Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Институт информационных технологий

Факультет Компьютерных Технологий

Контрольная работа

По дисциплине «Веб технологии»

Выполнил: студент гр. 681072 Иванов К.Е

Проверила: Коренская И.Н.

Минск 2019

**Задание 1**

Рекомендации по выполнению задания:

1. Выполнять только в любой из операционных систем семейства Windows!

2. Необходимо развернуть виртуальную машину.

3. Установить программу «Macromedia Dreamweaver» (можно использовать любой удобный редактор).

4. Ваша задача – добиться того, чтобы созданный HTML-файл выглядел как можно более похож на рисунок 1.

Порядок выполнения задания:

1. Разработать HTML-страницу, соответствующую приведённому ниже рис. 1 (полное соответствие, как на рисунке!).

Рисунок 1 – Образец верстки

2. Рекомендуемый заголовок страницы:

<html>

<head>

<meta http-equiv="content-type" content="text/html;charset=windows-1251" />

<title>Задание - 1</title>

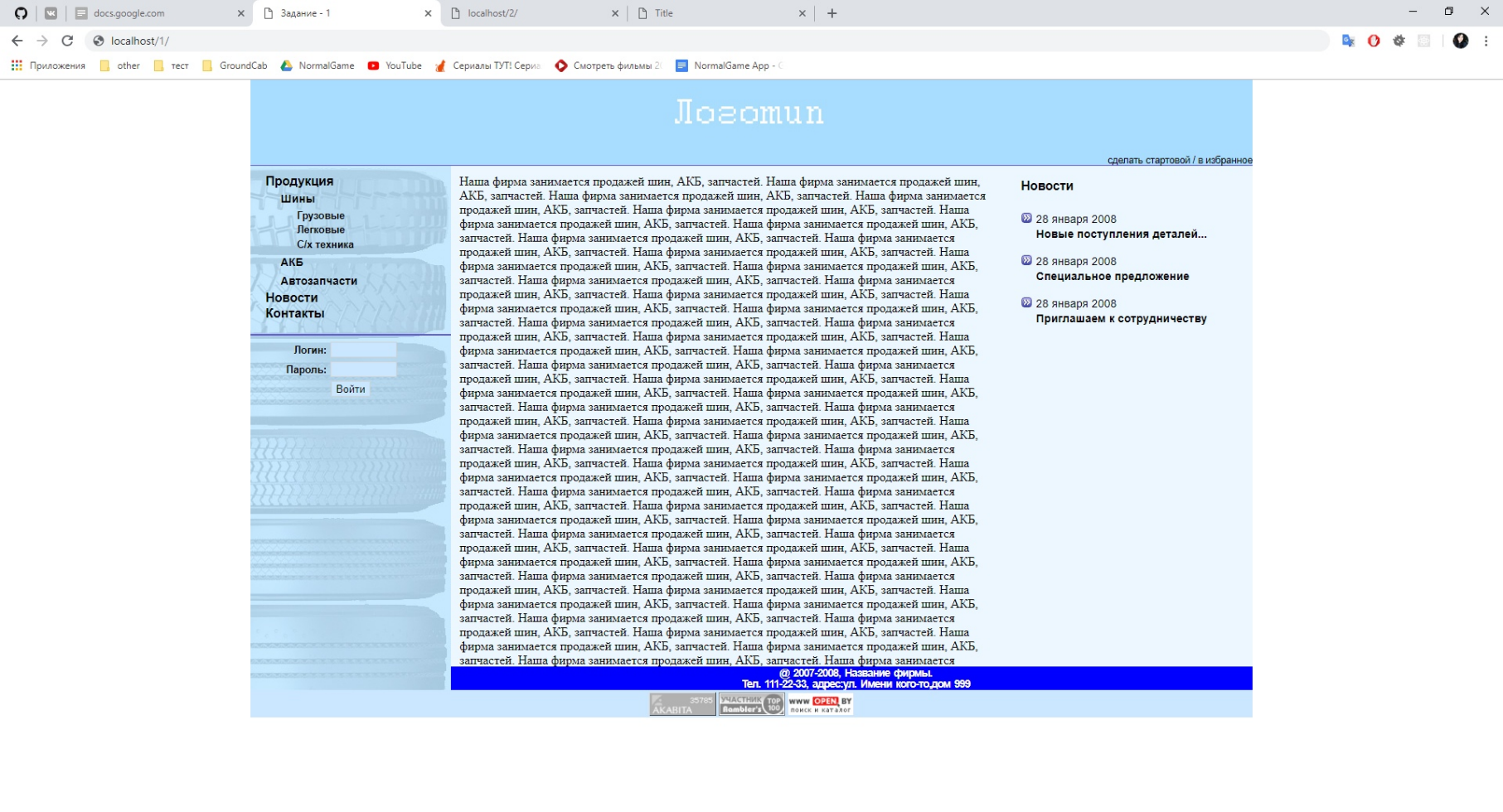
</head>

3. Необходимые графические элементы находятся в папке Graphics.

4. Использовать фреймы и каскадные таблицы стилей.

5. Рекомендуется использовать для выполнения только т.н. «блочную вёрстку».

