

практика, саме при розумному, раціональному та ефективному поєднанні новітніх інформаційно-комунікаційних технологій, цифрових освітніх ресурсів з різноманітними навчальними методиками сприяє особистістній орієнтації педагогічного процесу, підвищенню пізнавальної активності учнів, покращує ефективність управління навчальною діяльністю учнів, легкому засвоєнню учнями навчальних програм. Учні активно включаються в навчальну діяльність, зростає їх інтерес до вивчення автосправи, вони самостійно готують презентації, доповіді з певних тем і виступають з ними, тобто приймають активну участь у вивченні нового матеріалу.

ЛІТЕРАТУРА:

1. <http://mykhailo-heletiy.blogspot.com/> М. Гелетій

Кривий Ю.

Науковий керівник – Луцик І.Б.

РОЗРОБКА ІНФОРМАЦІЙНОГО ON-LINE РЕСУРСУ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ CODEIGNITER

Постановка проблеми. Одним із основних засобів комунікацій та спілкування між людьми як у бізнесі, так і в побуті поступово стає Інтернет. Тому на сьогоднішній день мати свій власний сайт вже не є забаганкою, а вимогою успішного бізнесу.

Для того, щоб привернути увагу відвідувача а також для зручності наповнення сайту, необхідно використовувати найновіші і популярні технології. Тому актуальним є вивчення сучасних способів проектування сайтів із застосуванням наборів різноманітних бібліотек із великим, добре розширюваним функціоналом інструментів для швидкого виконання рутинних завдань.

Метою статті є представлення способу розробки on-line ресурсів за допомогою фреймворку CodeIgniter мови програмування PHP, який забезпечує можливість створення динамічних сторінок.

Основна частина. Технології роботи в Інтернеті на даний час дуже швидко розвиваються, що є одним з найбільш значних досягнень XX - XXI століття. В основі всіх цих технологій лежить мова HTML (HyperText Markup Language - мова гіпертекстової розмітки), яка пройшла кілька етапів розвитку, що завершувалися появою нових версій [1, с.11].

За допомогою HTML можна створювати веб-сторінки зі статичним вмістом. Однак при перегляді різних сайтів ми можемо помітити, що їх вміст подібними сторінками не обмежується. Сучасні сайти наповнені сценаріями, додатками. Тому для створення інформаційного онлайн-ресурсу, доцільним є використання мови PHP, яка, зокрема, забезпечує можливість створення динамічних сторінок.

Слід зазначити, що для створення повноцінного, масштабного і надійного проекту необхідне використання фреймворків. Фреймворк – це набір різноманітних бібліотек (інструментів) для швидкої розробки повсякденних (рутинних) завдань. Головна мета фреймворку, надати програмісту зручне середовище для проекту з великим і добре розширюваним функціоналом [3].

Одним із найпопулярніших фреймворків мови PHP є CodeIgniter, в якому міститься набір бібліотек, завдяки яким сайт буде написаний швидше і працюватиме краще.

Встановлення CodeIgniter відбувається наступним чином:

Завантажуємо архів із офіційного сайту <http://www.codeigniter.com> (рис. 1).



Рис. 1. Офіційний сайт програмного каркасу CodeIgniter

2. Розпаковуємо архів у корінь папки на локальному сервері.
3. При переході в браузері за адресою нашого сайту має відобразитися наступна інформація (рис. 2):



Рис. 2. Вікно привітання CodeIgniter

Після встановлення даного фреймворку, у каталозі «WWW» тепер з'явилось багато різних каталогів. Для подальшої роботи найбільш використовують папку «application», у якій будуть розміщені всі необхідні файли.

Для створення веб-ресурсу цілком достатньо працювати з папками «Controllers» (контроллери), «Libraries» (бібліотеки), «Models» (моделі), «Views» (види). Крім того, для внесення і зміни налаштувань, слід використовувати файли із папки «Config».

