимо описать настройки в php.ini.

4. Опишите простой пример работы сессий в PHP.

Листинг 1 – Модификация файлов docker'а

```
<?php
if(isset($_POST['login']))
$mysqli = new mysqli("db", "user", "password", "appDB");
$result = $mysqli->query("SELECT * FROM loginInfo WHERE
name='{$_POST['login']}'");
foreach ($result as $row){
if($row['password']==$_POST['password']){
session_start();
$_SESSION['user_authorized'] = 1;
header("Location: http://localhost/menu.php");
exit;
}
}
header("Location: http://localhost/index.php");
```

В листинге представлен код обработчика формы авторизации в очень упрощенном и небезопасном виде. 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.

5. Опишите способы защиты сессии пользователя.

Одним из вариантов повышение безопасности веб-приложения является неадаптивное управление сессиями. По умолчанию в PHP адаптивное управление сессиями, которые несут не всегда оправданные риски. Если session.use_strict_mode включён, и обработчик сохранения сессии это поддерживает, неинициализированный сессионный ID отвергается и создается новый. Это защищает от атак, которые принуждают пользователя использовать заранее известный ID.

Вторым вариантом поддерживающим вариантом является пересоздание идентификаторов сессий. Пересоздание идентификаторов сессий сильно уменьшает риск кражи сессии, соответственно надо на периодической основе запускать session_regenerate_id(). Например, пересоздавать идентификатор сессии каждые 15 минут для особо секретных данных. Даже если сессию украдут, она достаточно скоро станет истекшей и попытка её использовать приведёт к ошибке истекшей сессии. Идентификатор сессии должен пересоздаваться при повышении привилегий пользователя, например при аутентификации. Функция session_regenerate_id() должна вызываться до записи авторизационной информации в $_SESSION. (session_regenerate_id() сохраняет данные текущей сессии автоматически). Убедитесь, что только текущая сессия отмечена как авторизованная. Разработчики НЕ ДОЛЖНЫ полагаться на механизм истечения срока действия идентификатора сессии с помощью

session.gc_maxlifetime. Атакующие могут периодически получать доступ к сессии для предотвращения её срока действия и продолжать использовать идентификатор жертвы, включая аутентифицированные сессии.

Третьим методом защиты является удаление сессий. Данные истекших сессий должны быть недоступны и удалены. Существующий механизм управления сессиями делает это не очень хорошо. Данные истекших сессий надо удалять так быстро, как только возможно. С другой стороны, данные активных сессий НЕ ДОЛЖНЫ удаляться сразу же. Для обеспечения этих противоречивых требований, вы ДОЛЖНЫ самостоятельно реализовать механизм контроля за истекшими сессиями на базе временных меток. Устанавливайте и управляйте временными метками жизни сессии через $_SESSION. Запрещайте доступ к данным истекших сессий. Если обнаружена попытка доступа к данным устаревшей сессии, снимайте статус авторизации со всех активных сессий пользователя и вынуждайте его переавторизоваться. Доступ к данным истекшей сессии может означать атаку. Для обеспечения такого поведения вы должны отслеживать все активные сессии пользователя.

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

Можно, например, для этого целесообразно использовать нереляционную базу данных типа ключ-значение вроде Redis.

7. Опишите жизненный цикл сессии.

1. Открытие файла сеанса.
2. Закрытие файла сеанса.
3. Чтение данных сеанса.
4. Запись данных сеанса.
5. Уничтожение сессии.
6. Сборка мусора из файла сессии и данных.

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

Нет, так как, не обрабатывая сессии на сторонах и клиента, и сервера накопится куча мусорных неактуальных файлов, которые в том числе поставят под удар безопасность приложения.

9. Опишите примеры настройки сессии во время выполнения.

С помощью функции ini_set возможна настройка во время исполнения, но важно ее проводить до выполнения какой-либо функциональности. Например, ini_set('session.gc_maxlifetime', 5) установит максимальное время жизни сессии 5 секунд, после чего сессия удалится.

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

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.

11. 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. Выражаясь простыми понятиями, поток (stream) — это ресурс (resource), который ведёт себя как источник непрерывной последовательности данных. Это означает, что из потока можно последовательно читать данные, равно как и записывать в него.

12. Как открыть и закрыть файл с помощью PHP.

