

УДК 004.738.5:001.89

Технологія розробки персонального сайту сучасного науковця

Вячеслав Осадчий¹, Ірина Сердюк², Дмитро Ліхачов³, Владислав Пятниченко⁴

*Мелітопольський державний педагогічний університет імені Богдана Хмельницького,
м. Мелітополь, Україна*

*¹доктор педагогічних наук, професор,
професор кафедри інформатики і кібернетики
<http://orcid.org/0000-0001-5659-4774>, osadchyi@mdpu.org.ua*

*²асистент кафедри інформатики і кібернетики
<https://orcid.org/0000-0001-6808-0586>, bastinda72@gmail.com*

*³студент IV курсу
<https://orcid.org/0000-0003-0759-8650>, likhachov1995@gmail.com*

*⁴студент магістратури
vladyslav.piatnychenko@gmail.com*

Анотація. Статтю присвячено проблемі підвищення статусу науковця, довіри до результатів його досліджень та потребою у поширенні інформації у світовому науковому співтоваристві. Проаналізовано сучасні тенденції у веб-програмуванні, характеристики процесу розробки веб-ресурсів. Розглянуто технологію створення сайтів, основні можливості Yii фреймворку. Здійснено порівняльний аналіз найбільш популярних фреймворків. Подано опис інтерфейсу та етапів розробки персонального сайту сучасного науковця. Розроблений сайт може бути використано сучасними науковцями для вирішення наукових і практичних завдань по представленню більш детальної персональної інформації та результатів власних досліджень усій науковій спільноті в мережі Інтернет.

Ключові слова: веб-ресурс; наукова комунікація; наукове співтовариство; персональний сайт; програмний засіб; фреймворк.

Development of the Personal Site of a Modern Scientist

Viacheslav Osadchy¹, Iryna Serdiuk², Dmytro Likhachov³, Vladyslav Piatnychenko⁴

*Bogdan Khmelnytsky Melitopol state pedagogical university,
Melitopol, Ukraine*

*¹Doctor of Pedagogic Sciences, Professor,
Professor of the Department of Informatics and Cybernetics
<https://orcid.org/0000-0001-5659-4774>, osadchyi@mdpu.org.ua*

*²assistant of the Department of Informatics and Cybernetics
<https://orcid.org/0000-0001-6808-0586>, bastinda72@gmail.com*

*³bachelor student
<https://orcid.org/0000-0003-0759-8650>, likhachov1995@gmail.com*

*⁴master student
vladyslav.piatnychenko@gmail.com*

Abstract. The article is devoted to the actual and practically important problem for modern science - to provide free access to the scientific creative heritage of the scientist. After all, the weight of the scientist in the professional community, his influence on the events occurring in his chosen field of science, today largely depends on how fully, constructively and organically presented the results of his research on the Internet. The central element of a visiting card of a modern scientist is a link to a personal website. Keeping your own site allows you to create an electronic library of scientific and methodological developments, share your thoughts, finds. In addition, the creation and maintenance of their site contributes to replenishing the portfolio of achievements of the scientist. The authors proceed from the basic provision that the personal page as a form of information provision in the scientific community gives scientists and specialists new opportunities for solving scientific and practical problems. The materials of the scientist, presented on the site, allow to show the level of erudition in the branches of science and culture, help to achieve public recognition.

The article analyzes the scientific and special literature on web programming, examines the requirements for the creation of sites, developed a personal site of a scientist, written recommendations on its use.

The developed personal site of a modern scientist is a significant source of biographical and bibliographic information, gives an opportunity to present to the international scientific community information about their activities and ideas in the form of published and unpublished scientific works, which enables the scientist to solve the issues of creating their own image in the scientific space.

Keywords: web-resource; scientific communication; scientific community; personal site; software tool; frameworks.

ВСТУП

Теперішнє тисячоліття характеризується бурхливим розвитком інформаційних технологій глобального характеру. Удосконалення комп'ютерної техніки, програмного забезпечення, створення автоматизованих систем обробки інформації, електронних баз і банків даних, найскладніших аналітичних та експертних систем, а також бурхливий розвиток різноманітних телекомунікаційних мереж, включаючи глобальну мережу Інтернет, істотно змінюють сучасні суспільні відносини. При переході до нового, інформаційного типу суспільства безліч сфер життєдіяльності людини – економіка, політика, наука, освіта, культура – стають неможливими без використання інформаційних технологій.

З появою мережі Інтернет, що забезпечує повсюдний доступ до інформаційних ресурсів у реальному режимі часу, революційні зміни відбулися і в системі наукових комунікацій. Мережа Інтернет створила умови для доступу вчених до наукових видань та різноманітних баз даних, значно розширила можливості їхнього спілкування.

Вага вченого в професійному співтоваристві, його вплив на події, що відбуваються в обраній ним галузі науки, сьогодні багато в чому визначаються тим, наскільки повно, конструктивно і органічно представлені результати його досліджень у мережі Інтернет. Центральний елемент візитної картки сучасного науковця – посилання на персональний сайт. Це посилання зараз стає обов'язковим атрибутом будь-якої наукової роботи, яка публікується. Сайт дозволяє представити більш детальну інформацію про науковця, його інтереси, погляди всій науковій спільноті. Матеріали науковця, викладені на сайті, дозволяють показати рівень ерудиції в галузях науки і культури, допомагають досягти суспільного визнання. Ведення власного сайту

дозволяє створити електронну бібліотеку науково-методичних розробок, ділитися своїми думками, знахідками. Крім того, створення і ведення свого сайту сприяє поповненню портфолію досягнень науковця.

Розглядаючи це питання доцільно звернутися до досліджень Т. А. Полілової ([Полилова, 2011](#)) та Л. В. Савенкової ([Савенкова, 2014](#)), які доводять, що персональні сторінки вчених стали невід'ємним компонентом наукового інтернет-середовища. В. Здановська ([Здановська, 2010](#)) розглядала питання бібліометричного аналізу та класифікації персональних сторінок вчених у мережі Інтернет. Значення веб-сторінки вчених як складової міжнародних систем наукової інформації розкрито у працях Пастушенко О. ([Пастушенко, 2015](#)). Данилюк С. акцентував увагу на тому, що найповніше особистість може бути реалізована в мережі за допомогою персональних сайтів ([Данилюк, 2010](#)).

Забезпечення вільного доступу до наукової творчої спадщини вченого та його школи формує високий ступінь довіри до авторитету вченого та підтримує його і його учнів.

Мета статті – опис розробки та застосування персонального сайту сучасного науковця. Для цього були поставлені такі завдання як аналіз спеціальної літератури та мережних джерел з метою вивчення досвіду розробки сайтів; аналіз існуючих каркасних додатків та визначення інструментарію для розробки програмного засобу; опис алгоритму застосування розробленого персонального сайту сучасного науковця.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

Сайт є набором веб-сторінок, об'єднаних загальною тематикою і зв'язаних між собою гіперпосиланнями, єдиною системою навігації. Як правило, має кінцеву кількість сторінок, головною задачею яких – за допомогою візуального оформлення максимально швидко та ефективно донести до користувача суть представленої інформації, ступінь її корисності й грандіозності ([Веб-сайт, 2011](#)).

Процес розробки онлайн-ресурсів, як різновиду програмного забезпечення, є багатограним і неоднозначним, на що впливають такі умови як вид програмного забезпечення, що має бути розроблене, вимоги замовника, обрана методологія програмування, наявність і достатність ресурсів для виконання розробки з урахуванням заданих обмежень.

Практикуючі програмісти і проект-менеджери по-різному бачать процес розробки онлайн-ресурсів. Шафер Д.Ф., Фатрелл Р.Т., Шафер Л.І процес розробки визначають як розробку і подання логічно пов'язаної, технічної специфікації програмного середовища, включаючи структури даних, програмну архітектуру, подання інтерфейсу, а також процедурні деталі ([Д. Шафер, Фатрелл, & Л. Шафер, 2004, с. 288](#)).