**Результат:**



**Исходный код:**

Index.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Лабораторная работа - 1</title>

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<div class="container">

<header class="header">

<a href="#" class="in-favorite">

сделать стартовой / в избранное

</a>

</header>

<aside class="aside">

<nav class="menu">

<a href="#" class="main">Продукция</a>

<a href="#" class="sub">Шины</a>

<a href="#" class="type">Грузовые</a>

<a href="#" class="type">Легковые</a>

<a href="#" class="type">С/х техника</a>

<a href="#" class="sub">АКБ</a>

<a href="#" class="sub">Автозапчасти</a>

<a href="#" class="main">Новости</a>

<a href="#" class="main">Контакты</a>

</nav>

<form action="" class="form">

<div class="row">

<div class="half">Логин:</div>

<div class="half">

<input type="text">

</div>

</div>

<div class="row">

<div class="half">Пароль:</div>

<div class="half">

<input type="text">

</div>

</div>

<div class="row">

<button>Войти</button>

</div>

</form>

</aside>

<main class="main-wrap">

<div class="content">

<iframe src="" frameborder="0"></iframe>

<div class="down-line">

<p>

@ 2007-2008, Название фирмы.

</p>

<p>

Тел. 111-22-33, адрес:ул. Имени кого-то,дом 999

</p>

</div>

</div>

<div class="news">

<h2 class="title">Новости</h2>

<div class="item">

<div class="ico">

</div>

<div class="date">

28 января 2008

</div>

<a href="#" class="theme">

Новые поступления деталей...

</a>

</div>

<div class="item">

<div class="ico">

</div>

<div class="date">

28 января 2008

</div>

<a href="#" class="theme">

Специальное предложение

</a>

</div>

<div class="item">

<div class="ico">

</div>

<div class="date">

28 января 2008

</div>

<a href="#" class="theme">

Приглашаем к сотрудничеству

</a>

</div>

</div>

</main>

<footer class="footer">

<a href="#">

<img src="img/image1.bmp" alt="">

</a>

<a href="#">

<img src="img/image1.gif" alt="">

</a>

<a href="#">

<img src="img/image3.bmp" alt="">

</a>

</footer>

</div>

</body>

</html>

**Задание 2**

Рекомендации по выполнению задания:

1. На компьютере, например, с именем vsit в папке www\_pub создать свою папку с уникальным именем (например, mysuperfolder).

2. В этой папке создавать файлы с расширением php.

3. Для просмотра работы программы, допустим, создан файл 1.php, нужно в браузере открыть URL: <http://vsit/mysuperfolder/1.php>

4. Задание выполнить только в любой из операционных систем семейства Linux!

5. Необходимо развернуть виртуальную машину.

Порядок выполнения задания:

1. Вывести название и версию используемой ОС.

2. Объявить переменные следующих типов:

\* целочисленную,

\* строковую,

\* дробную,

\* логическую,

\* массив.

3. Объявить переменные $a=555 и $b="ZZZ" и сложить их:

\* как числа,

\* как строки.

Результат сложения не помещать в новую переменную, а сразу выводить на экран.

4. Есть три сотрудника:

Иванов, тел. 111-22-33, e-mail: [ivanov@domain.com](mailto:ivanov@domain.com)

Петров, тел. 112-24-36, e-mail: [petrov@domain.com](mailto:petrov@domain.com)

Сидоров, тел. 113-25-37, e-mail: [sidorov@domain.com](mailto:sidorov@domain.com)

Объявить двухмерный массив:

1. первый уровень которого пронумерован, начиная с нуля,

2. второй уровень содержит элементы name, phone, email, в которых хранятся соответствующие данные вышеназванных сотрудников.

Вывести массив по уровням, указывая ключ в []. После ключа вывести символ « =», после которого указать значение элемента.

5. Дан массив, содержащий элементы: 1, 2, "A", 3.764, 34, "B", 12. Объявить массив, вывести его, проанализировать содержимое и удалить из него все элементы, не являющиеся целыми или дробными числами.

6. Сгенерировать HTML-таблицу, состоящую из трёх колонок и 1000 строк. В первой колонке разместить номера строк таблицы. Цвет каждой строки таблицы должен изменяться по алгоритму: R+1, G+1, B+1, начиная с 000000: первая строка: 000000, вторая – 010101, третья -020202 и т.д. Замечание: числа – шестнадцатеричные и максимальное значение цвета составляет FFFFFF.

Дополнительное задание (по желанию).