Основою програми є контролери, у яких визначається як необхідно реагувати на той або інший запит. Усі дії і функції, які виконуватимуться на сайті, будуть написані у файлах контроллерів, а вже з них викликатимуться бібліотеки і моделі, та відбуватиметься направлення на види.

Моделі – це класи, які створюються для роботи з інформацією, яка отримується з бази даних. Для інформаційного онлайн-ресурсу необхідний клас моделі з функціями для вставки, оновлення та видалення різних матеріалів. Всі ці дії можна буде виконувати завдяки моделям.

Бібліотеки використовуються для більш складних задач. Їх можна застосовувати для навігації по матеріалах, відображення видів, адміністраторської панелі. CodeIgniter має великий вибір вбудованих бібліотек, але також є можливість створювати свої власні.

Види – це проста веб-сторінка або її фрагмент, наприклад шапка, підвал, бічна панель або т.п. Види ніколи не викликаються безпосередньо: вони повинні бути завантажені з контроллерів, які керують всім, що відбувається. Саме в контроллерах має бути визначено, яке відображення завантажувати.

За допомогою фреймворка CodeIgniter можна дуже легко виводити інформацію на сайт із бази даних. Використаємо для цього приклад готельного комплексу. Для того, щоб виводити інформацію, створюємо у базі даних таблицю «Pages», з якої і будемо отримувати дані. Перед

тим як створювати файли у фреймворку, необхідно додати базу даних у файлі «database.php», який знаходиться у папці «Config».

Починаємо із створення файлів PHP. При кожному створенні файлу першою має бути написана наступна стрічка:

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
```

Перш ніж створити контроллер, в якому будуть написані усі функції, ми повинні створити моделі та бібліотеку, за допомогою яких буде налаштовано зв'язок із базою даних і наш контроллер зможе працювати без помилок.

Створюємо основну модель – «crud.php» і поміщаємо цей файл у папку «Models». CRUD – (англ. create read update delete) чотири базові функції управління даними «створення, читування, зміна і видалення». Файл повинен містити наступний код:

```
class Crud extends CI_Model
{
    public $table = '';
    public $idkey = '';
    public function __construct()
    public function get($obj_id)
    {
        $this->db->where($this->idkey,$obj_id);
        $query = $this->db->get($this->table);
        return $query->row_array();
    }
}
```

У цій моделі, значення таблиць пусті, тому що вони будуть підбиратись із інших моделей, а функція «get» перетворює таблицю у масив, з яким будуть працювати контроллери.

Наступна модель – «pages_model.php», яка написана для роботи з контроллером і таблицею «Pages». Дана модель розміщена в папці моделей, розширяє модель crud, та містить наступний код:

```
class Pages_model extends Crud
{
    public $table = 'pages';
    public $idkey = 'page_id';
}
```

Дані переходять у модель «crud», після чого контроллер може використовувати таблицю «Pages», яка тепер є масивом.

Після того, як моделі створено, необхідно створити бібліотеку – «display_lib.php» і розмістити у папці «Libraries». У дану бібліотеку записуємо наступну інформацію:

```
class Display_lib
{
    public function user_page($data,$name)
    {
        $CI = & get_instance();
        $CI->load->view('page_view',$data);
        $CI->load->view($name.'_view',$data);
    }
}
```

Завдяки цьому коду кожна функція у контроллері зможе завантажувати файл із видами, в якому буде внесена потрібна інформація.

Наступним нашим кроком є створення основного контроллера – «pages.php», який розміщуємо у папку «Controllers». Це є основний файл нашої програми. Саме завдяки контроллерам буде здійснюватися уся робота сайту.

Для зручності подальшого корегування, щоб бачити, що виконують функції на кожному етапі, доцільним є додавання коментарів. PHP надає декілька методів для вставки коментарів. Додовнивши код коментарями, заносимо його у файл «pages.php»:

```
class Pages extends CI_Controller
```

```

{
    public function __construct()
    {
        parent::__construct();
        $this->load->model('pages_model'); // Завантажує модель
    }
    public function index()
    {
        redirect(base_url()); // перенаправляє на головну сторінку
    }
    public function show($page_id)
    {
        $data = array(); // формуємо масив з даними таблиці для передачі
        $data['main_info'] = $this->pages_model->get($page_id);
        switch($page_id)
        {
            case 'index': // обираємо дані з таблиці
                $name = 'pages/mainpage';
                $this->display_lib->user_page($data,$name); // викликаємо вид
        }
        break;
    }
}
}
}

```

У даному контролері використовується оператор вибору Switch-case. Він забезпечує багатонаправлене (множинне) розгалуження у програмі. [2, с.265] Використовуючи цей оператор, ми можемо додати інші сторінки, дописавши перемикач *case*, як у прикладі з головною сторінкою. У попередньому коді використані також функції: *redirect* – перенаправляє на іншу сторінку, та *break* – перериває виконання поточної структури. Завдяки коду даного контролера, буде здійснюватись вивід головної сторінки із таблиці «Pages».

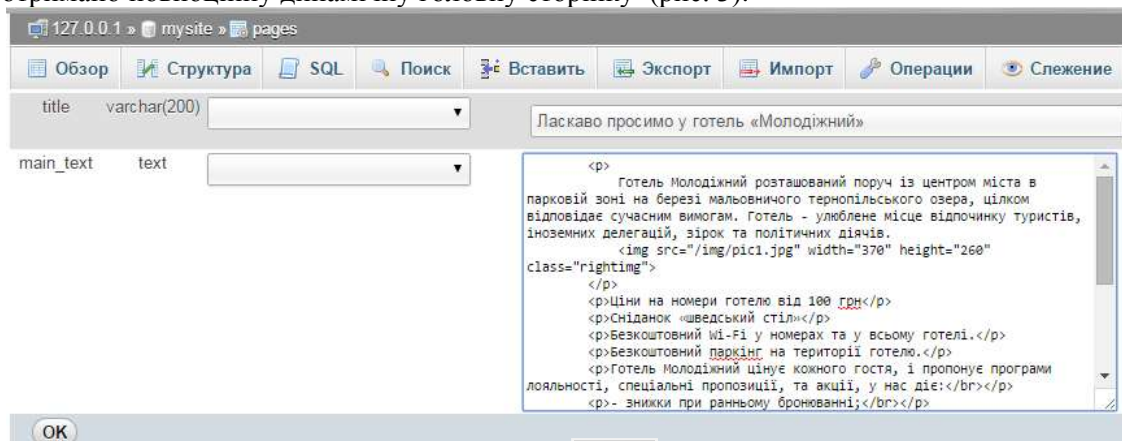
Крім того, необхідно додати також вид «mainpage_view.php», який буде розміщено за адресою: «\www\application\views\pages\». Даний вигляд буде виводити дані на головну сторінку із таблиці:

```

<div id="cont">
    <div id="blokgol">
        <p class="zagolovok"><?=$main_info['title'];?></p>
        <?=$main_info['main_text'];?>
    </div>
</div>

```

Прописавши моделі, та створивши контролери за допомогою фреймворка CodeIgniter нами отримано повноцінну динамічну головну сторінку (рис. 3).



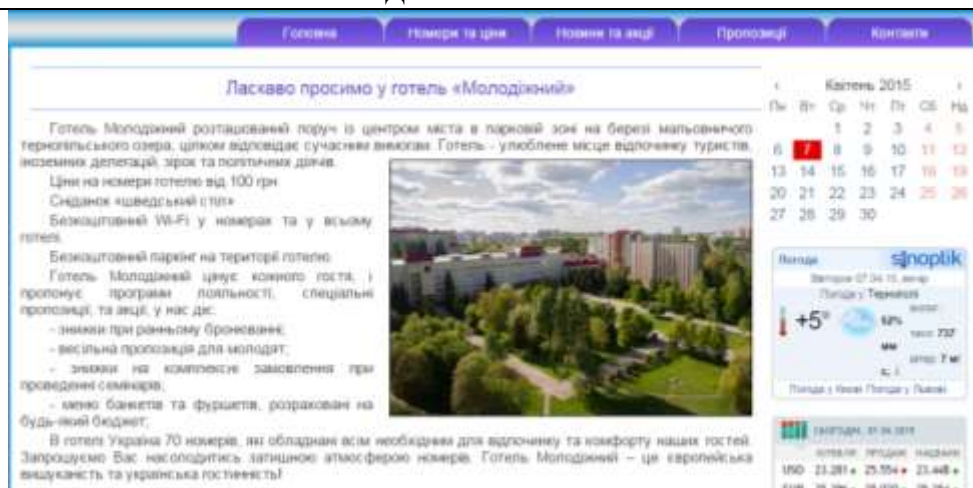


Рис. 3 Вивід головної сторінки із БД

Таким чином, за допомогою фреймворка CodeIgniter, ми отримали веб-сторінку, наповнення інформацією якої здійснюється з попередньо створеної таблиці із бази даних. Таблиця з даними перетворюється в масив за допомогою моделей, які ми створювали. Також в контролері ми маємо можливість виконувати різні операції з даними таблиці, які передаються у вид, за допомогою якого вони відображаються на нашій веб-сторінці.

Висновки.

Для розробки on-line ресурсів, які містять динамічні сторінки доцільним є використання фреймворку CodeIgniter мови програмування PHP. Даний фреймворк містить багато бібліотек, які набагато спрощують роботу веб-дизайнерів. За допомогою елементів цієї програмної платформи, а саме – контролерів, моделей, бібліотек та видів, здійснюється вивід інформації із бази даних.

Отже, знання і вміння працювати із фреймворками мови PHP, є необхідними для повноцінного створення інформаційного онлайн ресурсу.

ЛІТЕРАТУРА:

1. Котеров Д.В. PHP 5 в подленнике/ Котеров Д., Костарев А. — СПб.: БХВ-Петербург, 2015. — 1120 с.
2. Ташков П. А. Веб-мастеринг на 100 %: HTML, CSS, JavaScript, PHP, CMS, AJAX/ Ташков П. А. — СПб.: Питер, 2010. — 512 с.
3. Матеріали офіційного сайту фреймворку CodeIgniter : [Електронний ресурс]. Режим доступу: <http://www.codeigniter.com/>

Мостовий В.

Науковий керівник - доц. Павх І.І.

ПОРІВНЯЛЬНИЙ АНАЛІЗ ЕЛЕКТРИЧНИХ І ГІДРАВЛІЧНИХ ПІДСИЛЮВАЧІВ РУЛЬОВОГО УПРАВЛІННЯ

Електрика чи гідравліка підсилювача керма - вибираємо кращий варіант Вперше в автомобілях підсилювач рульового управління з'явився в 30-тих роках минулого століття. Спочатку це були прості пневматичні підсилювачі, що допомагали водієві впоратися з керуванням важким кар'єрним самоскидом. Поступово пневматику змінила гідравліка, і в 50-их роках в Європі та Америці стали випускатися перші легкові автомобілі, оснащені гідропідсилювачем керма. І вже зовсім недавно в побуті з'явився електропідсилювач рульового механізму, який поступово став витісняти гідравліку.

Відношення автомобілістів до цих систем неоднозначне. Одним подобається сучасний електропідсилювач (ЕПК), для інших переважніше класичний гідропідсилювача (ГПК). Кожна система має свої переваги і недоліки.

Що ж вибрати?

Виходячи з чисто логічного підходу, можна подумати, що нове інженерне рішення, в даному випадку ЕПК, має бути краще колишнього. Воно повинно враховувати недоліки