Буч Г., Рамбо Д. та Якобсон І. виділяють такі фази життєвого циклу програмного засобу: визначення вимог, аналіз, проектування, розробка (реалізація) і тестування ([Якобсон, Буч, & Рамбо, 2002, с. 41](#)).

Ніколаєв С. В. зазначає, що в ідеальному випадку процес розробки програмного забезпечення постає як послідовне виконання ряду етапів. Він відкидає етап аналізу, але додає такі етапи, як компонування і документування (Ніколаєв, 2002, с. 55-57):

1. Визначення вимог – у результаті виконання цього етапу потрібно зрозуміти, що саме потрібно зробити (створити), визначити комплекс вимог за основними характеристиками і документально відобразити це у вигляді технічного завдання на розробку системи.

2. Етап проектування – передбачає вирішення основних системних питань, серед яких розбиття на підсистеми, розробка сценарію роботи (у тому числі й діалогу з користувачем), структури даних, функціональної структури та ієрархії розбиття на програмні модулі.

3. Етап програмування – має за мету одержання вихідних текстів програмних модулів на будь якій алгоритмічній мові (іноді на кількох) і включає в себе алгоритмізацію, кодування (запис вихідного тексту алгоритмічною мовою) і налагодження.

4. Компонування – це скріплення окремих частин програми (модулів, процедур, функцій, підпрограм), написаних зазвичай різними людьми або навіть групами, в одну пов'язану програмну систему.

5. Тестування – має за мету визначити ступінь правильності вже готових програм після об'єднання їх в процесі програмування і компонування. Оскільки для складних програм повне тестування неможливо (внаслідок того ж чинника комбінаторної складності), то основне завдання на цьому етапі – це підвищення інформативності тестів і визначення їх місця в ієрархії залежно від наслідків, викликаних помилками, що виявляються.

6. Етап складання документації передбачає створення вичерпного комплексу документів, який в ідеалі повинен містити повну проектну та експлуатаційну документацію.

У процесі реальної розробки з'являються численні повернення і повтори (рис. 1).

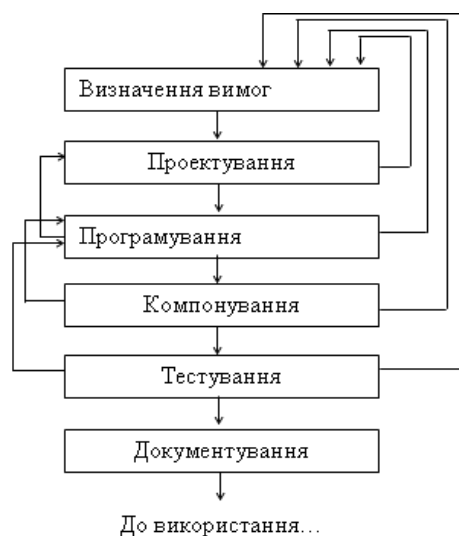


Рис. 1. Етапи розробки програмного забезпечення

Від вибору платформи для розробки залежить як сама по собі реалізація необхідного функціоналу, так і можливості подальшого розвитку веб-проекту. Для створення сайту, як правило, вибирається одна з платформ: CMS, фреймворк або SaaS-рішення.

Найбільш поширений варіант – це розробка на CMS-платформах.

Система управління сайтом (Content Management System) – це програмний продукт, який служить для розробки деяких стандартних різновидів сайтів. Майже всі CMS модульні, а модулі багатьох з них зібрані в комплекти, призначені для тих чи інших видів сайтів. Є коробкові CMS для простих сайтів, каталогів, інтернет-магазинів, блогів, порталів новин й інших видів сайтів.

Підтримка сайтів на CMS простіша, ніж робота з системами написаними «з нуля»: цьому сприяє як наявність документації, так і високий професійний рівень розробників CMS, який знаходить своє відображення в логічній і зрозумілій архітектурі системи.

Також наразі є популярним створення сайтів на SaaS-платформах. SaaS-платформа для створення сайтів – це можливість запустити досить простий веб-проект дуже швидко і вкрай дешево. Рішення підходить для простих сайтів, тимчасових проектів і для перевірки бізнес-ідей. SaaS-платформи, як і CMS, бувають специфічними (тільки для інтернет-магазинів, наприклад) і універсальними (для всіх типових видів сайтів).

На SaaS майданчиках має сенс створювати прості сайти, які не особливо вимогливі до дизайну й ні відразу, ні в подальшому не будуть вимагати доопрацювань програмної логіки.

У процесі розробки сайту ми стикнулися із такими завданнями, як, наприклад, форма реєстрації чи авторизації, можливість завантаження файлів та перевірка їх типів, розмірів і установки шляху завантаження файлу тощо. Для цього можна користуватися готовими власними бібліотеками або бібліотеками сторонніх розробників. Проте існують також рішення, інструменти, що полегшують роботу програміста, і що роблять розробку проекту більш передбачуваною і професійною. Ці рішення отримали назву Framework або каркасні системи для управління вмістом.

Framework (фреймворк, програмний каркас) – структура програмної системи, програмне забезпечення, що полегшує розробку і об'єднання різних компонентів великого програмного проекту. На відміну від бібліотек, які об'єднують набір підпрограм близькою функціональністю, фреймворк містить у собі велику кількість різних за призначенням бібліотек. Паралельно вживається термін каркас або програмний каркас.

Семенець С. В. наголошує на тому, що використання програмних каркасів дозволяє значно скоротити час створення програми. Метою конкретного програмного каркаса є визначення архітектури заданого класу систем або надання безлічі повторно використовуваних компонентів. Програмні каркаси використовуються як шаблони при створенні нової моделі ([Семенець, 2004, с. 133](#)).

Фреймворк має вбудовану підтримку таких баз даних як PostgreSQL, SQLite, MySQL, MSSQL або Oracle. Найбільш часто використовується реляційна база даних

MySQL, яка є безкоштовною, але водночас потужною щодо зберігання та обробки табличних даних ([Шаров, & Осадчий, 2014, с. 186](#)).

Серед фреймворків можна виділити фреймворки програмних систем, фреймворки додатку, фреймворк концептуальної моделі.

Фреймворк програмної системи (або підсистеми) – каркас програмної системи, що може включати допоміжні програми, бібліотеки коду, мову сценаріїв та інше програмне забезпечення, що полегшує розробку і об'єднання різних компонентів великого програмного пакету. Звичайно об'єднання відбувається за рахунок використання єдиного API. Прикладом може бути система керування контентом (CMS), веб-фреймворк Symfony.

Фреймворки додатку – мають стандартну структуру з метою створення засобів для автоматичного створення графічних інтерфейсів. Для забезпечення каркаса зазвичай використовують техніку об'єктно-орієнтованого програмування, наприклад, частини програми можуть успадковуватися від базових класів фреймворка. Можна назвати такі приклади фреймворків додатків: комерційний – MacApp, Metrowerks Powerplant і MacZoop під Macintosh, .NET Framework від Microsoft для Windows, вільні фреймворки, що є частинами проектів Mozilla, OpenOffice.org, GNOME і KDE; а також кросплатформні каркаси для операційних систем Linux, Macintosh и Windows – widget toolkit, wxWidgets, Qt, MyCore або FOX toolkit.

Фреймворк концептуальної моделі – абстрактне поняття структури, яке використовується у дослідженнях для визначення можливих способів вирішення проблеми, або подання ідеї ([Фреймворк, н.д.](#)).

Окремо можна виділити веб-фреймворки, тобто каркаси, які орієнтовані на клієнт-серверну технологію. Серед них виділяють фреймворки на ASP.NET, ColdFusion, Java, JavaScript, Perl, PHP, Python, Ruby та інших мовах програмування. Особливе місце займають php-фреймворки: CakePHP, CodeIgniter, Drupal, Horde, Joomla!, Kassel CMS, Kohana, Lithium, Midgard, MODx, Qcodo, Seagull, SilverStripe, Symfony, TYPO3, Xaraya, Yii, Zend Framework, Zeta Components. Їх використовують для створення сайтів різноманітно тематики і масштабу.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

З метою визначення швидкого, простого у вивченні, сучасного, безпечного php-фреймворку з вільною ліцензією був проведений аналіз Інтернет-джерел ([Количество строк кода, 2011](#); [Современные PHP-фреймворки, н.д.](#); [XHprof, 2009](#)). Нами були розглянуті фреймворки CakePHP, Code Igniter, Kohana, Symfony, Yii, Zend Framework і складена їх порівняльна таблиця ([табл. 1](#)).