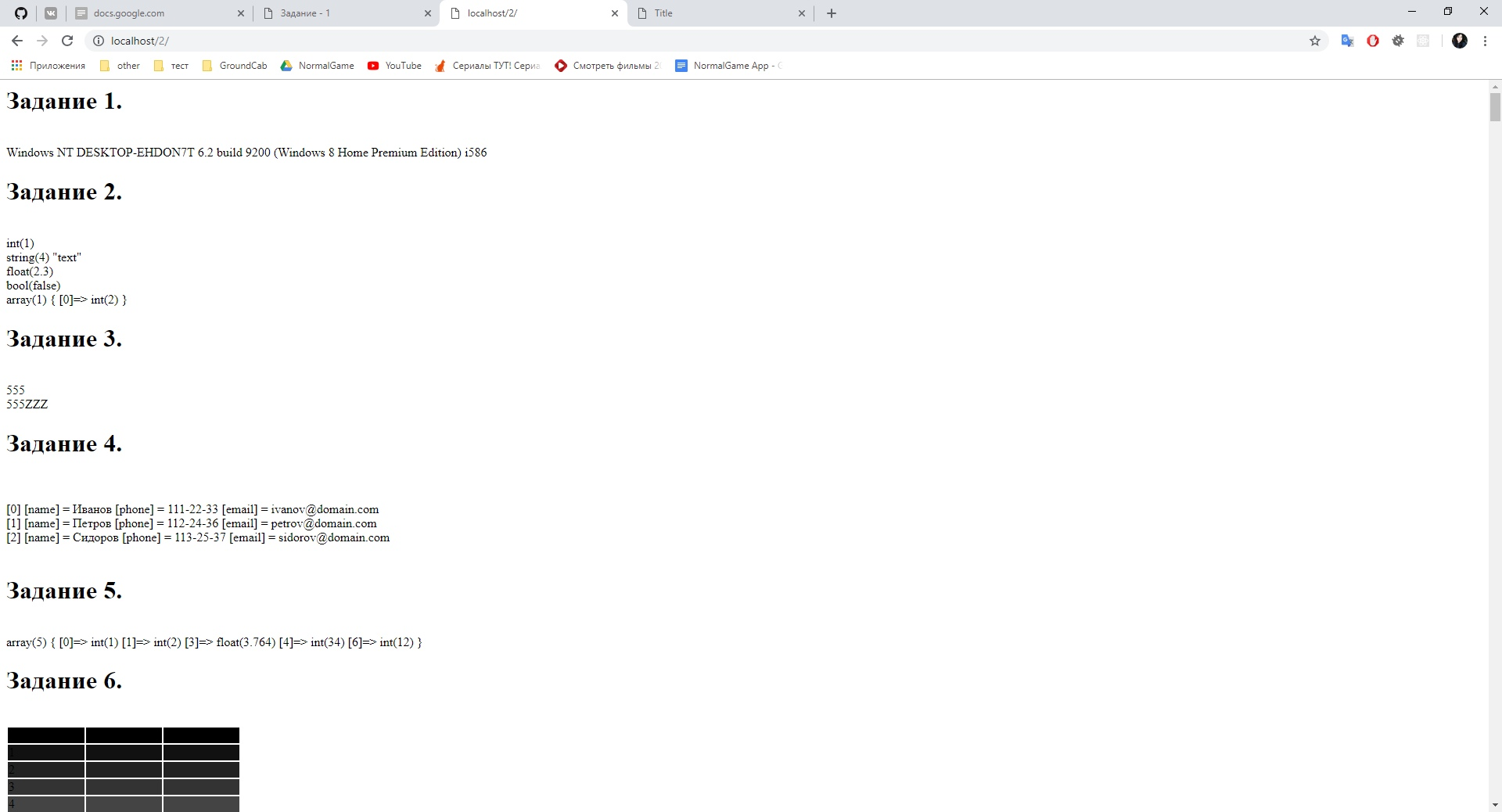
Создать HTML-форму произвольного вида (побольше полей) и написать скрипт, который получает данные из этой формы и сохраняет их в файл в виде:

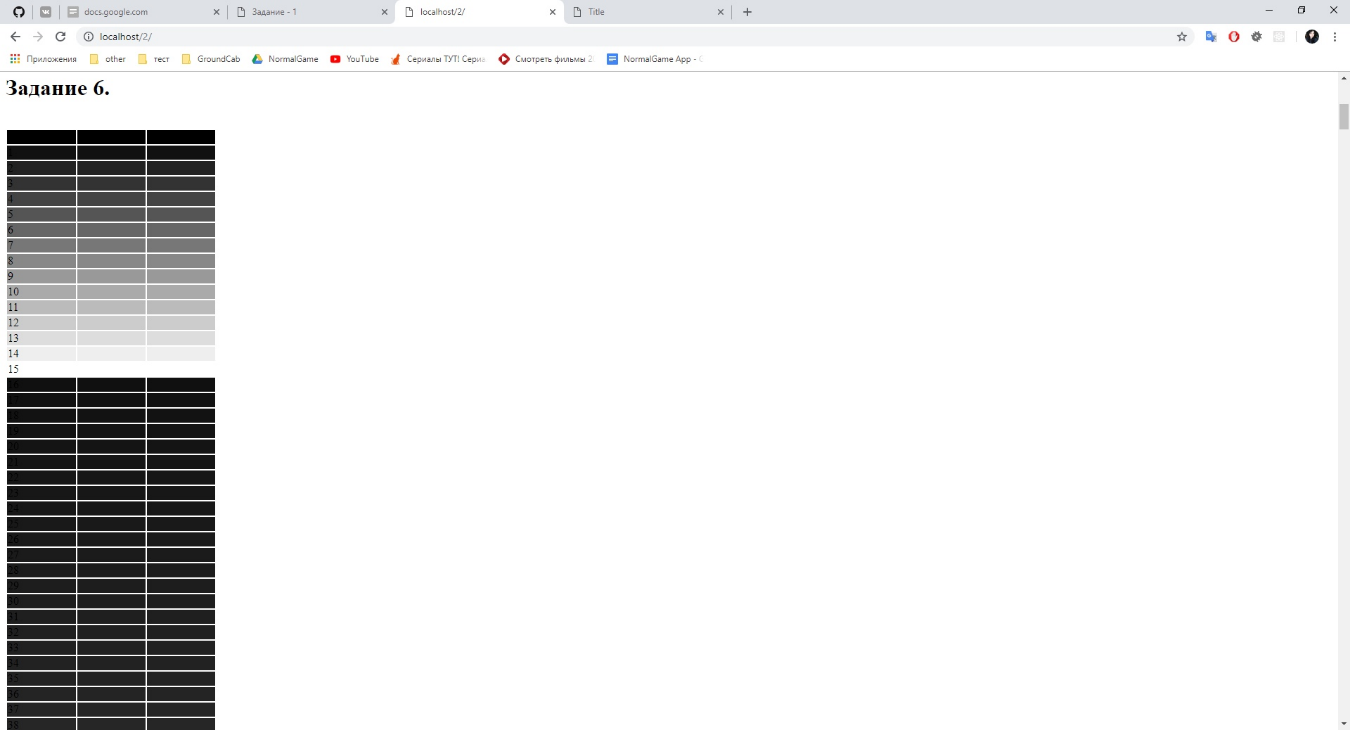
имя\_поля = "значение поля"

