

ЗАТВЕРДЖЕНО
Наказ Міністерства освіти і науки,
молоді та спорту України
29 березня 2012 року № 384

Форма № Н-6.01

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Кафедра програмного забезпечення

(повна назва кафедри, циклової комісії)

КУРСОВИЙ ПРОЕКТ

з дисципліни «Проектний практикум»

(назва дисципліни)

на тему: Розробка програмного забезпечення сайту кафедри програмного
забезпечення

Студента 2 курсу 1ПІ-12мс (д/н) групи
напряму підготовки 6.050103 «Програмна інженерія»

Рудного М. М.

(прізвище та ініціали)

Керівник: к.т.н., доц. Войтко В.В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____

Кількість балів: _____ Оцінка: ECTS _____

Члени комісії

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

м. Вінниця – 2014 рік

ВСТУП.....	5
1 АНАЛІЗ СТАНУ ПИТАННЯ І ПОСТАНОВКА ЗАДАЧ РОЗРОБКИ	8
1.1 Визначення суті технічної проблеми	8
1.2 Аналіз сучасного стану та перспективи розвитку сайтів кафедр навчальних закладів.....	9
1.3 Аналіз існуючих сайтів кафедр.....	10
1.4 Постановка задач розробки	13
1.5 Висновки	14
2 РОЗРОБКА СТРУКТУРИ І ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ САЙТУ... ..	15
2.1 Аналіз принципів реалізації web-сайту.....	15
2.2 Інформаційне забезпечення головної сторінки.....	25
2.3 Інформаційне забезпечення сторінки про методичну роботу	26
2.4 Інформаційне забезпечення сторінки викладацького складу	27
2.5 Інформаційне забезпечення сторінки абітурієнту	28
2.6 Розробка ER-моделі бази даних.....	29
2.7 Висновки	31
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ САЙТУ	32
3.1 Вибір середовища розробки веб-сайту	32
3.2 Варіантний аналіз і обґрунтування вибору програмних засобів для реалізації веб-сайта.....	33
3.2.1 Аналіз можливостей ASP та ASP .NET	34
3.2.2 Аналіз можливостей JavaScript.....	35
3.2.3 Аналіз можливостей Perl	36
3.2.4 Аналіз можливостей PHP	37
3.2.5 Обґрунтування вибору засобу реалізації веб-сайту	39
3.3 Варіантний аналіз систем керування базами даних	39
3.3.1 Аналіз можливостей SQL	40
3.3.2 Аналіз можливостей MySQL	41
3.3.3 Аналіз можливостей Oracle Database	42
3.3.4 Аналіз можливостей PostgreSQL	43
3.3.5 Обґрунтування вибору систем керування базами даних	44

3.4 Розробка бази даних засобами MySQL	44
3.4.1 Налаштування модуля database.....	45
3.4.2 Розробка баз даних.....	46
3.5 Розробка модуля управління новинами	51
3.6 Висновки	55
4 ОХОРОНА ПРАЦІ	56
4.1 Характеристика професії та приміщення	56
4.2 Аналіз стану умов охорони праці в приміщенні.....	60
4.3 Розробка заходів щодо поліпшення умов праці.....	66
4.4 Висновки до розділу 4	66
5 ТЕСТУВАННЯ РОБОТИ ВЕБ-САЙТУ	68
5.1 Тестування адмін-панелі сайту	68
5.2 Тестування клієнтської частини сайту.....	69
5.3 Тестування користувацького інтерфейсу сайту.....	69
5.4 Тестування роботи бази даних.....	70
5.5 Висновки	72
ВИСНОВКИ.....	73
ПЕРЕЛІК ПОСИЛАНЬ	74
ДОДАТКИ.....	76
ДОДАТОК А. ВИХІДНИЙ ТЕКСТ МОДУЛЯ УПРАВЛІННЯ НОВИНАМИ.....	77
ДОДАТОК Б. ЛІСТИНГ ГОЛОВНОГО КОНТРОЛЕРА САЙТУ	92
ДОДАТОК В. ЛІСТИНГ ГОЛОВНОГО ШАБЛОНУ.....	95
ДОДАТОК Г. ІЛЮСТРАТИВНИЙ МАТЕРІАЛ ДО ЗАХОСТУ БАКАЛАВРСЬКОЇ ДИПЛОМНОЇ РОБОТИ	101