З аналізу таблиці видно, що Yii Framework має більшість переваг. До того ж він не є громіздким, має якісну підсистему відкладеної ініціалізації.

Yii заснований на парадигмі MVC. Основна перевага – дуже висока швидкість роботи і, як наслідок, продуктивність. Фреймворк активно розвивається спільнотою. Yii не виглядає «монстром» у порівнянні з фреймворками Symfony і Zend Framework, кодова база яких вельми об'ємна. Фреймворк Yii досить простий в освоєнні й

у використанні, що сприяє швидкій розробці проектів на ньому. Однак при виборі в якості платформи для створення веб-проекту цього фреймворка варто все одно враховувати той факт, що швидкість розробки на ньому типових рішень все ж таки буде нижче, ніж розробка на CMS. Як і всі фреймворки, Yii «заточений» під розробку технічно складних веб-проектів: бізнес-додатків, веб-сервісів, а також сайтів зі складною бізнес-логікою і вимогливих до швидкодії.

Таблиця 1

Порівняльна таблиця найбільш популярних фреймворків

Параметр	CakePHP 2.0.4	Kohana 3.2	Symfony 2.0	Yii 2.0	Zend Framework 1.11
Короткий опис	Відкритий, реалізує патерн «Модель-Вид-Контролер», повністю сумісний як з PHP4, так і з PHP5.	Відкритий, реалізує патерн «Ієрархічні Модель-Вид-Контролер». Сумісний з PHP5.	Вільний, реалізує патерн «Модель-Вид-Контролер». Сумісний з PHP5.	Відкритий, вільний, реалізує патерн «Модель-Вид-Контролер». Сумісний з PHP5. Має інтеграцію з jQuery.	Вільний, відкритий, реалізує патерн «Модель-Вид-Контролер». PHP 5 і E_STRICT сумісний.
Аналіз	Багато вбудованого функціоналу. Всі компоненти досить тісно інтегровано.	Швидкий, гнучкий. Свій підхід до модульності. Погана документація.	Активно використовує командний рядок уаml. Потужний ORM, хороша система view, генератори коду, dependency injection для всього. Складний у вивченні.	Простіше у вивченні, ніж Zend і Symfony. Увібрав краще від Rails. Непоганий AR, хороша система view, генератори коду. Досить тісна інтеграція частин.	Академічно грамотний код. Дуже гнучкий. Вимагає гарного знання PHP і ООП і налаштування відповідно до своїх потреб перед тим, як використовуват и. Досить повна технічна документація.
Необхідний рівень знань	PHP, ООП, вміння розбиратися у вихідному кодї	PHP5, ООП	PHP5, ООП, ORM, консоль	PHP5, ООП	PHP5, ООП, шаблони проектування
Масштаби проектів	Маленькі – середні	Маленькі – великі	Великі	Маленькі – великі	Середні – великі
PHP5.4	Так	Так	Нї	Так	Так
Жорстка структура каталогів	Так	Так	Так	Нї (рекомендації)	Нї (рекомендації)
Офіційна підтримка інтернації	Так	Так	Так	Так	Так

Параметр	CakePHP 2.0.4	Kohana 3.2	Symfony 2.0	Yii 2.0	Zend Framework 1.11
Складність установки і налаштування	Низька	Низька	Висока	Середня	Висока
Потреба в налаштуванні	Небагато	Небагато	Багато	Небагато	Багато
Повна підтримка ORM	Так (не зручна)	Так	Так (Propel, Doctrine)	Active Record	Ні
Unit-тести для вихідного коду	Так	Не відомо	Так	Так	Так
Англомовне співтовариство	Так	Форум, блоги	Так	Так	Так
Ліцензія	MIT	BSD-style	MIT	New BSD	New BSD

З аналізу таблиці видно, що Yii Framework має більшість переваг. До того ж він не є громіздким, має якісну підсистему відкладеної ініціалізації.

Yii заснований на парадигмі MVC. Основна перевага – дуже висока швидкість роботи і, як наслідок, продуктивність. Фреймворк активно розвивається спільнотою. Yii не виглядає «монстром» у порівнянні з фреймворками Symfony і Zend Framework, кодова база яких вельми об'ємна. Фреймворк Yii досить простий в освоєнні й у використанні, що сприяє швидкій розробці проектів на ньому. Однак при виборі якості платформи для створення веб-проекту цього фреймворка варто все одно враховувати той факт, що швидкість розробки на ньому типових рішень все ж таки буде нижче, ніж розробка на CMS. Як і всі фреймворки, Yii «заточений» під розробку технічно складних веб-проектів: бізнес-додатків, веб-сервісів, а також сайтів зі складною бізнес-логікою і вимогливих до швидкодії.

Yii Framework складається з кількох основних частин: вхідний файл, додаток, контролер, дія, модель, представлення, макет, віджет, компонент, модуль.

Вхідний файл (зазвичай index.php) – це єдиний скрипт, з яким контактує користувач. Він читає конфігурацію і створює екземпляр класу CWebApplication.

Додаток (екземпляр класу CWebApplication) обробляє запит користувача, виконує роутинг і передає його на виконання відповідного контролера.

Контролер (екземпляр класу CController або похідного від нього) визначає основну логіку програми, взаємодіє з моделями і відображеннями.

Дія – це тип дії, що виконується користувачем, такі, як перегляд статті у блозі або надсилання коментаря.

Модель являє собою сутність даних. Наприклад, таблиця в базі даних (CActiveRecord) або форма на веб-сторінці (CForm). Вона займається безпосередньою обробкою даних: створенням, отриманням, зміною. Також представляє інтерфейс доступу до валідатора (validator), що перевіряє коректність даних, введених

користувачем. Модель зазвичай має ім'я, що відповідає таблиці в базі даних, яку вона представляє.

Представлення генерує готові частини сторінки, що віддається користувачу. Воно не використовує логіки, крім як умов і циклів. Подання для кожного контролера зазвичай роздільні.

Макет – це спеціальне подання для вставки інших уявлень. Зазвичай він містить частини інтерфейсу користувача, що використовується іншими уявленнями. Наприклад, основний макет може містити в собі шапку і підвал сторінки, місця для підключення інших уявлень і віджетів. За замовчуванням використовується макет `main.php`.

Віджет – це компонент для генерації самодостатнього елемента інтерфейсу користувача (наприклад, верхнє меню в тестовому додатку). У `yii` (бібліотеці розширень, що поставляються разом з `Yii`) є велика кількість готових віджетів.

Компонент представляє сутність будь-якого аспекту веб-додатка. Наприклад, авторизація користувачів. В `Yii` є багато готових компонентів, які можуть використовуватися як у готовому вигляді, так і розширюватися веб-розробниками.

Модуль – це самодостатня одиниця, що складається з контролерів, моделей, відображень тощо. Модулі використовуються для поділу сайту на логічні частини, наприклад форум і блог.

Основні переваги і можливості фреймворка `Yii`:

- забезпечує високу продуктивність порівняно з іншими `php`-фреймворками;
- заснований на парадигмі MVC (Модель-Представлення-Контролер);
- є інтерфейси `DAO` і `ActiveRecord` для роботи з базами даних (використовується `PDO`);
- підтримує інтернаціоналізацію;
- дозволяє кешувати як сторінки цілком, так і окремі фрагменти;
- здійснює перехоплення й обробку помилок;
- має функціонал роботи з формами, забезпечує їх побудову та валідацію;
- реалізовано аутентифікацію і авторизацію;
- зручний для реалізації `AJAX`-інтерфейсів, інтегрується з `jQuery`;
- у фреймворк вбудовані генератори базового `PHP`-коду для `CRUD`-операцій (скаффолдинг);
- підтримує теми оформлення;
- має можливість підключення сторонніх бібліотек;
- працює з міграціями баз даних (генерація, застосування і відкат);
- дозволяє здійснювати автоматичне тестування і вести розробку в стилі `TDD`;
- підтримує стиль `REST`.