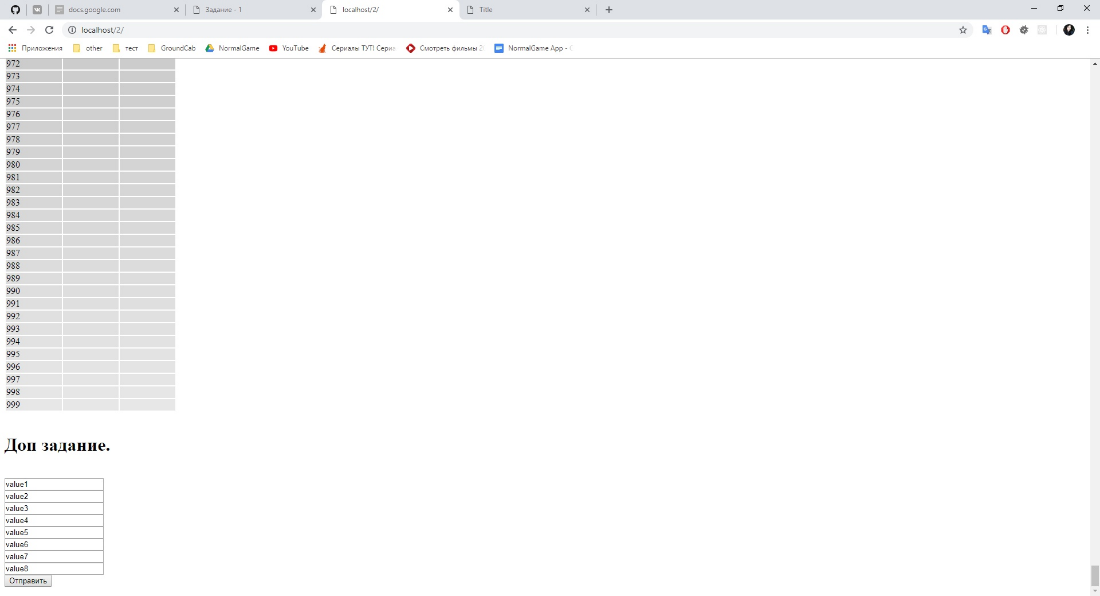
имя\_поля = "значение поля"

имя\_поля = "значение поля"

**Результат:**







**Исходный код:**

**Index.php:**

<?php

echo php\_uname();

$int = 0;

$string = "";

$double = 0.0;

$boolean = false;

$array = [];

echo "<br>";

$a = 555;

$b = 'ZZZ';

echo $a + $b;

echo "<br>";

echo $a . $b;

$employers = [

[

"name" => "Иванов",

"phone" => "111-22-33",

"email" => "[ivanov@domain.com](mailto:ivanov@domain.com)"

],

[

"name" => "Петров",

"phone" => "112-24-36",

"email" => "[petrov@domain.com](mailto:petrov@domain.com)"

],

[

"name" => "Сидоров",

"phone" => "113-25-37",

"email" => "[sidorov@domain.com](mailto:sidorov@domain.com)"

]

];

echo "<br>";

foreach ($employers as $index => $employer) {

echo "[$index] ";

foreach ($employer as $key => $info) {

echo "[$key] = $info; ";

}

echo "<br>";

}

$numbers = [1, 2, "A", 3.764, 34, "B", 12];

$numbers = array\_filter($numbers, function ($item) {

return is\_numeric($item);

});

echo '<pre>' . print\_r($numbers, true) . '</pre>';

const ROW\_COUNT = 1000;

const COL\_COUNT = 3;

echo '<table width="500">';

for ($row = 0; $row < ROW\_COUNT; $row++) {

echo "<tr style='background-color: #" . dechex($row % 256) . dechex($row % 256) . dechex($row % 256) . "'>";

for ($column = 0; $column < COL\_COUNT; $column++) {

echo "<td>" . (!$column ? $row : '') . "</td>";

}

echo "</tr>";

}

echo '</table>';

**Задание 3**

Рекомендации по выполнению задания:

1. На компьютере с именем, например, vsit в папке www\_pub создать свою папку с уникальным именем (например, mysuperfolder).

2. Скопировать в эту папку папки css, images, templates, а затем создать в ней файл index.php, в котором в дальнейшем писать код.

3. Чтобы посмотреть, как работает программа, надо в браузере открыть такой URL: <http://vsit/mysuperfolder/>

4. Все задания выполняются в одном файле index.php. Результатом работы является страница сайта, видимая в браузере.

5. ЖЁСТКОЕ ТРЕБОВАНИЕ: на весь скрипт index.php может быть ТОЛЬКО ОДИН оператор вывода echo.

6. Замечание 1: работу удобно выполнять с помощью функций.

7. Замечание 2: максимально используйте документацию по php.