ВСТУП

Сучасною об'єктивною реальністю є широке впровадження у всі сфери життєдіяльності особи, суспільства та держави сучасних інформаційних технологій, розгортання на їх основі різноманітної складності локальних і глобальних інформаційних систем, призначених для прискорення обміну інформацією та доступу до різноманітних інформаційних джерел. Провівши аналіз сучасного розвитку інформаційних технологій і телекомунікацій, як один з критеріїв рівня розвитку держави та інструмент інтенсифікації всіх процесів у суспільстві, Україна, як і будь-яка інша країна, зацікавлена у розвитку інформаційної інфраструктури, в тому числі й шляхом використання у всіх сферах господарської діяльності можливостей, що надаються глобальними мережами передачі даних [1].

Всесвітня павутини являє собою надзвичайно перспективний засіб комунікації, що може пояснити його сучасний надзвичайний розвиток. Сьогоднішні засоби зв'язку дозволили об'єднати розрізнені комунікаційні системи у глобальну мережу. Завдяки цьому людина отримала можливість обмінюватися інформацією в межах всієї планети, не залежно від кордонів і відстаней. Саме глобальна мережа є однією з ознак переходу суспільства від індустріальної стадії розвитку до інформаційної [1].

За 40 років глобальна павутина - Інтернет переріс із секретної закритої технології, використовуваної військовими організаціями і навчальними закладами, що працюють на військові і розвідувальні програми, у загальнодоступну, відкриту систему, що стала впливати на розвиток усієї планети і окремої людини [2].

За останні декілька десятиріч років завдяки бурхливому розвитку Інтернету в програмуванні виділяють окремий напрямок – Web-програмування, тобто створення скриптів для Web, так як в час стрімкого прогресу просто красиво оформлений текст і картинки на веб-сайті вже нікого не здивують. Та й вимоги до сайтів дуже змінилися, і тепер для успішного представництва організації або

особи в Інтернет необхідно надати своїм відвідувачам різні можливості: контактні форми, каталоги продукції, підписки, розсилання, пошук по сайту й багато чого іншого. Інтерактивність та адаптивність сайту забезпечується за рахунок сучасних технологій веб-програмування в обличчі HTML5 і CSS3. Технології веб-програмування на стороні клієнта включають в себе набір різних засобів і мов програмування або мови розмітки тексту. Ці програмні засоби розділяються на два основні види: серверні і клієнтські. Серверні скрипти виконуються на стороні сервера, тобто того комп'ютера, на якому розміщені вихідні файли сайту і база даних. До серверних скриптів належать PHP, Perl, ASP.NET, NodeJS тощо. Вони виконуються ще до завантаження сторінок сайту на клієнтській стороні. У свою чергу, клієнтські скрипти виконуються на комп'ютері клієнта вже після завантаження сторінки із сервера й не вимагають її додаткового перезавантаження (Javascript, Flash, Vbscript і Silverlight) [2].

В нашій час великої популярності набувають так звані сайти-візитки окремих організацій, персон або підрозділів. У таких «онлайн» візитках зазвичай наводиться детальна або конкретна інформація про власника сайту, і наводиться невеликий перелік основних характеристик, такий як викладачі. Основним завданням таких сайтів є надання детальної інформації про місце навчання, в якому планується проведення тривалого часу. Перш ніж поступати в конкретний навчальний заклад, необхідно зібрати якомога більше інформації про заклад, кафедри і викладачів, де можна також переглянути детальну інформацію і біографію. Також зростає кількість людей, які воліють попередньо знайти всю необхідну інформацію в інтернеті. Охопити цей сегмент абітурієнтів звичайні вступні комісії не в змозі, тому що фізично розміщені на певних територіях. Але при наявності в кафедри власного сайту, саме такі відвідувачі складуть основу цільової аудиторії, що дозволить кафедрі збільшити кількість студентів [3].