Робота над створенням персонального сайту сучасного науковця велася у декілька етапів:

1. Проектування. Мета проектної роботи – спрогнозувати або задати поведінку користувача на сайті. Зрозуміла структура сторінок, доступність основного і допоміжного меню дозволяють утримувати користувачів на сайті. Тому довелося враховувати всі деталі до розробки дизайну та програмування.

Для початку роботи над створенням персонального сайту сучасного науковця було розроблене технічне завдання та опис основних вимог до нього. Був розроблений макет та вибране програмне забезпечення (фреймворк) для майбутнього персонального сайту сучасного науковця.

В результаті проектування дало чітке уявлення про те, що і як буде працювати, в якому місці будуть розташовуватися функціональні елементи та інформаційні блоки.

2. Розробка дизайну. Дизайн сайту повинен адекватно відображати характер діяльності замовника, органічно поєднуватися з інформаційним наповненням, структурою сайту і принципами організації інформації. Тому, якщо на етапі проектування визначалася поведінка користувачів, то тут оформлялися графічні елементи.

3. Верстка. Верстка є процес інтеграції текстового вмісту, графіки і програмних компонентів в єдине ціле, тобто надання сторінкам остаточного вигляду. У процесі верстки сторінки набули вигляду, в якому вони постали перед кінцевим користувачем (за винятком інформаційного наповнення). На даному етапі відбувався додатковий контроль якості виконуваної роботи, проводилася оптимізація web-сторінок під особливості конкретних браузерів, що використовуються відвідувачами сайтів для навігації по Інтернету. Враховувалися особливості подання сторінок при різних настройках кольору.

4. Програмування. На даному етапі відбувалася розробка і підключення програмних компонентів сайту, які повинні забезпечити відвідувачів необхідними функціональними можливостями. Саме цей етап був найбільш складним у реалізації у технологічному плані. Більшість рішень, що розроблялися на даному етапі, ґрунтуються на технологіях роботи з базами даних і на побудові динамічно генерованих сторінок сайту на основі інформації, що міститься в базі даних сайту. Нами було розроблено такі сторінки: «Головна», «Біографія», «Діяльність», «Кафедра», «Студентам», «Новини», «Контакти».

5. Наповнення сайту. Ефективність роботи будь-якого сайту нерозривно пов'язана з поняттям контенту, тобто текстової та графічної інформації, розміщеної на ньому. Тому найтривалішим етапом було заповнення персонального сайту інформацією, що здійснювалось через адміністративну панель.

6. Тестування. Тестуванню отриманого продукту до моменту появи його в мережі ми приділили особливу увагу. На цьому етапі перевірялися зручність навігації, цілісність даних, коректність посилань і орфографія. Альфа-тестування полягало у тому, що було перевірено функціональність відповідного розробленого інформаційного ресурсу технічному завданню. Були усунуті знайдені помилки, виправлено функціонал, налагоджено роботу сайту. Бета-тестування здійснювалось

викладачами кафедри інформатики і кібернетики, які перевіряли зручність використання створеного інформаційного ресурсу.

7. Публікація. Публікація – це забезпечення хостингу інтернет-сайту і «прив'язка» сайту до попередньо зареєстрованого доменного імені. В результаті персональний сайт сучасного науковця став доступним усім користувачам мережі Інтернет.

8. Просування (розкрутка). Готовий, наповнений інформацією сайт ще не гарантує збільшення кількості відвідувачів. Головна причина цього – користувачі Інтернет можуть не зустріти ваш сайт в мережі. За статистикою більшість користувачів знаходять інформацію і приходять на сайти через пошукові системи. Саме тому популярність ресурсу часто залежить від його позиції за результатами пошуку. Щоб персональний сайт сучасного науковця займав лідируючі позиції в пошукових системах необхідно провести ряд заходів по його просуванню в мережі. Тобто зараз проводиться свого роду «рекламна кампанія» з впізнавання сайту і підвищення його відвідуваності. Сюди входить реєстрація сайту в пошукових системах, обмін посиланнями, банерна реклама тощо.

9. Підтримка. Підтримка сайту – об'ємна і рутинна робота, спрямована на підтримку працездатності сайту на інформаційному і технічному рівнях. Адміністрування сайту вимагає уважності, точності, акуратності, пунктуальності, знань інтернет-маркетингу, копірайтингу, пошукової оптимізації, верстки.

Пропонуємо інструкцію та рекомендації щодо адміністрування та використання програмного продукту.

Вхід до адміністративного розділу авторизований ([рис. 2](#)).

Головна сторінка Кафедра Опис кафедри Працівн

Home / Login

Login

Please fill out the following fields to login:

Username

Username cannot be blank.

Password

Remember Me

Login

Рис. 2. Авторизація для входу до адміністративного розділу

Після авторизації у адміністратора з'являється меню для роботи із сайтом ([рис. 3](#)). Меню дозволяє додавати та переглядати: новини, статичні сторінки та завантажувати зображення.

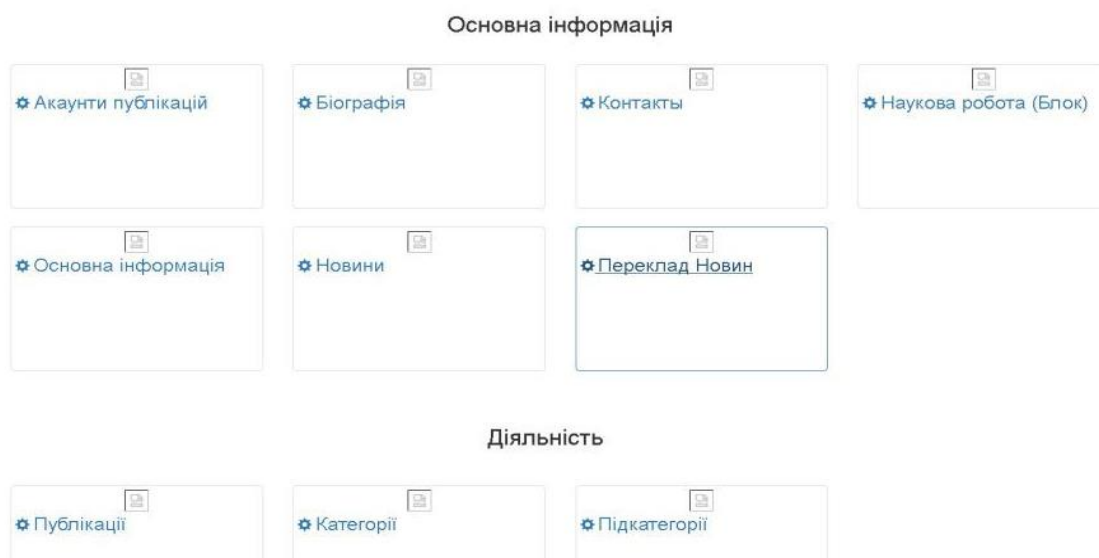


Рис. 3. Головне меню

Додавання інформації про науковця, його фото здійснюється після входу до пункту меню «Основна інформація» (рис. 4).