8. Можно выполнять в любой ОС.

Порядок выполнения:

1. Прочитать содержимое шаблона main.tpl в текстовую переменную.

2. Плейсхолдер {MAIN\_MENU} заменить на содержимое шаблона main\_menu.tpl

3. Блоки:

а) с основным текстом страницы,

б) новостями,

в) областью с адресом и копирайтом

удалить из шаблона main.tpl, пометив места их расположения плейсхолдерами.

Содержимое удаляемых блоков разместить в отдельных шаблонах. Обеспечить замену плейсхолдеров содержимым соответствующих шаблонов (аналогично пункту 2).

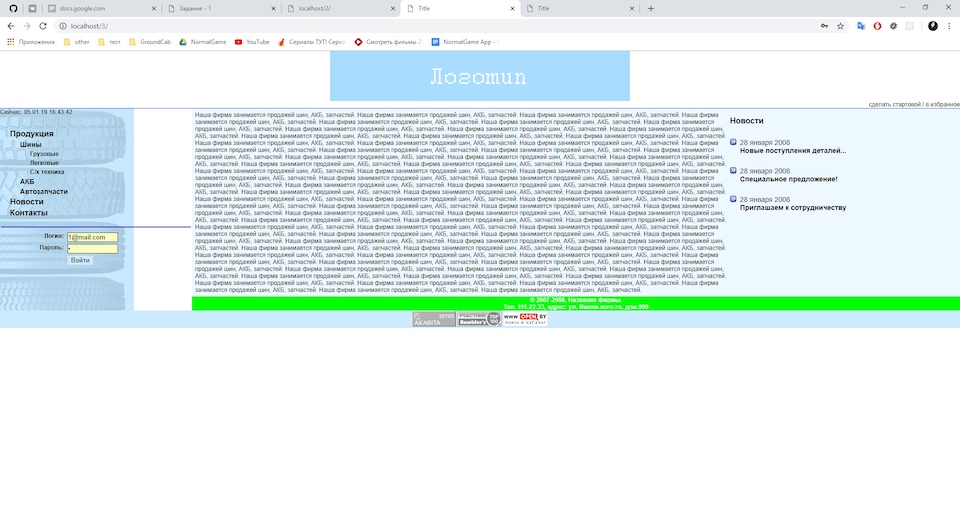
4. Плейсхолдеры {TODAY\_D}, {TODAY\_M}, {TODAY\_Y}, {NOW\_H}, {NOW\_M}, {NOW\_S} заменить на фрагменты текущей даты и времени: день, месяц, год, час, минута, секунда соответственно.

5. Создать конфигурационный файл site.cfg (формат разработать самостоятельно; примечание: делать этот файл в формате php ЗАПРЕЩЕНО, т.е. это должен быть "простой текстовый файл"), в котором разместить две переменные: main\_color и copyright\_color, в которых хранить (и использовать для управления сайтом) цвета области с основным текстом страницы и области с адресом и копирайтом.

6. Дополнительное задание (по желанию): разработать формат файла для хранения новостей, перенести туда все новости, написать функцию, формирующую список новостей на основе этого файла и шаблонов, определяющих отображение одной отдельной новости.

7. Дополнительное задание (по желанию): меню сайта расположить в отдельном файле (формат разработать самостоятельно) в виде иерархической структуры. Прочитать этот файл в многомерный массив. На основе полученного массива сформировать главное меню сайта.

**Решение:**



**Исходный код:**

index.php:

<?php

$main = file\_get\_contents("templates/main.tpl");

$config = json\_decode(file\_get\_contents("site.cfg"), true);

date\_default\_timezone\_set("Europe/Minsk");

$replace = [

"main\_menu" => file\_get\_contents("templates/main\_menu.tpl"),

"today\_d" => date("d"),

"today\_m" => date("m"),

"today\_y" => date("y"),

"now\_h" => date("H"),

"now\_m" => date("i"),

"now\_s" => date("s"),

"news" => file\_get\_contents("templates/news.tpl"),

"main\_text" => file\_get\_contents("templates/main\_text.tpl")

];

$replace += $config;

$main = str\_replace(array\_map(function($item){return "{".strtoupper($item)."}";}, array\_keys($replace)), $replace, $main);

echo $main;

**Задание 4**

Рекомендации по выполнению задания:

0. Продолжаем работать с результатами, полученными в задании 3. Соответственно, см. Задание 3 на предмет того, что и куда копировать, как настраивать и т.д.

1. Все задания выполняются в одном файле index.php. Результатом работы является страница сайта, видимая в браузере.

2. ЖЁСТКОЕ ТРЕБОВАНИЕ: на весь скрипт index.php может быть ТОЛЬКО ОДИН оператор вывода echo.

3. Рекомендация 1: эту работу удобно выполнять с помощью функций.

4. Рекомендация 2: максимально используйте документацию по php.

5. Рекомендуется использовать средство PhpNyAdmin для упрощения процесса создания БД. Оно доступно по URL'у: <http://vsit/phpmyadmin/>

6. Можно выполнять в любой ОС на языках Java, JavaScript или php.

Порядок выполнения:

1. Создать БД с уникальным именем.

2. В созданной БД создать таблицу news, содержащую поля:

\* уникальный идентификатор новости,

\* дата публикации новости,

\* заголовок новости,

\* текст новости.

3. Написать функцию, отображающую на сайте N самых свежих новостей (N задаётся как параметр функции). Новости сортируются по убыванию даты. Рекомендуется для выполнения задания использовать шаблоны.

