МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «ВИТЕБСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ П.М. МАШЕРОВА»

Факультет математики и информационных технологий

Кафедра информатики и информационных технологий

КУРСОВАЯ РАБОТА  
 по дисциплине «Дополнительные главы информатики»

СОВРЕМЕННЫЕ WEB-ТЕХНОЛОГИИ ДЛЯ СОЗДАНИЯ РАСПРЕДЕЛЁННЫХ ПРИЛОЖЕНИЙ

Моисейченков Дмитрий Юрьевич

4 курс, 44 группа

Руководитель:

Шедько Василий Викторович

старший преподаватель

Витебск, 2019

**Реферат**

Курсовая работа 26 с., 5 рис., 6 источников.

WEB-ТЕХНОЛОГИИ ПРИЛОЖЕНИЕ ИНТЕРНЕТ-МАГАЗИН

Объект исследования – методы и технологии создания распределённых приложений.

Предмет исследования − реализация в программировании методов и технологий создания распределённых приложений.

Цель проекта – ознакомиться с различными методами, технологиями по созданию распределённых приложений, сделать сравнительную характеристику данных методов и технологий.

Методы исследования: анализ, синтез, изучение литературы, практическая реализация.

СОДЕРЖАНИЕ

[Введение 4](#_Toc7628895)

[1.Понятие базы данных 6](#_Toc7628896)

[1.1 Определение базы данных 6](#_Toc7628897)

[1.2 Реляционная система управления базами данных MySQL 6](#_Toc7628898)

[2. Программное проектирование интерактивного новостного сайта 8](#_Toc7628899)

[3. Практическая реализация интерактивного новостного сайта 9](#_Toc7628900)

[3.1 Создание базы данных 10](#_Toc7628901)

[3.2 Создание шапки сайта 11](#_Toc7628902)

[3.3 Подключение к базе данных 12](#_Toc7628903)

[3.4 Создание главной страницы сайта, вывод постов на главную страницу 12](#_Toc7628904)

[3.5 Создание файла subscribe.php для записи в базу данных новых подписчиков и файла post.php для просмотра новостных постов в полном их объёме. 15](#_Toc7628905)

[3.6 Создание фильтрации постов по разделам новостного сайта 17](#_Toc7628906)

[4. Тестирование интерактивного новостного сайта 20](#_Toc7628907)

[4.1 Результаты тестирования 22](#_Toc7628908)

[Заключение 23](#_Toc7628909)

[Список использованных источников 24](#_Toc7628910)

[ПРИЛОЖЕНИЕ А 25](#_Toc7628911)

# Введение

Современное информационное поле состоит из массы событий, объектов и явлений. Открытое программное обеспечение стало основным структурным элементом при создании некоторых крупнейших веб-сайтов. С началом стремительного роста этих веб-сайтов возникли практические передовые методы и основные руководящие принципы их архитектуры. Данная курсовая работа стремится охватить некоторые ключевые вопросы, которые следует учитывать при проектировании больших веб-сайтов, интернет-магазинов, а также некоторые базовые компоненты, используемые для достижения этих же целей.

Что именно означает создание и управление масштабируемым веб-сайтом или приложением? На примитивном уровне это просто соединение пользователей с удаленными ресурсами через Интернет. А ресурсы или доступ к этим ресурсам, которые рассредоточены на множестве серверов и являются звеном, обеспечивающим масштабируемость веб-сайта.

Как большинство вещей в нашей жизни, время, которое потрачено заранее на планирование и построение веб-службы, может помочь в дальнейшем; понимание некоторых компромиссов, стоящих позади больших веб-сайтов, может принести плоды в виде более умных решений при создании меньших веб-сайтов. Ниже рассмотрены самые ключевые принципы, влияющие на проектирование крупных веб-систем:

* Доступность: длительность работоспособного состояния веб-сайта критически важна по отношению к функциональности и репутации многих компаний. Для некоторых наиболее крупных розничных онлайн-магазинов, недоступность, даже в течение нескольких минут, может привести к тысячам или миллионам долларов потерянного дохода. Таким образом, разработка постоянно доступных и эластичных к отказу систем, в свою очередь, является и фундаментальным, и главным технологическим требованием. Высокая доступность в распределенных системах требует внимательного рассмотрения избыточности для ключевых компонентов, быстрого восстановления после частичных системных отказов и сглаженного сокращения возможностей при возникновении небольших проблем в работе.
* Производительность: Производительность любого веб-сайта стала очень важным показателем для большинства современных сайтов. Скорость веб-сайта влияет на работу и удовлетворенность пользователей, а также ранжирование поисковыми системами (Google, Яндекс) — фактор, который непосредственно влияет на удержание аудитории и доход сайта. В результате, ключом является создание системы, которая оптимизирована для быстрых ответов и низких задержек.
* Надежность: система должна быть действительно надежной, так, чтобы определенный запрос на получение данных единообразно возвращал определенные, нужные в данную секунду, данные. В случае изменения данных или обновления, то тот же запрос должен возвращать уже совершенно новые данные. Пользователи должны знать, если что-то записано в систему или храниться в ней, то можно быть уверенным, что оно и будет оставаться на своем месте для возможности извлечения данных впоследствии следующего использования сайта в своих целях.
* Масштабируемость: Когда дело доходит до любой крупной распределенной системы, размер оказывается всего лишь одним пунктом из целого списка, который необходимо учитывать. Не менее важным являются усилия, направленные на увеличение пропускной способности для обработки больших объемов нагрузки, которая обычно и называется «масштабируемостью системы». Масштабируемость может относиться к различным параметрам системы: количество выделяемого дополнительного трафика, с которым она без проблем сможет справиться, насколько легко можно нарастить ёмкость запоминающего устройства, или насколько много транзакций может быть обработано в определённый момент времени.
* Управляемость: проектирование системы, которая проста в эксплуатации также еще один важный фактор. Управляемость системы можно приравнять к масштабируемости операций «обслуживание» и «обновления». Для обеспечения управляемости необходимо рассмотреть вопросы простоты диагностики и понимания возникающих проблем, легкости проведения обновлений или модификаций, прихотливости системы в эксплуатации. (То есть, работает ли она как положено без отказов или исключений?)
* Стоимость: Стоимость является важным фактором. Она, очевидно, может включать в себя расходы на аппаратное и программное обеспечение, однако важно также рассмотреть и другие аспекты, необходимые для развертывания и поддержания системы в работе. Можно выделить основные: количество времени разработчиков, требуемое для построения системы, объем оперативных усилий, необходимые для запуска системы и даже достаточный уровень обучения — все должно быть предусмотрено на этапе запуска. Стоимость же будет представлять собой общую стоимость владения.

# 1.Программное обеспечение по технологии RAD

Жизненный цикл ПО по методологии RAD состоит из четырех фаз:

* фаза анализа и планирования требований;
* фаза проектирования;
* фаза построения;
* фаза внедрения.

На фазе анализа и планирования требований пользователи системы определяют основные функции, которые она должна будет выполнять, выделяют наиболее приоритетные из них, описывают информационные потребности. Ограничивается масштаб проекта, определяются временные рамки для каждой из последующих фаз. Кроме того, определяется сама возможность реализации данного проекта в установленных рамках финансирования, на данных аппаратных средствах и т.п. Результатом данной фазы должны быть список и приоритетность функций будущей информационной системы, предварительные функциональные и информационные модели информационных систем.

На фазе проектирования часть пользователей принимает участие в техническом проектировании системы под руководством специалистов-разработчиков. CASE-средства используются для быстрого получения работающих прототипов приложений. Пользователи, непосредственно взаимодействуя с ними, уточняют и дополняют требования к системе, которые не были выявлены на предыдущей фазе. Более подробно рассматриваются процессы системы. Анализируется и, при необходимости, корректируется функциональная модель. Каждый процесс рассматривается детально. При необходимости для каждого элементарного процесса создается частичный прототип: экран, диалог, отчет, устраняющий неясности или неоднозначности. Определяются требования разграничения доступа к данным. На этой же фазе происходит определение набора необходимой документации.

После детального определения состава процессов оценивается количество функциональных элементов разрабатываемой системы и принимается решение о разделении информационных систем на подсистемы, поддающиеся реализации одной командой разработчиков за приемлемое для RAD-проектов время - порядка 60 - 90 дней. С использованием CASE-средств проект распределяется между различными командами (делится функциональная модель). Результатом данной фазы должны быть:

* общая информационная модель системы;
* функциональные модели системы в целом и подсистем, реализуемых отдельными командами разработчиков;
* точно определенные с помощью CASE-средства интерфейсы между автономно разрабатываемыми подсистемами;
* построенные прототипы экранов, отчетов, диалогов.

Все модели и прототипы должны быть получены с применением тех CASE-средств, которые будут использоваться в дальнейшем при построении системы.

Данное требование вызвано тем, что в традиционном подходе при передаче информации о проекте с этапа на этап может произойти фактически неконтролируемое искажение данных. Применение единой среды хранения информации о проекте позволяет избежать этой опасности.