Открывается файл с помощью fopen(), закрывается с помощью fclose().

13. Как производится чтение и запись файлов в PHP.

Читается файл с помощью fread(), запись в файл происходит с помощью fwrite().

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

Часть файла можно считать с помощью file_get_contents() с указанием длины считывания, последовательно файл можно считывать той же функцией, указывая отступ, равный уже считанной длине, весь файл считывается той же функцией без указания того и другого.

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

Файл создаётся автоматически при попытке выполнить fread() или fwrite() на несуществующем файле, а удаление происходит с помощью unlink().

16. 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 отображение прав доступа в виде восьмеричного числа.

filesize возвращает размер файла в байтах.

filetype возвращает тип файла.

17. 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.

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

Сначала создается новый документ DOMDocument с параметром конструктора версией разметки xml/html, также вторым необязательным параметром является тип кодировки. Далее идет задание красивого форматированного вывода за счет изменения соответствующего свойства документа. Следующим шагом идет создание новых узлов документа. Документ содержит узлы, которые являются экземплярами класса DOMElement. Дальше идет добавление дочерних узлов в конец списка потомков с помощью метода appendChild.

19. Что такое JSON?

JSON (JavaScript Object Notation) — текстовый формат обмена данными, основанный на JavaScript. Как и многие другие текстовые форматы, JSON легко читается людьми. Несмотря на происхождение от JavaScript, формат считается независимым от языка и может использоваться практически с любым языком программирования. Для многих языков существует готовый код для создания и обработки данных в формате JSON.

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

Декодирование – json_decode(), кодирование – json_encode().

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

В силу своей простоты, JSON является неверным в случае несоответствия скобочного и кавычечного баланса, незаключения названий переменных в кавычки и неверного положения скобок.

22. Опишите создание, сохранение, парсинг XML-документа с помощью PHP.

Сначала создается новый документ DOMDocument с параметром конструктора версией разметки xml/html, также вторым необязательным параметром является тип кодировки. Далее идет задание красивого форматированного вывода за счет изменения соответствующего свойства документа. Следующим шагом идет создание новых узлов документа. Документ содержит узлы, которые являются экземплярами класса DOMElement. Дальше идет добавление дочерних узлов в конец списка потомков с помощью метода appendChild.

Далее возможен поиск по примеру всех тегов по имени (в примере book) с помощью метода getElementsByTagName, возвращающего итерируемый тип DOMNodeList.

23. Что такое драйвер в рамках взаимодействия с СУБД?

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

24. Опишите добавление записи в рамках использования модуля mysqli для взаимодействия с БД MYSQL.

Сначала необходимо подключиться к серверу баз данных, а точнее к конкретной базе с помощью mysqli_connect(). Далее следует сделать INSERT запрос с помощью mysqli_query().

25. Что такое постоянное соединение, опишите проблемы данного подхода и его решение в модуле mysqli.

Постоянное соединение отличается от обычного тем, что PHP сервер подключён к MYSQL серверу постоянно, а не подключается каждый раз, когда возникает необходимость. Чтобы открыть постоянное соединение в mysqli, необходимо перед именем хоста приписать ”p:”.

26. Опишите основные особенности БД MongoDB.

1. Это кроссплатформенная документоориентированная база данных NoSQL с открытым исходным кодом.

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

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

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

5. 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).

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

Для добавления в коллекцию могут использоваться три ее метода:

insertOne(json): добавляет один документ

insertMany(json): добавляет несколько документов

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

Чтобы получить базу данных, использовать getDatabase(), чтобы получить коллекцию, использовать getCollection(), чтобы получить документы, использовать find(), чтобы пройтись по данным, итерировать по документам циклом.

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

Сначала необходимо подключиться к серверу баз данных, а точнее к конкретной базе с помощью mysqli_connect(). Далее следует сделать SELECT запрос с помощью mysqli_query().

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

Чтобы поиск записей был ограниченным каким-то условием, необходимо использовать параметризированный find().

5. Ссылка на удалённый репозиторий проекта

https://github.com/Kvadr0n/NOT-JAVA

6. Список использованной литературы

1. Википедия - Redis [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Redis, свободный;

2. Википедия - Cookie [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Cookie, свободный.

3. Википедия – Сессия (веб-аналитика) [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Сессия_(веб-аналитика), свободный.