4. В созданной БД создать таблицу pages, содержащую поля:

\* уникальный идентификатор страницы,

\* информация о родительской странице,

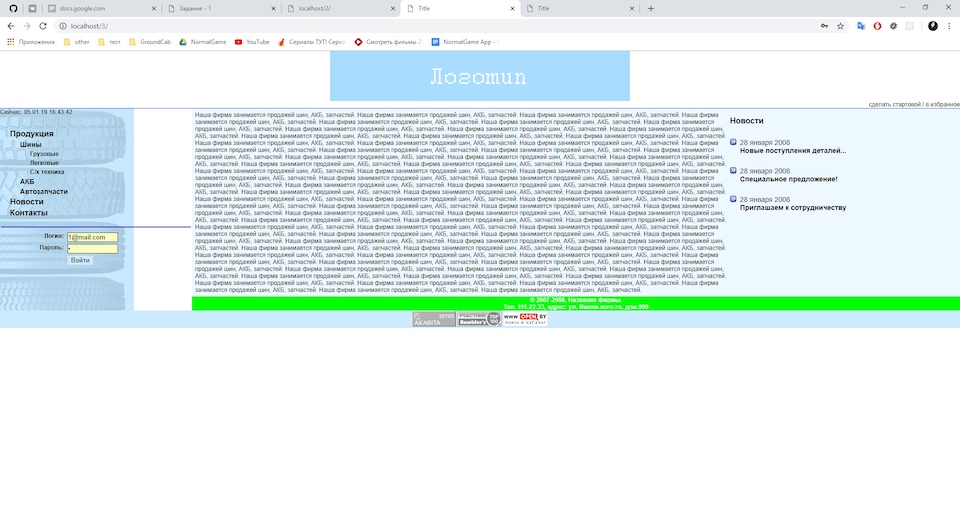
\* текст для отображения в меню.

5. На основе таблицы, разработанной в пункте 4, сформировать меню сайта. Меню должно иметь не менее трёх уровней вложенности. По наведению курсора мыши или при клике по пункту меню должен открываться ("разворачиваться") его первый подуровень, остальные пункты меню при этом находятся в "свёрнутом" состоянии.

6. Дополнительное задание (по желанию): добавить в таблицу pages поле «текст страницы» и модифицировать свой скрипт так, чтобы по клику на пункт меню в основной области отображался текст соответствующей страницы.

7. Дополнительное задание (по желанию): создать таблицу для хранения информации о пользователях сайта. Получить информацию из полей «Логин» и «Пароль» (форма на странице уже есть) и, если пользователь ввёл верные данные, заменить форму на текст в виде «Здравствуйте, ИМЯ\_ПОЛЬЗОВАТЕЛЯ!». Используя механизм сессий, добиться того, чтобы пользователь оставался «залогиненным» при переходе между различными страницами сайта.

**Решение:**



**Исходный код:**

Index.php:

<?php

header('Content-type: text/html; charset=utf-8');

date\_default\_timezone\_set("Europe/Minsk");

include\_once "ViewHelper.php";

$viewHelper = new ViewHelper();

echo $viewHelper->getTemplate();

DBHelper.php:

<?php

class DBHelper

{

private $mysql;

public function \_\_construct()

{

$db = json\_decode(file\_get\_contents("db.cfg"), true);

$this->mysql = new mysqli($db['host'], $db['user'], $db['password'], $db['database']);

$this->mysql->set\_charset("utf8");

if ($this->mysql->connect\_error) {

die('Connection error');

}

}

public function getNews($count = 3)

{

$count = (int)$count ? (int)$count : 3;

$result = $this->mysql->query("SELECT \* FROM news ORDER BY date DESC LIMIT $count");

$news = [];

while ($data = $result->fetch\_assoc()) {

$news[] = $data;

}

return $news;

}

public function getMenu()

{

$menu = [];

$result = $this->mysql->query("SELECT \* FROM pages");

while ($data = $result->fetch\_assoc()) {

$menu[] = $data;

}

return $this->formatMenu($menu);

}

private function formatMenu($menu, $parentId = null)

{

$result = [];

foreach ($menu as $menuItem) {

if ($parentId == $menuItem["parent\_id"]) {

$result[] = $menuItem;

if ($submenu = $this->formatMenu($menu, $menuItem['id'])) {

$result[count($result) - 1]['submenu'] = $submenu;

}

}

}

return $result;

}

}

**Задание 5**

Рекомендации к выполнению задания:

1. Максимально используйте документацию по php или Java и думайте.

2. Можно выполнять в любой ОС на языках php, JavaScript или Java.

Порядок выполнения задания:

1. Написать регулярное выражение, проверяющее, является ли строка корректным e-mail адресом.