## 1.1 Определение базы данных

База данных − это совокупность структур, предназначенных для хранения больших объемов информации и программных модулей, осуществляющих управление данными, их выборку, сортировку и другие подобные действия. Информация базы данных хранится в одной или нескольких таблицах. Любая таблица с данными состоит из набора однотипных записей, расположенных друг за другом. Они представляют собой строки таблицы, которые можно добавлять, удалять или изменять. Каждая запись является набором именованных полей, или ячеек, которые могут хранить самую разнообразную информацию, начиная от даты рождения и заканчивая подробным описанием кулинарного рецепта. Однотипные поля разных записей образуют столбец таблицы. Создав одну таблицу, вы уже получаете полноценную базу данных. Однако в реальной жизни структуры баз данных, а соответственно и способы их создания, намного сложнее.

## 1.2 [Реляционная система управления базами данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94) MySQL

Не все базы данных создаются на основе одних и тех же принципов, но традиционно в них применяется идея организации данных в виде записей. Каждая запись имеет фиксированный набор полей. Записи помещаются в таблицы, а совокупность таблиц формирует базу данных.

Для работы с базой данных необходима СУБД (система управления базами данных), т.е. программа, которая берет на себя все заботы, связанные с доступом к данным. Она содержит команды, позволяющие создавать таблицы, вставлять в них записи, искать и удалять таблицы.

MySQL - это быстрая, надежная, открыто распространяемая СУБД. MySQL, как и многие другие СУБД, функционирует по модели "клиент/сервер". Под этим подразумевается сетевая архитектура, в которой компьютеры играют роли клиентов либо серверов [1].

Строки таблиц могут быть связаны друг с другом одним из трех способов. Простейшее отношение — "один к одному". В этом случае строка первой таблицы соответствует одной единственной строке второй таблицы. На диаграммах такое отношение выражается записью 1:1.

Отношение "один ко многим" означает ситуацию, когда строка одной таблицы соответствует нескольким строкам другой таблицы. Это наиболее распространенный тип отношений. На диаграммах он выражается записью 1:N.

СУБД обеспечивает безопасность данных. Пользователям предоставляются определенные права доступа к информации. Некоторым пользователям разрешено лишь просматривать данные, тогда как другие пользователи могут менять содержимое таблиц.

СУБД поддерживает параллельный доступ к базе данных. Приложения могут обращаться к базе данных одновременно, что повышает общую производительность системы. Кроме того, отдельные операции могут "распараллеливаться" для еще большего улучшения производительности.

Наконец, СУБД помогает восстанавливать информацию в случае непредвиденного сбоя, незаметно для пользователей создавая резервные копии данных. Все изменения, вносимые в базу данных, регистрируются, поэтому многие операции можно отменять и выполнять повторно.

# 2. Программное проектирование интерактивного новостного сайта

Вначале перед созданием самого интерактивного сайта составим список требований - пунктов, который будет в дальнейшем являться главным ориентиром в процессе написания самой программы.

Этапы создания новостного сайта:

1. Создать базы данных для шапки сайта, постов и хранения подписчиков.

2. С помощью языков программирования PHP, HTML и набора инструментов BOOTSTRAP создать структура сайта, организовать его работоспособность.

Требования к сайту:

1. Сайт должен иметь шапку с разделами новостей по тематике.

2. На главной странице сайта должны публиковаться все посты, хранящиеся в базе данных.

3. На каждой странице раздела должны располагаться посты, удовлетворяющие тематике самого новостного раздела.

4. В отдельной области сайта должен располагаться список всех подписчиков новостного портала и строка для регистрации новых подписчиков по email.

5. Все вносимые пользователем изменения должны синхронизироваться с базой данных и наоборот.

6. Около каждого поста должны располагаться дата и время его создания, а также переход на саму страницу поста или на главную страницу.

# 3. Практическая реализация интерактивного новостного сайта

Для создания интерактивного новостного сайта, постоянно ссылающего на базу данных, будет использоваться фреймворк для создания современных, кросс-браузерных и стандартизованных интерфейсов. Продуманная структура кода HTML, PHP, JavaScript и CSS даст нам возможность создавать множество самых разнообразных элементов интерфейса и сетку сайта.

Основные инструменты Bootstrap:

* Сетки — заранее заданные размеры колонок, которые можно сразу же использовать, например ширина колонки 90px относится к классу .span2;
* Шаблоны — фиксированный или резиновый шаблон документа;
* Типографика — описания шрифтов, определение некоторых классов для шрифтов таких как код, цитаты и т.п.;
* Медиа — представляет возможности управления изображениями и видео;
* Таблицы — средства оформления таблиц, вплоть до добавления функциональности для обеспечения возможности сортировки;
* Формы — классы для оформления не только форм, но и некоторых событий происходящих с ними;
* Навигация — классы оформления для вкладок, страниц, меню и панелей инструментов;
* Алерты — оформление диалоговых окон, подсказок и всплывающих окон.

Для создания базы данных, а также для дальнейшей работы с ней, будет использоваться MySQL — свободная реляционная система управления базами данных и PHPMyAdmin — веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL [2].