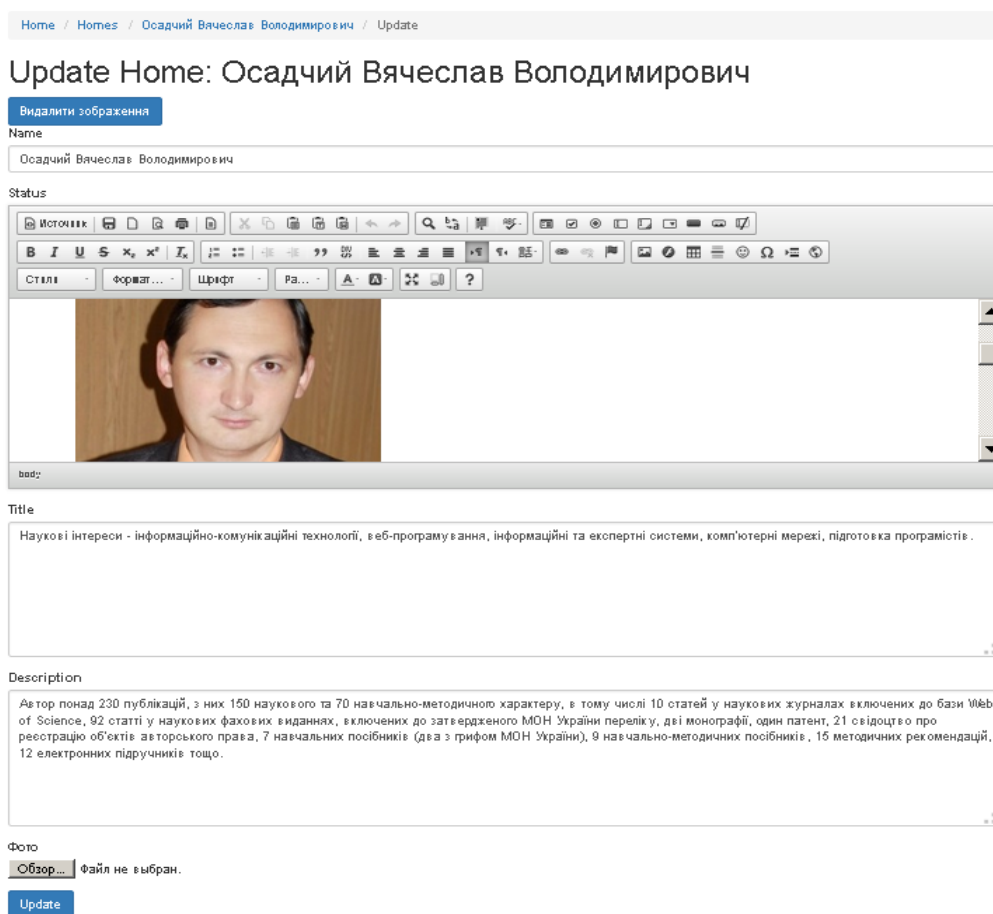
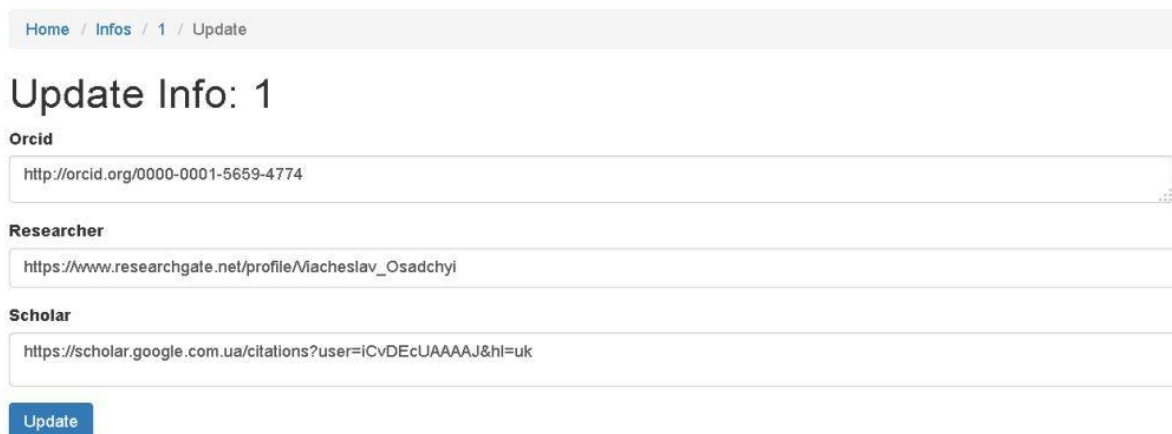


Рис. 4. Додавання інформації про науковця

Пункт головного меню «Акаунти публікацій» дозволяє створити посилання на ORCID, ResearchID та Google Scholar (рис. 5).



Home / Infos / 1 / Update

Update Info: 1

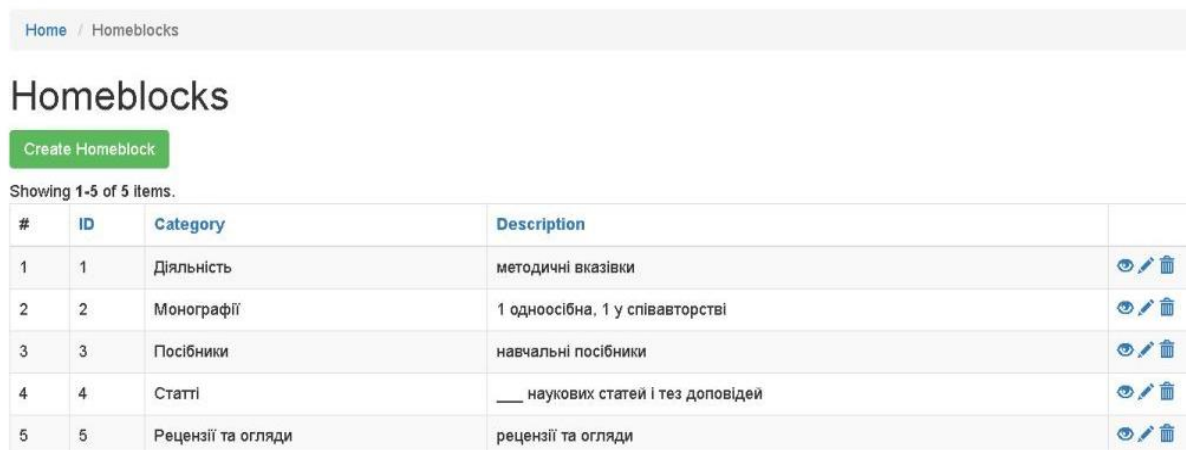
Orcid

Researcher

Scholar

Рис. 5. Посилання на інші ресурси

У блоці «Наукова робота» можна керувати списками публікацій науковця: додавати чи видаляти категорії публікацій, тут же додається опис категорії (рис. 6).



Home / Homeblocks

Homeblocks

Showing 1-5 of 5 items.
















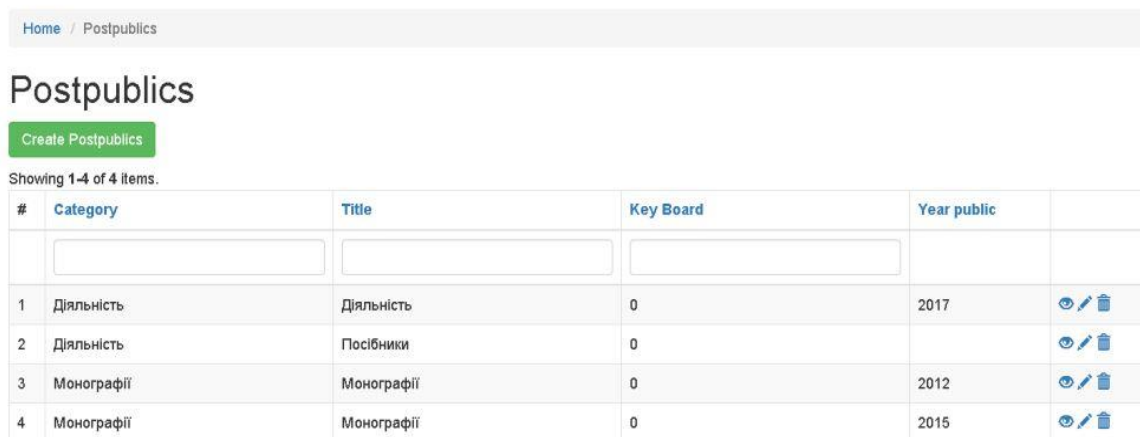
#	ID	Category	Description	
1	1	Діяльність	методичні вказівки	  
2	2	Монографії	1 одноосібна, 1 у співавторстві	  
3	3	Посібники	навчальні посібники	  
4	4	Статті	___ наукових статей і тез доповідей	  
5	5	Рецензії та огляди	рецензії та огляди	  

Рис. 6. Керування категоріями публікацій

Розміщення списку самих публікацій та посилань на них здійснюється при переході до пункту «Публікації» (рис. 7).