Примеры корректных адресов:

[vasya-pupkin@mail.com](mailto:vasya-pupkin@mail.com)

[vasya\_pupkin@mail.com](mailto:vasya_pupkin@mail.com)

[vasya.pupkin@mail.com](mailto:vasya.pupkin@mail.com)

[v.v.pupkin@firma.mail.com](mailto:v.v.pupkin@firma.mail.com)

[v.v.pupkin@firma-mail.com](mailto:v.v.pupkin@firma-mail.com)

[v.v.pupkin12@firma\_mail.com](mailto:v.v.pupkin12@firma_mail.com)

[v.v.pupkin-director@firma.mail.com](mailto:v.v.pupkin-director@firma.mail.com)

Примеры некорректных адресов:

[-vasya--pupkin@mail.com](mailto:-vasya--pupkin@mail.com)

[vasya\_pupkin@mail..com](mailto:vasya_pupkin@mail..com)

[vasya.-pupkin@mail.com](mailto:vasya.-pupkin@mail.com)

v.v.pup [kin@firma.mail.com](mailto:kin@firma.mail.com)

[v.v.pup#kin@firma-mail.com\_](mailto:v.v.pup#kin@firma-mail.com_)

2. Написать регулярное выражение, удаляющее из текста HTML-комментарии, т.е. конструкции вида

<!-- комментарий -->

Конструкции вида:

<!--[if IE]>

<link href="/css/invstroyIEfix.css" rel="stylesheet" type="text/css" />

<![endif]-->

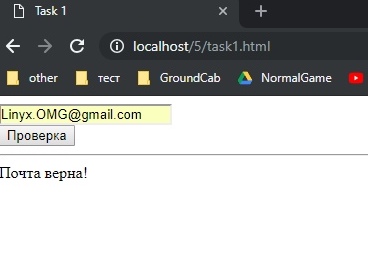
удалять не надо.

3. Написать регулярное выражение, очищающее текст от HTML-тегов.

4. Написать регулярное выражение, выделяющее красным жирным шрифтом все слова с длиной в 5 и более символов, состоящие только из заглавных букв.

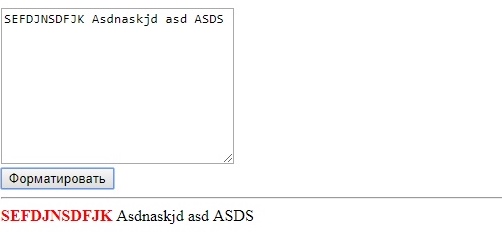
5. Дополнительное задание (по желанию): написать скрипт, представляющий HTML-документ в виде дерева элементов, которое хранится в массиве.

**Решение:**









**Исходный код:**

task1.html:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>Task 1</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<script src="task1.js"></script>

</head>

<body>

<input type="text" value="" id="email">

<br>

<button id="check">Check valid</button>

<br>

<hr>

<span id="result"></span>

<br>

</body>

</html>

task2.html:

<html>

<head>

<meta charset="utf-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>Task 2</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<script src="task2.js"></script>

</head>

<body>

<button id="remove-comments">Remove comments</button>

<xmp id="before">

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<!--[if IE]>

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<![endif]-->

<title>Page Title</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" type="text/css" media="screen" href="main.css" />

<script src="main.js"></script>

</head>

<body>

<!--comment1-->

<!-- comments2 -->

<!--

multiline comment

-->

</body>

</html>

</xmp>

<xmp id="after"></xmp>

</body>

</html>

**Задание 6**

**Рекомендации по выполнению:**

1. На компьютере, например, с именем **vsit** в папке **www\_pub** создать свою папку с уникальным именем (например, **mysuperfolder**).
2. В этой папке создавать файлы с расширением **php**.
3. Для просмотра работы программы, допустим, создан файл 1.php, нужно в браузере открыть URL: **http://vsit/mysuperfolder/1.php**
4. Максимально используйте документацию по php и думайте.

**Порядок выполнения задания:**

1. Представить новости (см. задание 4) в формате RSS.

Пример представления:

<?xml version="1.0" encoding="windows-1251" ?>

<rss version="2.0">

<channel>

<title>**Название ленты новостей**</title>

<link>**http://www.somesite.com/rss/**</link>

<description>**Описание ленты новостей**</description>

<lastBuildDate>**Sun, 18 Jan 2009 16:49:01 +0200**</lastBuildDate>

<item>

<title>**Заглавие новости (дата публикации новости)**</title>

<link>**http://www.somesite.com/news/newsid4792/**</link>

<description>**Краткое содержание новости**</description>

<comments>**http://www.somesite.com/news/newsid4792/**</comments>

<pubDate>**Fri, 16 Jan 2009 21:09:01 +0200**</pubDate>

<guid>**http://www.somesite.com/news/newsid4792/**</guid>

</item>

……………………………………….

</channel>

</rss>

Где:

**Название ленты новостей** – произвольное название (например,. "Наши новости").

**http://www.somesite.com/rss/** – URL ленты новостей. Можете взять значение из примера.

**Описание ленты новостей** – произвольное описание (например,. "Самые свежие в мире новости у нас!")

**Sun, 18 Jan 2009 16:49:01 +0200** – дата обновления ленты новостей. Генерируйте на основе текущей даты и времени в таком же формате.

