Публикация на тему

PHP базовый

компьютерного проектирования

Анотация

-

-

Автор

Михалькевич Александр Викторович

Публикация

Наименование PHP базовый

Автор А.В.Михалькевич

Специальность компьютерного проектирования,

Анотация

-

Anotation in English

-

Ключевые слова php

Количество символов 8781

Содержание

Введение
1 Инструментарий
2 Разработка макета
3 Разработка приложения с помощью простого шаблона проектирования
4 Подключение к базе данных MySQL
5 Регистрация пользователей
6 Авторизация пользователей
7 Кабинет пользователя и выход пользователя
Заключение
Список использованных источников
Приложения

Введение

Request

1 Инструментарий

Для Windows - наилучшим решением является

OpenServer, т.к. уже содержит несколько версий PHP, базы данных и СУБД для баз данных, на выбор, а также два сервера Apache и Nginx

Git - система контроля версий

github.com - удаленный git-репозиторий

среда разработки - на выбор, это может быть PHPStorm, Netbeanse, Aton или Notepad++.

Вопросы: Серверные языки программирования Язык программирования PHP, особенности и определения Сервера для PHP Суперглобальные переменные PHP

2 Разработка макета

Макет главной страницы - это файл index.html, содержащий подключения стилей и js-скриптов.

Первоначальная структура макета:

media/css/style.css

/js

index.html

Данная структура моожет быть расширена использованием css-фрэймворка Bootstrap

media/css/style.css

/js

/bootstrap

index.html

Готовая структура bootstrap-шаблона доступна по ссылке

https://github.com/mikhalkevich/bootstrap4

Вопросы: Использование простого шаблона проектирования

3 Разработка приложения с помощью простого шаблона проектирования

1. Шаблон главной страницы, файл index.html переименовываем в index.php

2. В папку с проектом добавляем папку templates, содержащую файлы top.php, bottom.php

Структура проекта:

/media/css

/templates/top.php

/bottom.php

index.php

3. Содержимое файла index.php

```
require_once('templates/top.php');
//меняющаяся часть шаблона
require_once('templates/bottom.php')
```

Файлы top.php и bottom.php содержат неизменную часть шаблона. Изначально, это в основном html-код.

Вопросы: Использование функции require() для шаблонизации проекта. Создание переменных для конфигурации проекта Response, вывод ответа Request, запросы пользователя Сохранение данных на стороне пользователя, Cookie Сохранение данных на стороне сервера, Session Типы данных PHP Перевод строки в массив, использование функции explode() Перевод массива в строку, использование функции implode() Отличие одинарных и двойных ковычек при работе со строками Сохранение файлов на сервере, использование функции move_uploaded_file() Формат даты, использование функции date() Преимущества фрэймворков перед системами управления контентом (CMS) Объектно-ориентрованное программирование в PHP. Создание классов и объектов. Методы и свойства класса 4 основополагающих принципа ООП: наследование, инкапсуляция, полиморфизм и абстракция

4 Подключение к базе данных MySQL

В поекте создадим папку /config, с пока еще единственным файлом config.php. Листинг файла:

```
error_reporting(E_ALL & ~E_NOTICE);

$dblocation = "localhost";

$dbname = "wwwwwwww";

$dbuser = "root";

$dbpasswd = "";

$link = mysqli_connect($dblocation, $dbuser, $dbpasswd, $dbname);

if (!$link){
    exit('Error connect database');
}
```

Подключение конфигурационного файла осуществляется в базовом шаблоне, в файле templates/top.php:

```
require_once('config/config.php')
```

Теперь, в любом исполняемом файле возможны запросы в базу данных через переменную подключения $link.

```
    if(isset($_GET['url'])){
        $url = $_GET['url'];
```

```
    }else{
        $url = 'index';
    }
    $query = "SELECT * FROM main_texts WHERE url = '$url'";

    $tab = mysqli_query($link, $query);

    if(!$tab){
        exit($query);
    }

    $arr = mysqli_fetch_array($tab);
```

Вопросы: Установка MySQL Система управления базами данных PHPMyAdmin Функциональное подключение к базе данных Выполнение sql-запросов с помощью mysqli_ -подобных функций Вставка данных, использование оператора INSERT Вывод данных, использование оператора SELECT Удаление данных, использование оператора DELETE Обновление данных, использование оператора UPDATE