## 3.1 Создание базы данных

База данных News – главное хранилище данных работы, в соответствии с планом, должна содержать информацию о постах новостного сайта, его шапке (разделах) и о пользователях (подписчиках). Следовательно, для запланированной правильной работы сайта потребуется 3 таблицы: categories (таблица для хранения разделов сайта), posts (таблица для хранения постов сайта) и subscribers (таблица для хранения email-адресов подписчиков и их индивидуальных ключей). С полями данных таблиц можно подробнее ознакомиться на рисунке 3.1.1.

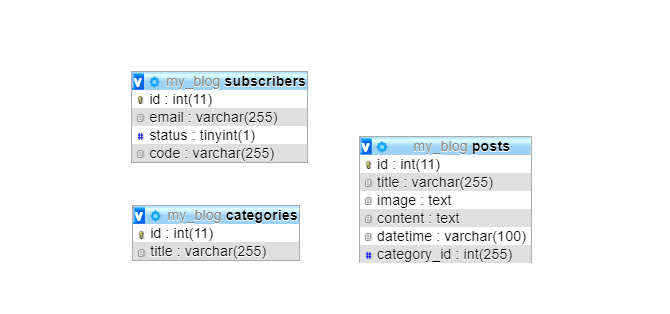


Рисунок 3.1.1 – Структура полей таблиц базы данных

С помощью связи один ко многим свяжем между собой две таблицы постов и категорий. Ведь какой-то одной категории может по тематике подходить несколько написанных на сайте постов.

Следующим действием заполним созданные таблицы данными. Таким образом, в таблице categories имеются такие разделы как: Новости дня, Происшествия, Новости мира, Новости кино, Новости финансов, Новости спорта. В таблице постов создадим несколько записей, некоторые из которых подходят к одной и той же категории по своей тематике. Также в таблицу пользователей добавим несколько пользователей и присвоим им индивидуальные коды [3].

На этом этапе можно сделать вывод, что база данных – каркас сайта − была создана успешно. Поэтому можно переходить уже непосредственно к разработке представления страниц новостного сайта, постоянно взаимодействующих с базой News.

## 3.2 Создание шапки сайта

Для того чтобы создать шапку сайта создадим php – файл header.php, в котором определим сетку сайта, разметку полей, колонок, определим наличие изображения на главной полосе сайта. Реализация данного участка кода представлена в листинге 3.2.1.

Листинг 3.2.1

<div class="navbar navbar-inverse navbar-fixed-top">  
 <div class="container">  
 <div class="navbar-header">  
 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#responsive-menu">  
 <span class="sr-only">Открыть навигацию</span>  
 <span class="icon-bar"></span>  
 <span class="icon-bar"></span>  
 <span class="icon-bar"></span>  
 </button>  
 <a class="navbar-brand" href="/curs">  
 NewsNow!  
 </div>  
 <img src="/curs/public/img/a.png" width="60" height="25" class="d-inline-block align-top" alt=""></a>  
 <div class="collapse navbar-collapse" id="responsive-menu">

Далее заполним новостной сайт разделами, извлечёнными из базы данных с помощью функции get\_categories. Исходный код, удовлетворяющий всем выше сказанным словам, представлен в листинге 3.2.2.

Листинг 3.2.2

function get\_categories(){  
 global $link;  
 $sql = "SELECT \* FROM `categories`";  
 $result = mysqli\_query($link,$sql);  
 $categories=mysqli\_fetch\_all($result,MYSQLI\_ASSOC);  
 return $categories;  
 }

<?php  
$categories = get\_categories();  
?>

<?php foreach($categories as $category): ?>  
<li><a href="/curs/category.php?id=<?=$category["id"]?>"><?=$category

["title"]?></a></li>  
<?php endforeach; ?>  
<?php endif; ?>

## 3.3 Подключение к базе данных

Для того чтобы подключится к базе данных был создан специальный php-файл database.php. В данном файле происходит подключение к базе данных с помощью функции mysqli(), которая принимает 4 аргумента: имя сервера базы данных, имя пользователя, пароль и название базы данных [4]. Также следует установить нужную кодировку во избежание ошибок в воспроизведении текста на страницах новостного сайта. Данное подключение представлено в листинге 3.3.1.

Листинг 3.3.1

<?php  
$link = new mysqli('localhost','dima','dima','news');  
$link->set\_charset("utf8");

## 3.4 Создание главной страницы сайта, вывод постов на главную страницу

Главная страница сайта (файл index.php) будет представлена таким образом: слева будут располагаться новости, хранящиеся в базе данных, причем ниже каждой новости будет располагаться дата и время его опубликования. Также у каждого новостного поста будет специальная кнопка для перехода на саму страницу поста (файл post.php), причем переход можно будет осуществить, не только нажав на неё, но и на изображение или заголовок поста. Для работы с новостными постами из базы данных будет использоваться специальная функция для получения записей get\_posts. Исходный код листинга 3.4.1 удовлетворяет описанному выше.