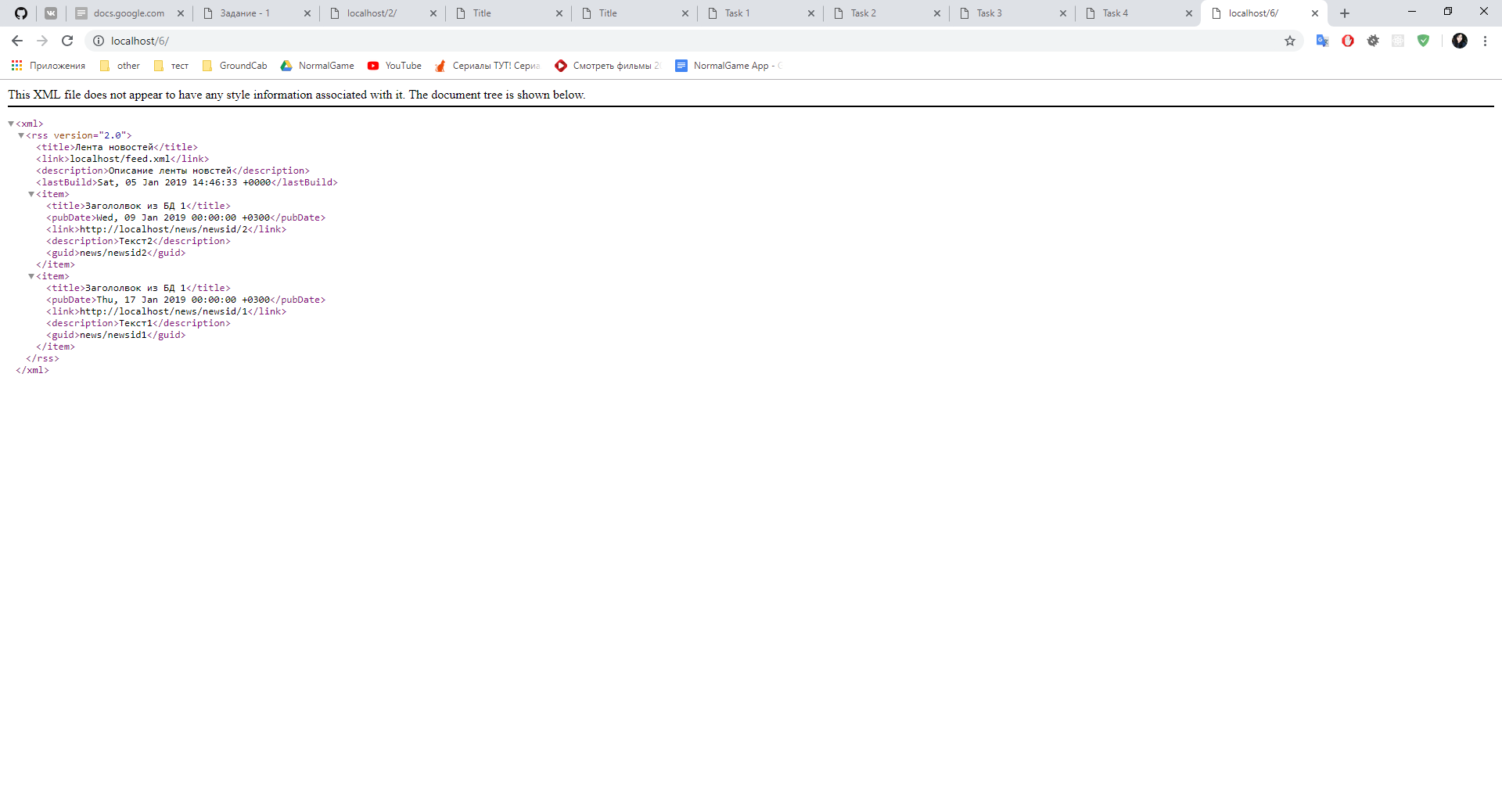
**Заглавие новости (дата публикации новости)** – заглавие и дата публикации новости в произвольном формате (например, "Скоро каникулы! (01.01.2018)")

**http://www.somesite.com/news/newsid4792/** - ссылка на полный текст новости. Нужно менять только идентификатор (помечено красным). Остальное значение можно просто переписать из примера.

**Fri, 16 Jan 2009 21:09:01 +0200** – дата публикации новости в ленте. Генерируйте на основе даты-времени внесения новости в БД в таком же формате.

1. Дополнительное задание (по желанию): написать скрипт, анализирующий RSS-ленту, извлекающий оттуда новости и размещающий их в БД.

**Решение:**



**Исходный код:**

**Index.php:**

<?php

header("Content-Type: text/xml");

header("Expires: Thu, 19 Feb 1998 13:24:18 GMT");

header("Last-Modified: " . gmdate("D, d M Y H:i:s") . " GMT");

header("Cache-Control: no-cache, must-revalidate");

header("Cache-Control: post-check=0,pre-check=0");

header("Cache-Control: max-age=0");

header("Pragma: no-cache");

$db = json\_decode(file\_get\_contents("db.cfg"), true);

$mysql = new mysqli($db['host'], $db['user'], $db['password'], $db['database']);

$mysql->set\_charset("utf8");

$news = [];

$result = $mysql->query("SELECT

title,

date,

CONCAT('http://" . $\_SERVER["HTTP\_HOST"] . "/news/newsid/', id) AS link,

body AS description,

CONCAT('news/newsid', id) AS guid

FROM news ORDER BY date ASC");

while ($data = $result->fetch\_assoc()) {

$news[] = $data;

}

$xml = new SimpleXMLElement("<xml/>");

$rss = $xml->addChild("rss");

$rss->addAttribute("version", "2.0");

$rss->addChild("title", "Лента новостей");

$rss->addChild("link", $\_SERVER["HTTP\_HOST"] . "/feed.xml");

$rss->addChild("description", "Описание ленты новстей");

$rss->addChild("lastBuild", gmdate("r"));

foreach ($news as $newsItem) {

$item = $rss->addChild("item");

foreach ($newsItem as $key => $newsPartItem) {

if ($key == "date") {

$date = (new DateTime($newsPartItem))->format("r");

$item->addChild("pubDate", $date);

} else {

$item->addChild($key, $newsPartItem);

}

}

}

echo $xml->asXML();