Розглянемо докладніше на прикладі додавання монографії. Обираємо категорію (Category) та заголовок (Title), до поля Опис (Description) додається бібліографічне посилання, а до поля Посилання (Links) адреса місця фактичного розташування публікації, заповнюємо поле Рік публікації (Year public). По закінченні натискаємо кнопку Update (рис. 8).



Home / Postpublics

Postpublics

Create Postpublics

Showing 1-4 of 4 items.









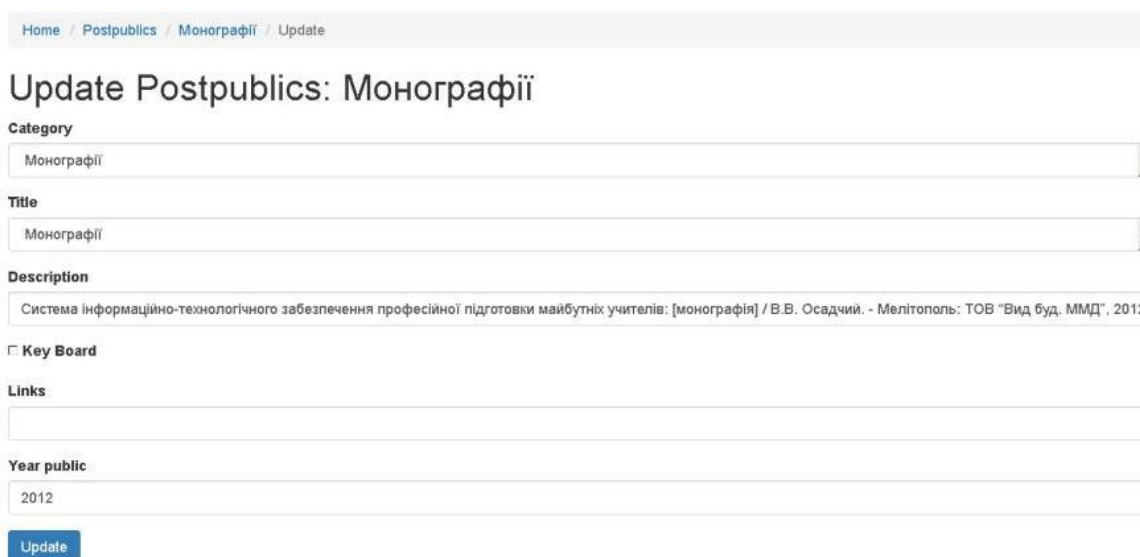
#	Category	Title	Key Board	Year public	
	<input type="text"/>	<input type="text"/>	<input type="text"/>		
1	Діяльність	Діяльність	0	2017	 
2	Діяльність	Посібники	0		 
3	Монографії	Монографії	0	2012	 
4	Монографії	Монографії	0	2015	 

Рис. 7. Керування списком публікацій



Home / Postpublics / Монографії / Update

Update Postpublics: Монографії

Category
Монографії

Title
Монографії

Description
Система інформаційно-технологічного забезпечення професійної підготовки майбутніх учителів: [монографія] / В.В. Осадчий. - Мелітополь: ТОВ "Вид буд. ММД", 2012

Key Board

Links

Year public
2012

Update

Рис. 8. Додавання публікації

Аналогічно додаються посібники, статті, методичні вказівки тощо.

До сторінки «Студентам» додаються посилання на курси, які веде науковець. Можуть додаватися плани і конспекти лекцій та інші навчально-методичні матеріали ([рис. 9](#)).

Розглянемо докладніше на прикладі додавання посилання на курс. Обираємо категорію (Category), до поля Опис (Description) додається назва курсу та за бажанням коротка анотація, а до поля Посилання (Links) адреса місця фактичного розташування, обираємо курс, на якому за навчальним планом вивчається дисципліна (Kurs). По закінченні натискаємо кнопку Update ([рис. 10](#)).

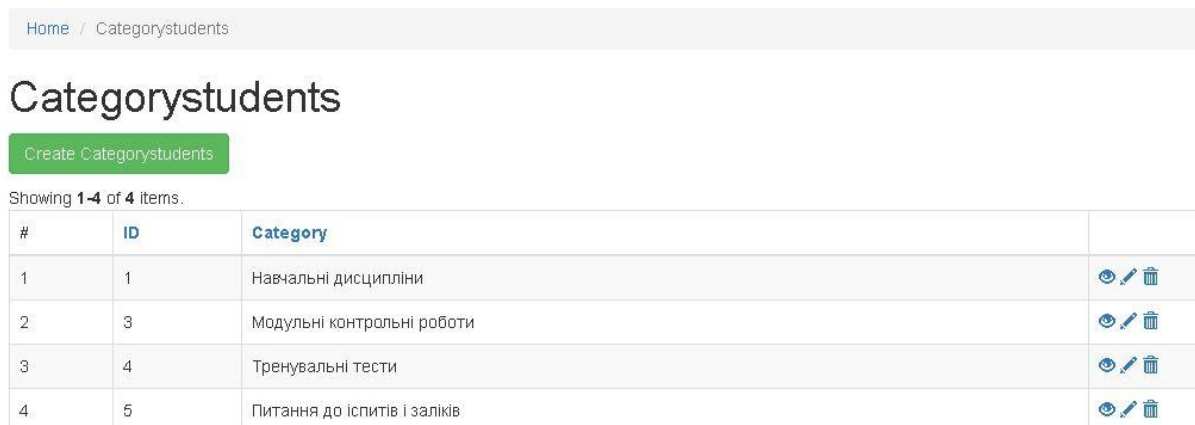


Рис. 9. Додавання інформації для студентів

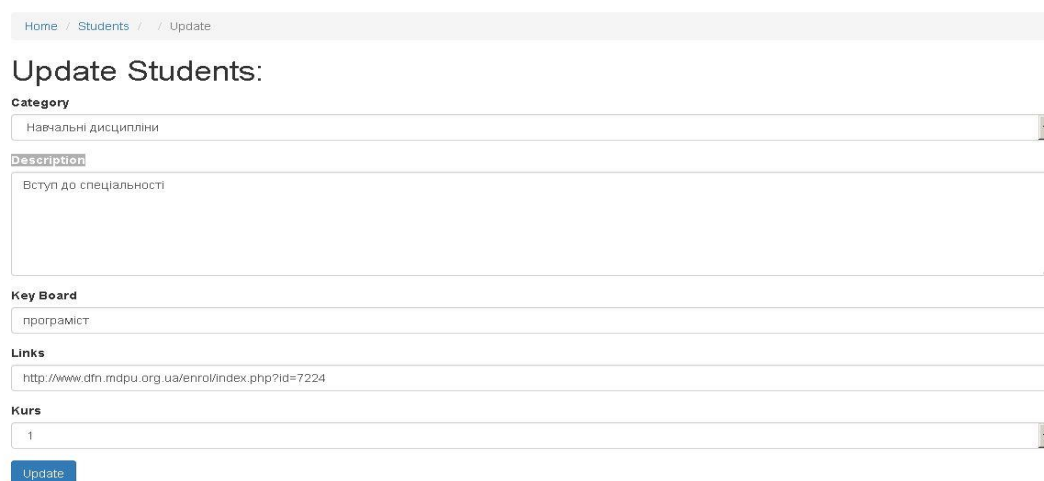


Рис. 10. Додавання посилання до навчальної дисципліни

Отже, розроблений нами персональний сайт науковця містить такі розділи: головна сторінка; блок новин; біографія; діяльність; інформація про кафедру; розділ для студентів; контакти.