Тому, розробка та впровадження Web-сайту кафедри програмного забезпечення є актуальною та доцільною.

Основним завданням є оновлення сайту кафедри програмного забезпечення до рівня сучасного розвитку інтернет технологій.

Предметом дослідження роботи є засоби реалізації Web-ресурсів.

Основними завданням є реалізація засобів розробки структури Web-сайту та інтерфейсу, вибір засобів для їх реалізації, програмну реалізацію web – сайту.

В першому розділі роботи проаналізовано сучасний стан розвитку інтернет-технологій та сайтів даної тематики. Досліджено основні функціональні можливості, якими повинен бути наділений сайт кафедри.

Другий розділ містить аналіз інформаційного забезпечення та проведено дослідження предметної області, виділено ключові розділи.

В третьому розділі описаний процес розробки сайту і його окремих складових. Також вибрані способи та засоби для створення даного проекту актуальним та таким, що відповідає сучасним тенденціям.

Четвертий розділ містить дані про охорону праці під час роботи з даним проектом та загальні правила.

До п'ятого розділу винесено один із останніх та не менш важливих етапів – а саме остаточне тестування сайту перед процесом запуску на робочому сервері.

Мета виконання бакалаврської роботи – покращення якості візуалізації інформації про кафедру програмного забезпечення Вінницького національного технічного університету шляхом створення сучасного веб-сайту.

Об'єктом дослідження виступають Web-технології.

Робота виконана на замовлення кафедри програмного забезпечення Вінницького національного технічного університету.

Виконання даної бакалаврської дипломної роботи дозволяє отримати практичні навички створення повноцінних веб-сайтів з використанням різноманітних за складністю і реалізацією систем керування контентом та використання заздалегідь розробленого дизайну.

1 АНАЛІЗ СТАНУ ПИТАННЯ І ПОСТАНОВКА ЗАДАЧ РОЗРОБКИ

1.1 Визначення суті технічної проблеми

Створення сайту - надзвичайно складний процес, тим більше якщо сайт особа вирішила робити самостійно. Ще донедавна розробка сайтів була доступна лише великим організаціям, підприємствам. Однак на сьогоднішній день розробка сайтів в Україні та за її межами – це популярна послуга, що доступна всім бажаючим. Попит на послуги розробки сайтів зростає і в освітній галузі. Сьогодні, все більшої популярності набирають інформаційні сайти окремих організацій або підрозділів [4].

Отже, визначимо проблеми, які вирішує сайт кафедри.

Для більшості абітурієнтів період подання документів до ВУЗів є «гарячим часом», за який потрібно визначитися з навчальним закладом, в який будуть подаватися документи для навчання. Для вибору закладу і відповідної кафедри необхідно детально ознайомитися з усіма деталями, пов'язаними з навчальним процесом. І тут виникає проблема недостачі інформації, яку необхідно знайти в дуже короткий час.

Офіційний сайт кафедри дає детальну інформацію про умови і терміни навчання, оперативно надає відвідувачам інформацію про вступну компанію, дату та місця подання документів, загальну і залишкову кількість місць для вступу на контрактну та державну форму навчання на денному або заочному відвідуванні.