Листинг 3.4.1

<div class="container">  
 <div class ="row">  
 <div class="col-md-9">  
 <div class="page-header">  
 <h1>Все Записи: </h1>  
 </div>  
 <?php  
 $posts= get\_posts();  
 ?>  
  
 <?php foreach ($posts as $post):?>  
  
  
 <div class="row">  
 <div class="col-md-3">  
 <a href="/curs/post.php?post\_id=<?=$post['id']?>" class="thumbnail">  
 <img src="<?=$post['image']?>" alt="">  
 </a>  
 </div>  
 <div class="col-md-9">  
 <h4><a href="/curs/post.php?post\_id=<?=$post['id']?>"><?=$post['title']?></a></h4>  
 <p>  
 <?=mb\_substr($post['content'],0,300, 'UTF-8').'...'?>  
 </p>  
 <p><a class="btn btn-info btn-sm" href="/curs/post.php?post\_id=<?=$post['id']?>">Читать полностью</a></p>  
 <br/>  
 <ul class="list-inline">  
  
 <li><i class="glyphicon glyphicon-calendar"></i><a><?=$post['datetime']?></a>  
  
 </ul>  
 </div>  
 </div>  
 <hr>  
 <?php endforeach; ?>

Справа же будет располагаться специальная форма для регистрации новых подписчиков новостного сайта. Вся информация будет автоматически поступать в базу данных, причём одного и того же пользователя зарегистрировать не удастся. Для реализации задуманного следует создать отдельный файл sidebar.php. Исходный код файла sidebar.php представлен в листинге 3.4.2

Листинг 3.4.2

<br></br>  
<div class="well">  
 <div class="form-group">  
 <form action="/curs/subscribe.php" method="post">  
 <h4>Подпишись нам сайт</h4>  
 <input type="email" name="email" value="" class="form-control" placeholder="Введите @email">  
 <p></p>  
 <button type="submit" class="btn btn-success">Подписаться</button>  
 </div>  
 <div class="form-group">  
 <?php  
 $emails=get\_subscribers();  
 ?>  
 <h4>Мои подписчики:</h4>  
 <?php foreach($emails as $email): ?>  
 <p><?=$email["email"]?></p>  
 <?php endforeach; ?>  
 </div>  
</div>

Реализация функций get\_posts и get\_subscribers упомянутых в листингах 3.4.1 и 3.4.2 представлена в листинге 3.4.3.

Листинг 3.4.3

function get\_posts(){  
 global $link;  
 $sql = "SELECT \* FROM `posts`";  
 $result = mysqli\_query($link,$sql);  
 $posts=mysqli\_fetch\_all($result,MYSQLI\_ASSOC);  
 return $posts;  
 }

function get\_subscribers(){  
 global $link;  
 $sql = "SELECT email FROM `subscribers`";  
 $result = mysqli\_query($link,$sql);  
 $emails=mysqli\_fetch\_all($result,MYSQLI\_ASSOC);  
 return $emails;  
}

## 3.5 Создание файла subscribe.php для записи в базу данных новых подписчиков и файла post.php для просмотра новостных постов в полном их объёме.

В файле subscribe.php главную роль будет играть функция, напрямую общающаяся с базой данных, которая будет вставлять новые строки в таблицу подписчиков на сервере базы. Реализация данной функции представлена в листинге 3.5.1.

Листинг 3.5.1

<?php  
  
 if (isset($\_POST['email']))  
 {  
 $email =trim($\_POST['email']);  
 $insert\_result=insert\_subscriber($email);  
 $header ='Location: /curs/?subsribe=';  
 $header .= $insert\_result;  
 header ($header);  
 }  
 else {  
 header('Location:/curs');  
 }

function insert\_subscriber($email){  
 global $link;  
 $email = mysqli\_real\_escape\_string($link,$email);  
 $query = "SELECT \* FROM subscribers WHERE email='$email'";  
 $result = mysqli\_query($link,$query);  
  
 if (!mysqli\_num\_rows($result)){  
 $subcriber\_code = generate\_code();  
  
 echo 'Спасибо за регистрацию- это ваш личный пароль';  
 echo $subcriber\_code;  
 $insert\_query = "INSERT INTO subscribers (email,code) VALUES('$email','$subcriber\_code')";  
 $result =mysqli\_query($link,$insert\_query);  
 if ($result)  
 {return 'CREATED';}  
 else  
 {return 'FAIL';}  
 }  
 else {return 'EXISTS';}

В файле же post.php будет располагаться страница с картинкой новости, с полным текстом новостного поста, а также временем и датой поста вместе с переходом на главную страницу сайта. Это можно увидеть в листинге 3.5.2.