5 Регистрация пользователей

Для реализации регистрации пользователей нам потребуется таблица users: id(AI), name(string), email(string), password(string), status(string), lastvisit(datetime), datereg(date)

PHP-файл, принимающий данные регистрационной формы со следующими post-данными: name, email, pass

```
require_once 'templates/top.php';

if (!empty($_POST)){

    /**
     * Проверяем пароли
     */
    $error = [];

    if ($_POST['pass'] != $_POST['pass-repeat']){
        $error[] = 'Не совпадают пароли';
    }

    /**
     * Проверяем есть ли такой пользователь в БД
     */
    $query = "SELECT * FROM users WHERE email = '".$_POST['email']."'";

    $usr = mysqli_query($link, $query);

    /**
     * Если пользователя нет, то отключаемся. Иначе, добавляем ошибку
     */
    if(!$usr)
```

```
    {
        exit($query);
    }

    $check = @mysqli_fetch_array($usr);

    #print_r($check);

    if($check['id'])
    {
        $error[] = "Пользователь с таким именем уже существует";
    }


    if (empty($error)){

        $query = "INSERT INTO users VALUES (NULL,
                                            '{$_POST['name']}',
                                            '{$_POST['email']}',
                                            '{$_POST['pass']}',
                                            'default',
                                            NOW(),
                                            NOW()
                                            )";

        $result = mysqli_query($link, $query);

        if (!$result){
            exit('Ошибка');
        }else{
            echo 'Пользователь добавлен';
        }

    }else{

        foreach ($error as $item) {
            echo $item . '
';
        }

    }

}
```

Вопросы: Регистрация и авторизация пользователя Создание формы и принятие данных элеменов форм, использование типа запроса POST

6 Авторизация пользователей

Авторизация пользователей осуществляется в отдельном файле login.php, который содержит

авторизационную форму с полями email и pass, а также код, обрабатывающий форму:

```
require_once 'templates/top.php';
if (!empty($_POST)){

    $query = "SELECT * FROM users WHERE email='{$_POST['email']}' AND
password='{$_POST['pass']}' AND status='default' LIMIT 1";

    $result = mysqli_query($link, $query);

    if (!$result){
        echo 'Ошибка';
    }

    $user = mysqli_fetch_array($result);

    if ($user['id']){

        $_SESSION['user_id'] = $user['id'];
        $_SESSION['user_name'] = $user['name'];

 // перенаправление

    }else{
        echo 'Не удалось войти';
    }
}
```

Использование сессионной переменной, файл top.php:

```
session_start();

require_once('config/config.php');
$profile_user = [];
$query = "SELECT * FROM main_texts";
$tab = mysqli_query($link, $query);
if (!$tab) {
    exit('error query');
}
if ($_SESSION['user_id']) {
    $query = "SELECT * FROM profiles WHERE user_id = " .
$_SESSION['user_id'];
    $usr = mysqli_query($link, $query);
    if (!$usr) {
        exit($query);
    }
    $profile_user = @mysqli_fetch_array($usr);
}
```

7 Кабинет пользователя и выход пользователя

После авторизации для пользователя создается сессионная переменная

```
$_SESSION['user_id']
```

Тогда проверка авторизации пользователя выглядит следующим образом:

```
if($_SESSION['user_id']){
// пользователь авторизован
}else{
// пользователь не автоирзован
}
```

Сохранение данных в таблицу articles с привязкой к текущему пользователю:

```
    $query = "INSERT INTO `articles` (`id`, `title`, `slug`, `body`, `lang`, `user_id`, `putdate`) VALUES (Null, '{$_POST['title']}', '{$_POST['slug']}', '{$_POST['body']}', 'ru', '{$_SESSION['user_id']}', NOW())";

    $result  = mysqli_query($link, $query);
```

Вывод записей текущего пользователя по $_SESSION['user_id']

```
    $query = "SELECT * FROM articles WHERE user_id = '".$_SESSION['user_id']."'";

    $usr = mysqli_query($link, $query);

    if(!$usr)
    {
        exit($query);
    }

    $check = @mysqli_fetch_array($usr);

    #print_r($check);
```

Выход пользователя реализован в файле logout.php

```
session_start();

session_destroy();

header('location:index.php');
```

Вопросы: Вывод статей из базы данных, использование типа запроса GET

Заключение

Список использованных источников

1. [url] **Документация PHP** http://php.net

Приложения