Кожному абітурієнту цікаво знати детальну інформацію про викладачів, які будуть навчати його наступні роки на даній кафедрі. Цю інформацію з легкістю можна знайти на офіційному сайті кафедри, де вона доступно розідлена по викладачах і відповідних розділах їхньої діяльності. Також можна дізнатися загальну інформацію по кафедрі, а саме дані про методичну роботу, науково-дослідну роботу, дізнатися детальну інформацію кафедри. Основним можна

назвати можливість переглянути особисте життя кафедри на сторінках новин, де детально все описано в текстовому та графічному вигляді.

Сайт корисний не лише даними про кафедру і викладачів, які працюють на ній, а й детальною інформацією про те, як можна зв'язатися з представниками цієї кафедри, або як добратися до місця розташування. На даному сайті може бути розміщена інформація про партнерів та спонсорів кафедри, які надають фінансову допомогу, або допомогу іншого роду.

Останнім часом усе більше зростає кількість людей, які воліють попередньо знайти всю необхідну інформацію в інтернеті. Охопити цей сегмент споживачів звичайній "офлайн" кафедрі дуже непросто. Але при наявності в неї власного офіційно веб-сайту, саме такі відвідувачі складуть основу цільової аудиторії, що дозволить кафедрі і навчальному закладу збільшити кількість і якість студентів.

Отже, розробка офіційного сайту кафедри програмного забезпечення є актуальною і необхідною.

1.2 Аналіз сучасного стану та перспективи розвитку сайтів кафедр навчальних закладів

На даний момент назріла нагальна потреба в комплексному дослідженні Інтернет-ринку загалом. Мета дослідження - визначення складових частин концепції створення і функціонування сайтів-візиток кафедр навчальних закладів встановлення конкурентних, огляд проблем і перспектив напрямку, оцінка законодавчої бази в Україні щодо провадження цього виду діяльності і даного формату представлення інформації[5].

Першим об'єктом дослідження була база даних пошукової системи Google. Пошук інформації здійснювався з використанням фільтра – «Шукати тільки на території України», лише українською мовою, часовий діапазон в один рік. Станом на 14.04.14 р. у базі даних Яндекс містилося 41 сайт, які були релевантні відносно запиту «кафедра програмного забезпечення». Слід відмітити, що релевантність – це відповідність інформації сайту запиту користувача, «на думку»

алгоритму пошукової системи. Враховуючи дану примітку, реально існуючих сайтів кафедр програмного забезпечення набагато менше, оскільки в результаті пошуку отримується також інформація про каталоги сайтів, Інтернет-версії друкованих видань, оглядові публікації тощо.

У результаті PEST-аналізу було виділено декілька груп факторів (політичні, економічні, соціальні, технологічні), які істотно впливають на розвиток та впровадження комп'ютерних технологій та засобів телекомунікацій у практичну сферу.

До факторів політичного спрямування належать:

- відсутність в Україні визнання на законодавчому рівні необхідності наявності даних сайтів;
- підпорядкування роботи кафедр навчальним закладам, які не спішають впроваджувати сучасні технології.

Слід наголосити і на ряді інших економічних переваг сайтів, зокрема, вони є менш затратними в порівнянні зі «звичайними» представницькими стендами, дозволяють абітурієнту легко порівняти послуги різних навчальних закладів.

Безумовно, об'єм і якість сайтів кафедр навчальних закладів далекий від аналогічного у США чи ЄС, чисельність Інтернет-аудиторії теж менша, а головне, відсутність у значної частини населення усвідомлення того, що всесвітня мережа – це місце, де можна знайти потрібну інформацію для вступу в навчальний заклад. Однак, згідно статистичних даних, 45% Інтернет-аудиторії за сприятливих умов готові формувати думку на заклади через всесвітню мережу. Проте враховуючи постійно зростаючу чисельність української Інтернет-аудиторії, перспективність веб-сайтів кафедри різних навчальних закладів безперервно зростає.

1.3 Аналіз існуючих сайтів кафедр

На сьогодні в мережі Інтернет існує значна кількість Web-сайтів різних кафедр. Величезна кількість з них знаходиться в межах Вінницького національного технічного університету, але більшість, звичайно, по інших

навчальних закладах всієї країни. Особливої уваги потребують сайти кафедр нашого університету і устарівша версія сайту кафедри програмного забезпечення.