На головній сторінці персонального сайту розміщено фото, стислі відомості про науковця та його наукові інтереси, а також посилання на ORCID, ResearcherID та Google Scholar (рис. 11).

Верхнє меню дозволяє перейти на сторінку з каталогом публікацій науковця (рис. 12), де можна знайти більш докладну інформацію за типами: монографії, посібники, методичні вказівки тощо (рис. 13).

На персональному сайті науковця також можна знайти інформацію про структурний підрозділ (кафедру), яке є основним місцем роботи науковця.

Цей розділ присвячено кафедрі інформатики і кібернетики Мелітопольського державного педагогічного університету імені Богдана Хмельницького. Тут подано основну інформацію про діяльність кафедри, перелік спеціальностей, для яких кафедра є випусковою, вказано на яких посадах можуть працювати випускники.

Осадчий В.В.
Персональний сайт

ГОЛОВНА НОВИНИ БІОГРАФІЯ ДІЯЛЬНІСТЬ КАФЕДРА СТУДЕНТАМ КОНТАКТИ

Осадчий Вячеслав Володимирович

Доктор педагогічних наук, професор, завідувач кафедри інформатики і кібернетики Мелітопольського державного педагогічного університету імені Богдана Хмельницького

БІОГРАФІЯ

Наукові інтереси - інформаційно-комунікаційні технології, веб-програмування, інформаційні та експертні системи, комп'ютерні мережі, підготовка програмістів.

Автор понад 230 публікацій, з них 150 наукового та 70 навчально-методичного характеру, в тому числі 10 статей у наукових журналах включених до бази Web of Science, 92 статті наукових фахових виданнях, включених до затвердженого МОН України переліку, дві монографії, один патент, 21 свідоцтво про реєстрацію об'єктів авторського права, 7 навчальних посібників (два з грифом МОН України), 9 навчально-методичних посібників, 15 методичних рекомендацій, 12 електронних підручників тощо.

Монографії (2)
1 одноосібна, 1 у співавторстві

Посібники (14)
Навчальні та навчально-методичні посібники

Методичні рекомендації (20)
Методичні рекомендації для забезпечення якісної підготовки студентів

Охоронні документи (0)
Патенти та авторські свідоцтва

Поділитись у соціальних мережах

ORCID
ORCID (англ. Open Researcher and Contributor ID) – єдиний міжнародний реєструючий для коректного цитування статей.
ПЕРЕЙТИ

Researcher
ResearcherID – ідентифікатор ученого (дослідника), що дає змогу формувати список власних публікацій.
ПЕРЕЙТИ

Google Scholar
Академія Google (англ. Google Scholar) - безкоштовна пошукова система за текстами наукових публікацій.
ПЕРЕЙТИ

Рис. 11. Головна сторінка сайту

Осадчий В.В.
Персональний сайт

ГОЛОВНА НОВИНИ БІОГРАФІЯ ДІЯЛЬНІСТЬ КАФЕДРА СТУДЕНТАМ КОНТАКТИ

ГОЛОВНА / ДІЯЛЬНІСТЬ

Каталог публікацій

А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Э Ю Я

Монографії (2)
Посібники (14)
Методичні рекомендації (20)

Назва Публікації

ЗНАЙТИ

Монографії
□ Монографії (2)

Посібники
□ Посібники (14)

Рис. 12. Каталог публікацій науковця

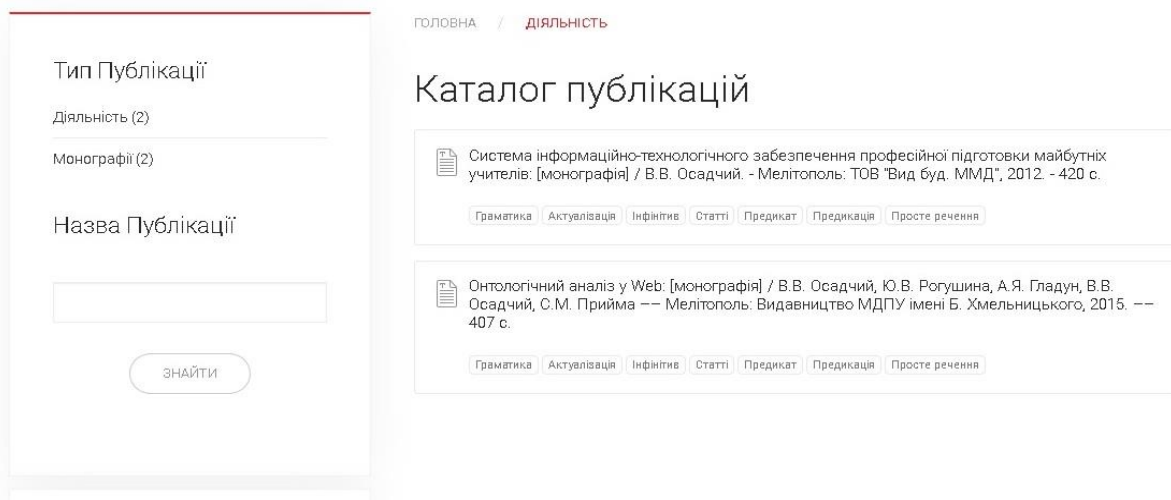


Рис. 13. Список монографій автора

Із верхнього меню можна здійснити перехід до розділу, де розміщені посилання на дисципліни, які викладає науковець (рис. 14). До цього розділу можуть бути додані повнотекстові конспекти лекцій, програми підготовки до атестації, приклади виконання індивідуальних завдань та інші навчально-методичні матеріали. Тобто все те, що науковець, як викладач, вважає за доцільне оприлюднити.

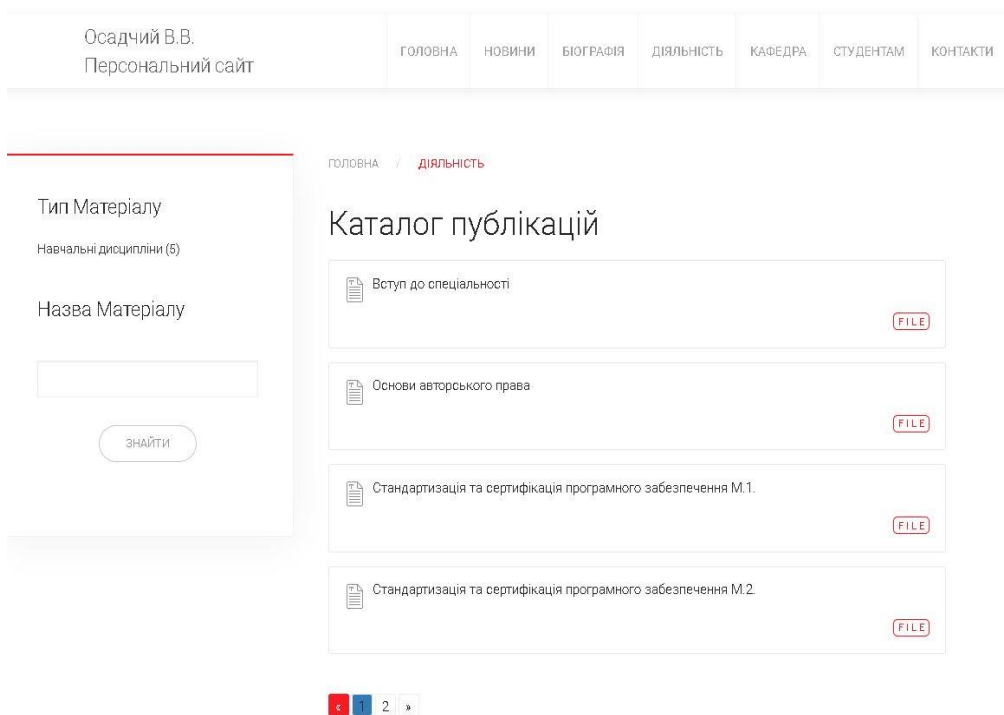


Рис. 14. Посилання на курси


Розділ «Контактна інформація», що доступний з головного меню включає в себе: інформацію про науковця, його номер телефону, E-mail. Також безпосередньо з цієї сторінки можна скористатися формою для листування (рис. 15).