Листинг 3.5.2

<div class="container">  
 <div class ="row">  
 <div class="col-md-9">  
 <div class="page-header">  
 <h1><?=$post['title']?></h1>  
 </div>  
 <ul class="list-inline">  
 <li><i class="glyphicon glyphicon-list"></i> <a href="/curs">NewsNow!</a> | </li>  
 <li><i class="glyphicon glyphicon-calendar"></i> <a><?=$post['datetime']?></a>  
  
 </ul>  
 <div class="post-content">  
 <img src="<?=$post['image']?>"align="left" style="padding:0 10px 10px 0">  
 <?=$post['content']?>  
  
 </div>  
 </div>  
 <div class="col-md-3">  
 sidebar  
 </div>

## 3.6 Создание фильтрации постов по разделам новостного сайта

Данная операция фильтрации новостей будет производиться в файле category.php. Основой будет являться связь один ко многим между таблицами category и posts в базе данных News. В таблице posts содержится поле category\_id, по которому и можно будет понять, к какому разделу сайта будет относиться данная новость. Создание category.php и функция get\_post\_by\_id для извлечения новостей по их post\_id представлено в листинге 3.6.1.

Листинг 3.6.1

function get\_post\_by\_id($post\_id){  
 global $link;  
 $sql = "SELECT \* FROM `posts` WHERE `id` = ".$post\_id;  
 $result = mysqli\_query($link,$sql);  
 $post=mysqli\_fetch\_assoc($result);  
 return $post;  
 }

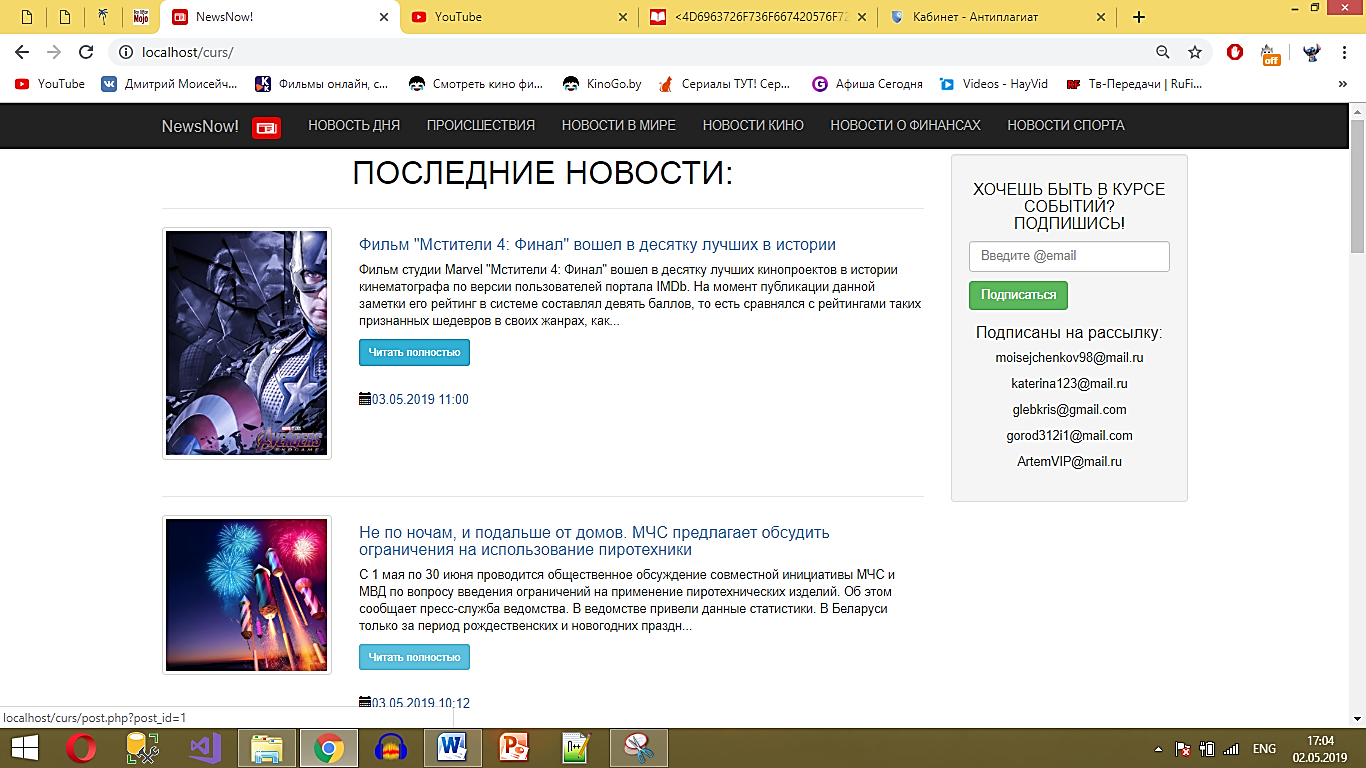
<?php  
$category\_id = $\_GET['id'];  
$posts= get\_posts\_by\_category($category\_id);  
$category\_title=get\_category\_title($category\_id);  
?>  
<?php foreach ($posts as $post):?>  
  