Розглянемо детальніше дані аналоги, визначимо їх переваги і недоліки. Найбільш підходящими для порівняння аналогами були визнані наступні:

1. Сайт <http://lot.vntu.edu.ua/>. Даний сайт наведено на рисунку 1.1.

Переваги:

- простий інтерфейс;
- інтерактивність сайту;
- вдало підібрані кольори сайту;
- багато мов для вибору;

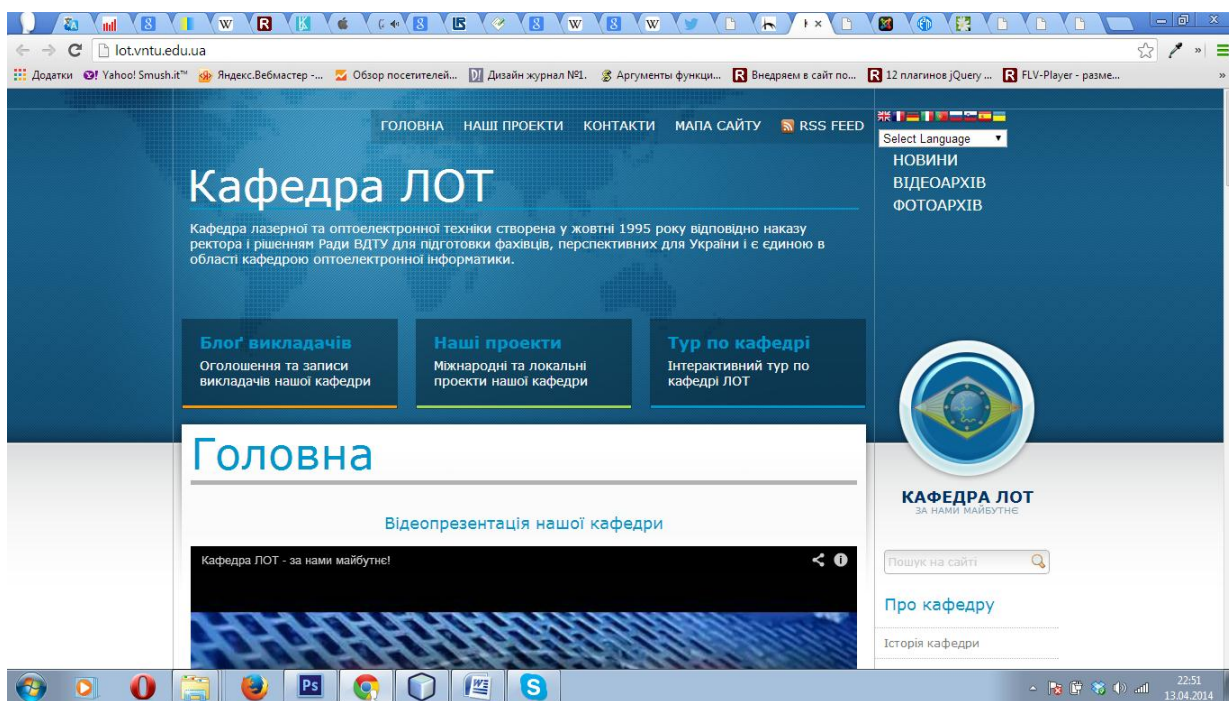


Рисунок 1.1 – Сайт кафедри ЛОТ «<http://lot.vntu.edu.ua/>»

Недоліки:

- переклади здійснюються автоматично за допомогою сервісу Google translate, що забезпечує низьку якість перекладу;
- сайт реалізований на двіжку Wordpress, що не є хорошим тоном і свідчить про низку якість і кваліфікації тих розробників;

- сайт є дуже важким і довго завантажується, що є наслідком непрофесійного використання системи контролю контентом.