Осадчий В.В.
Персональний сайт

ГОЛОВНА НОВИНИ БІОГРАФІЯ ДІЯЛЬНІСТЬ КАФЕДРА СТУДЕНТАМ КОНТАКТИ

Контактна Інформація

Ви можете скористатися формою для листування безпосередньо з цієї сторінки. Для цього необхідно заповнити всі, без винятку, поля контактної форми.


Осадчий Вячеслав
Володимирович

Доктор педагогічних наук,
професор, завідувач кафедри
інформатики і кібернетики
Мелітопольського державного
педагогічного університету імені
Богдана Хмельницького

+38 (097) 930 86 18
poltom55@gmail.com

Надіслати лист

Ваше ім'я:

Поштова адреса:

Заголовок листа:

Текст:

Введіть текст який на малюнку




Рис. 15. Контактна інформація з формою для листування

Для того, щоб надіслати листа потрібно: ввести своє ім'я, електронну адресу, тему повідомлення, текст повідомлення, код перевірки і натиснути кнопку «Надіслати».

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Отже, персональну веб-сторінку сьогодні має сенс створити кожному науковцю – її існування спрощує багато форм професійних контактів, дозволяє конструктивно заявити про себе, окреслити коло своїх інтересів і можливостей.

Розроблений персональний сайт сучасного науковця є значним джерелом біографічної та бібліографічної інформації, дає можливість представити міжнародній науковій спільноті відомості про свою діяльність та ідеї у вигляді опублікованих і неопублікованих наукових праць, що дає можливість ученому вирішувати питання створення власного іміджу в науковому просторі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- Веб-сайт: визначення й застосування. Статті про розробку та просування веб-сайтів. (2011). *Webtec.com.ua*. Взято з <http://www.webtec.com.ua/uk/articles/index/view/2011-05-05/web-site>.
- Данилюк, С. (2010). Персональний сайт як спосіб презентації особистості науковців. *Наукові записки Кіровоградського державного педагогічного університету імені Володимира Винниченка*. Серія: Філологічні науки, 89 (5), 190-194.
- Здановська, В. (2010). Персональні сторінки науковців як джерело біобібліографічної інформації. *Наукові праці Національної бібліотеки України ім. В. І. Вернадського*, 27, 31-35.
- Количество строк кода в популярных php-фреймворках. (2011). *Korzh.net*. Взято с <http://korzh.net/2011-04-kolichestvo-strok-koda-v-populyarnyx-php-frejmworkax.html>.
- Николаев, С. В. (2002). *Проектирование программного обеспечения: Учебное пособие*. Таганрог: Изд-во ТРТУ.
- Пастушенко, О. (2015). Розвиток системи персоніфікації вченого в інформаційному середовищі Інтернет. *Наукові праці Національної бібліотеки України ім. В. І. Вернадського*, 41, 295-309.
- Полилова, Т. А. (2011). Персональные веб-страницы в научном сообществе. В *Научный сервис в сети Интернет: экзафлопное будущее* (С. 476-479). Новороссийск.
- Савенкова, Л. В. (2014). Персональні сайти вчених у інформаційному середовищі університетської бібліотеки. *Вісник Одеського національного університету*. Серія: Бібліотекознавство. Бібліографознавство. Книгознавство, 19, 153-160.
- Семенец, С. В. (2004). Создание программного каркаса Windows CE. *Проблеми програмування*, 2,3, 133-137.
- Современные PHP-фреймворки - полный список лидеров | Редакция электронного журнала для веб-разработчиков. (н.д.). *Phpinside.ru*. Взято с <http://phpinside.ru/?q=node/841>.
- Фреймворк. (н.д.). *Ru.wikipedia.org*. Взято с <http://ru.wikipedia.org/wiki/Фреймворк>.
- XHprof: производительность популярных PHP фреймворков | Блог веб-разработчика. (2009). *Tigor.com.ua*. Взято с <http://tigor.com.ua/blog/2009/12/17/xhprof-php-framework-performance-comparision>.
- Шаров, С. В., & Осадчий, В. В. (2014). *Бази даних та інформаційні системи*. Навчальний посібник. Мелітополь: Вид-во МДПУ ім. Б. Хмельницького.
- Шафер, Д. Ф., Фатрелл, Р. Т., & Шафер, Л. И. (2004). *Управление программными проектами: достижение оптимального качества при минимуме затрат*. Москва: Вильямс.
- Якобсон, А., Буч, Г., & Рамбо Дж. (2002). *Унифицированный процесс разработки программного обеспечения*. Санкт-Петербург: Питер.

REFERENCES (TRANSLATED AND TRANSLITERATED)

- Website: Definition and Application. Articles on the development and promotion of websites. (2011). *Webtec.com.ua*. Retrieved from <http://www.webtec.com.ua/uk/articles/index/view/2011-05-05/web-site>. (in Ukrainian)
- Danyliuk, S. (2010). Personal website as a way to present a personality of scientists. *Naukovi zapysky Kirovohradskoho derzhavnogo pedahohichnoho universytetu imeni Volodymyra Vynnychenka*. Serii: Filolohichni nauky, 89 (5), 190-194. (in Ukrainian)

- Zdanovska, V. (2010). Personal pages of scientists as a source of bibliographic information. *Naukovi pratsi Natsionalnoi biblioteky Ukrainy im. V. I. Vernadskoho*, 27, 31-35. (in Ukrainian)
- Number of lines of code in popular php-frameworks. (2011). *Korzh.net*. Retrieved from <http://korzh.net/2011-04-kolichestvo-strok-koda-v-populyarnyx-php-frejmworkax.html>. (in Russian)
- Nikolaev, S. V. (2002). *Software Design: A Tutorial*. Taganrog: Izd-vo TRTU. (in Russian)
- Pastushenko, O. (2015). Development of the personification system of the scientist in the Internet information environment. *Naukovi pratsi Natsionalnoi biblioteky Ukrainy im. V. I. Vernadskoho*, 41, 295-309. (in Ukrainian)
- Polilova, T. A. (2011). Personal web pages in the scientific community. In *Nauchnyj servis v seti Internet: ehkzaflopsnoe budushchee* (pp. 476-479). Novorossijsk. (in Russian)
- Savenkova, L. V. (2014). The personal sites of scientists in the university library's information environment. *Visnyk Odeskoho natsionalnoho universytetu. Seriya: Bibliotekoznavstvo. Bibliohrafoznaustvo. Knyhoznaustvo*, 19, 153-160. (in Ukrainian)
- Semenets, S. V. (2004). Creating a software framework Windows CE. *Problemy prohramuvannia*, 2,3, 133-137. (in Russian)
- Modern PHP-frameworks – a complete list of leaders | Editorial of the electronic journal for web developers. (n.d.). *Phpinside.ru*. Retrieved from <http://phpinside.ru/?q=node/841>. (in Russian)
- Framework. (n.d.). *Ru.wikipedia.org*. Retrieved from <http://ru.wikipedia.org/wiki/Фреймворк>. (in Russian)
- XHprof: performance of popular PHP frameworks | Blog of the web developer. (2009). *Tigor.com.ua*. Retrieved from <http://tigor.com.ua/blog/2009/12/17/xhprof-php-framework-performance-comparision>. (in Russian)
- Sharov, S. V., & Osadchyi, V. V. (2014). *Databases and Information Systems. Tutorial*. Melitopol: Vyd-vo MDPU im. B. Khmelnytskoho. (in Ukrainian)
- Shafer, D. F., Fatrell, R. T., & Shafer, L. I. (2004). *Management of software projects: achieving optimal quality with minimum costs*. Moscow: Vil'yams. (in Russian)
- Yakobson, A., Buch, G., & Rambo Dzh. (2002). *Unified software development process*. Saint-Petersburg: Piter. (in Russian)