 <div class="row">  
 <div class="col-md-3">  
 <a href="/curs/post.php?post\_id=<?=$post['id']?>" class="thumbnail">  
 <img src="<?=$post['image']?>" alt="">  
 </a>  
 </div>  
 <div class="col-md-9">  
 <h4><a href="/curs/post.php?post\_id=<?=$post['id']?>"><?=$post['title']?></a></h4>  
 <p>  
 <?=mb\_substr($post['content'],0,150, 'UTF-8').'...'?>  
 </p>  
 <p><a class="btn btn-info btn-sm" href="/curs/post.php?post\_id=<?=$post['id']?>">Читать полностью</a></p>  
 <br/>  
 <ul class="list-inline">  
 <li><i class="glyphicon glyphicon-list"></i> <a href="/curs">NewsNow!</a> | </li>  
 <li><i class="glyphicon glyphicon-calendar"></i> <?=$post['datetime']?>  
  
 </ul>  
 </div>  
 </div>  
 <hr>  
<?php endforeach; ?>

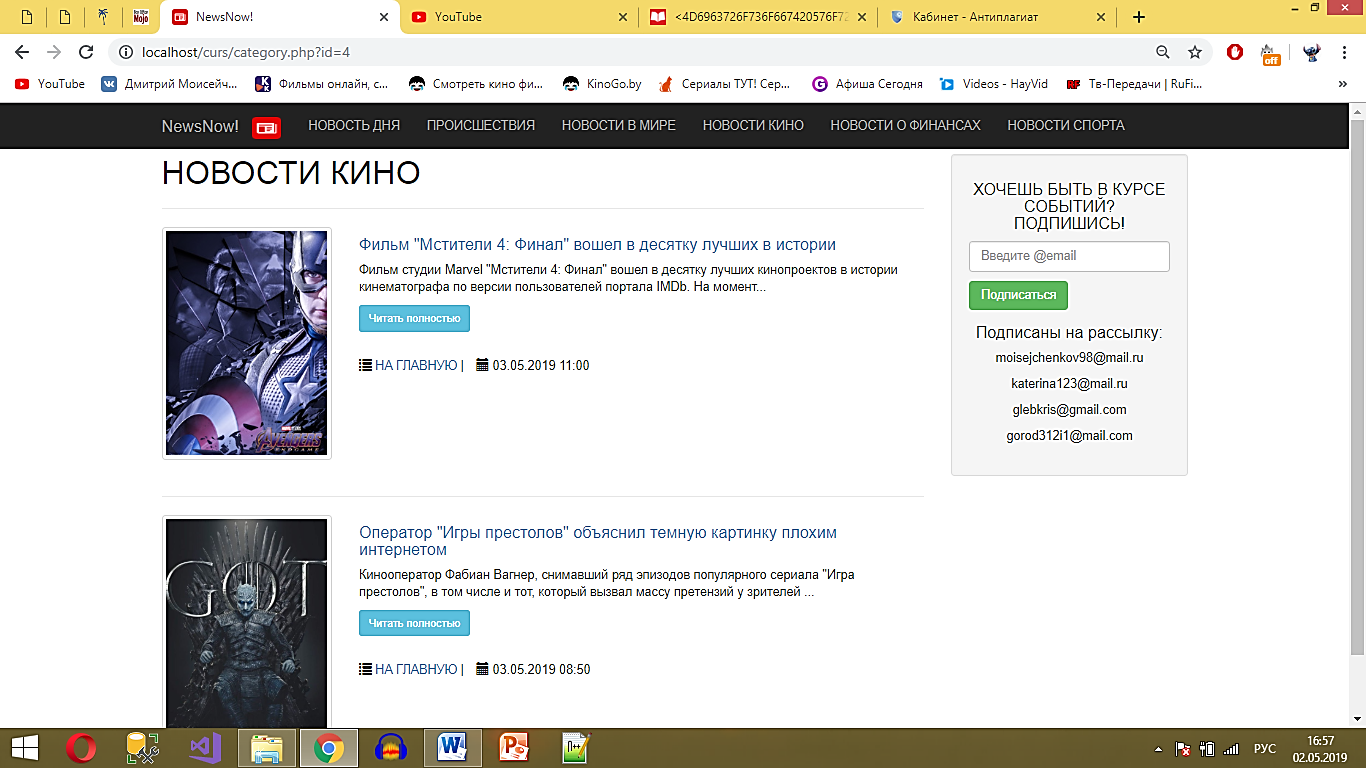
После того, как будет создан последний файл, описанный выше, можно сделать вывод, что все запланированные условия в пункте 2 были с успехом выполнены. Поэтому можно переходить к следующему пункту курсовой работы, а именно, к тестированию интерактивного новостного сайта уже в браузере.