2. Сайт «inrtzp.vntu.edu.ua». Даний сайт наведено на рисунку 1.2.

Переваги:

- простота використання;
- швидкість роботи;
- зручна система навігації;

Недоліки:

- багатомовність реалізована некоректно;
- сайт реалізований з використанням устарівших технологій;
- відсутність адмін-панелі.



Рисунок 1.2 – Сайт кафедри електроніки

2. Сайт «rmos.vntu.edu.ua». Даний сайт наведено на рисунку 1.3.

Переваги:

- швидкість роботи;
- зручна система навігації;

Недоліки:

- повністю відсутня багатомовність;

- сайт реалізований з використанням устарівших технологій;
- відсутність адмін-панелі;
- морально застарілий.

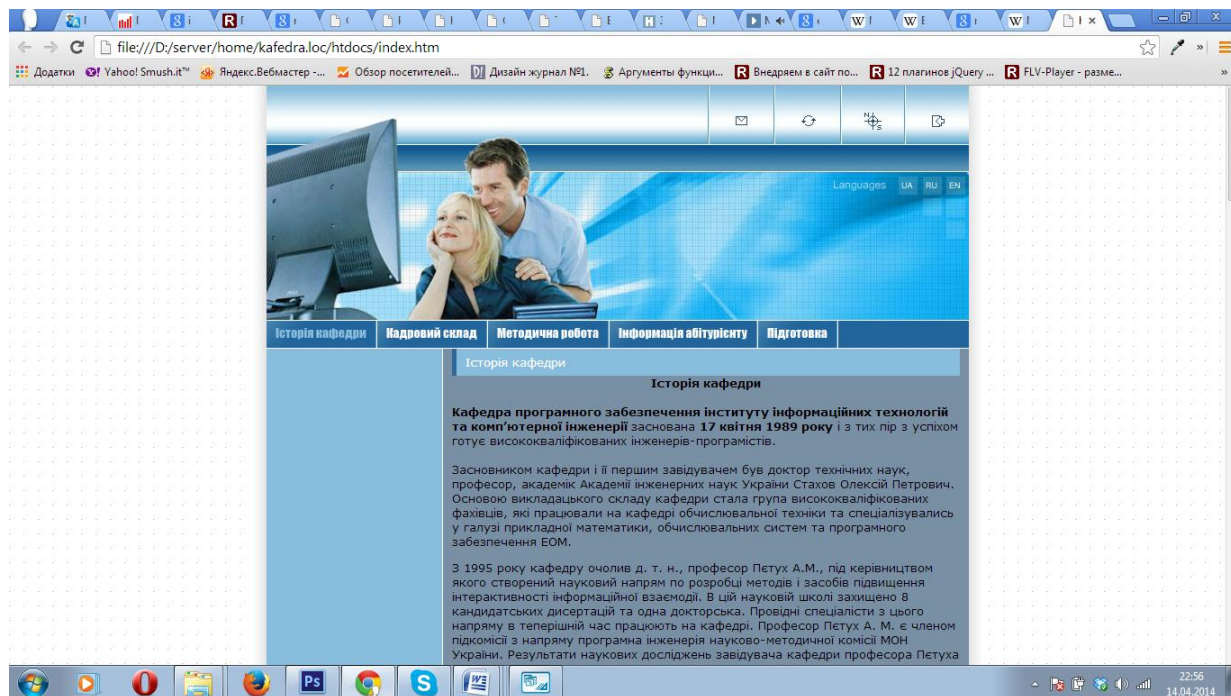


Рисунок 1.3 –Застарілий сайт кафедри ПЗ

На основі детального аналізу сайтів інших кафедр, можна стверджувати, що доцільною є розробка сайту кафедри програмного забезпечення, який повинен максимально можливо поєднати переваги кожного з розглянутих сайтів та виправити їхні недоліки.