# 4. Тестирование интерактивного новостного сайта

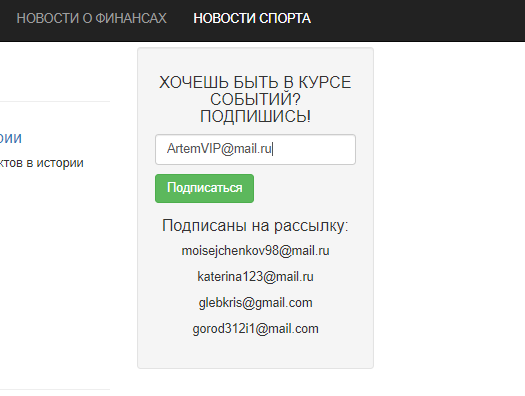
После успешной реализации всех запланированных этапов и после того, как все условия курсовой работы были соблюдены. Можно сделать вывод, что цель, поставленная в начале курсовой работы, была достигнута. Однако теперь следует протестировать работоспособность новостного сайта.

Результаты тестирования представлены на рисунках 4.1- 4.4

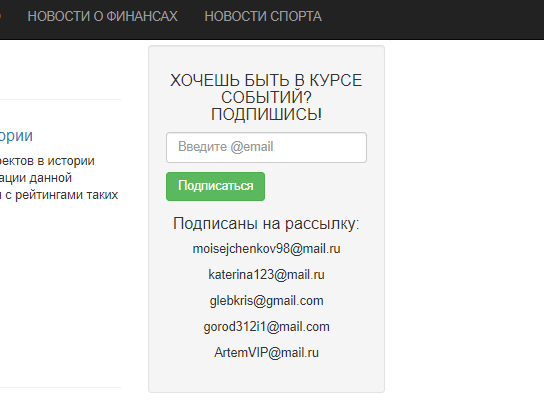


**Рисунок 4.1 – Главная страница новостного сайта**

**Рисунок 4.2 – Раздел Новости Кино**



**Рисунок 4.3 – Регистрация подписчика сайта**



**Рисунок 4.4 –** **Подписчик добавлен**

## 4.1 Результаты тестирования

Как видно из тестов программы, новостной сайт работает без ошибок. При переходе на другие страницы сайта ошибок обнаружено также не было. Запись о новом пользователе (подписчике) была добавлена в базу данных без каких-либо затруднений. Все новости фильтруются по своей тематике и каждому разделу соответсвует определённый набор новостных постов, удовлетворяющих данной тематике. Интерфейс сайта прост и удобен. Все главные элементы сайта выполняют свои функции правильно. Следовательно, можно сделать вывод, что интерактивный новостной сайт был создан успешно. Значит цель, которая была поставлена в начале курсовой работы, была достигнута.

# Заключение

В результате выполнения курсовой работы были получены навыки работы и поиска учебной литературы, улучшены навыки изучения и анализа различных источников, были подробно описаны все этапы работы и использовании представленных алгоритмов.

Интерактивный новостной сайт − главная цель работы − является удобным и простым в понимании сайтом. Сам новостной портал работает правильно, без ненужных прерываний и задержек при переходе на страницы.

Кроме того, были углублены и закреплены практические навыки работы с языками программирования PHP,NTML и с [реляционной системой управления базами данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94) MySQL. При тестировании и отладке программы каких-либо сбоев в работе новостного сайта не произошло.

Следовательно, можно сделать вывод, что поставленная нами цель в начале курсовой работы была достигнута.

# Список использованных источников

1. MySQL − Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/MySQL> − Дата доступа: 10.12.2018

2. PhpMyAdmin − Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/PhpMyAdmin − Дата доступа: 10.12.2018

3. Дронов, В.  PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов/ В.Дронов, −М.: BHV 2016.− 688 с.

4. Янк, К. PHP и MySQL. От новичка к профессионалу/ К.Янк. −М.: Эксмо, 2017.− 384 с.

# ПРИЛОЖЕНИЕ А

**Содержание электронного носителя**

На электронном носителе расположены следующие директории и файлы:

− Исходный код программы

* Файл «Курсовая работа «Создание интерактивного новостного сайта Моисейченков Д.Ю.» с документацией к курсовой работе

− Презентация «Создание интерактивного новостного сайта Моисейченков Д.Ю»