На основі проведеного аналізу різних сайтів буде сформована постановка задачі для розробки даного проекту.

1.4 Постановка задач розробки

Основним завданням розробки веб-сайту вважаємо:

1. Провести аналіз та систематизацію характеристик сайтів різноманітних кафедр та інститутів.
2. Визначити систему критеріїв для проведення порівняльного оцінювання основних характеристик і експлуатаційних показників сайтів кафедр

або інститутів.

3. Визначити принципи реалізації динамічних web-сайтів.
4. Розробити структуру Back-end частини сайту кафедри ПЗ.
5. Розробити адміністративну частину сайту.
6. Провести варіантний аналіз програмного забезпечення з метою вибору засобів реалізації сайту.
7. Розробити засоби реєстрації і авторизації користувачів на сайті.
8. Розробити засоби створення, редагування і відображення інформації про викладачів.
9. Протестувати розроблений програмний продукт.

1.5 Висновки

В даному розділі було проведено детальний аналіз предметної області, визначено суть технічної проблеми, актуальність питання, проаналізовано сучасний стан та перспективи розвитку сайтів-візиток різних кафедр навчальних закладів. Також було здійснено огляд існуючих сайтів на дану тематику, визначені їх переваги і недоліки, які будуть враховані при розробці сайту кафедри програмного забезпечення.

У заключному пункті даного розділу було здійснено постановку задач розробки.

2 РОЗРОБКА СТРУКТУРИ ТА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ САЙТУ

Інформаційне забезпечення сайту є одним із найважливіших етапів розробки веб-сторінок. Якщо створення структури та оформлення передбачає привертання уваги користувачів до зовнішнього вигляду сайту, то контент робить ставку безпосередньо на обсяг і якість отриманої відвідувачем інформації.

Наповнення контентом сайту сьогодні є одним з невід'ємних заходів з підтримки працездатності ресурсу. Інформація час від часу обов'язково повинна оновлюватися, вдосконалюватися. Це дає змогу ресурсу довше залишатись більш привабливим для користувачів. Інформаційне наповнення сайту включає в себе написання цікавих матеріалів, поділ їх на розділи, додавання новин корисних повідомлень. Крім текстового контенту, сайт повинен містити графічні, фото та відео матеріали. Чим наочнішими вони будуть, тим привабливішим залишатиметься ресурс для користувачів. Повинна оновлюватися інформація про викладачів кафедри і дані про методичну роботу. Все це буде легко робити в адмін панелі сайту.

2.1 Аналіз принципів реалізації web-сайту

В основному будь-який сайт є сукупністю динамічних або статичних сторінок, які містять тексти, картинки, мультимедійні чи інші об'єкти. Для комфортного перебування відвідувача на сайті і для полегшення пошуку потрібної інформації, сайт повинен мати чітку і продуману структуру.

Веб-сторінка – інформаційний ресурс, доступний у мережі World Wide Web, який можна переглянути у веб-браузері. Зазвичай ця інформація записана у форматі HTML або XHTML і може містити гіпертекст із навігаційними гіперпосиланнями на інші веб-сторінки.

Веб-сайт, сайт – сукупність веб-сторінок, доступних в Інтернеті, які об'єднані як за змістом, так і навігаційно.

Структура сайту – це внутрішній устрій сайту, його «кістяк», розташування сторінок, розділів, підрозділів, додаткових матеріалів. І першочерговим завданням дизайнера є створення строгого порядку з хаотичного скупчення інформації [6].

Створення веб-сайту починається зі створення інформаційної моделі сайту. Будь-яку веб-сторінку можна оцінити за двома параметрами: зміст та зовнішній вигляд. Зовнішній вигляд кожного сайту є унікальним.

Загалом виділяють три типи структур веб-сайтів – лінійну, деревоподібну та довільну.

Додаткові структури сайту:

- стандартна;
- каскад;
- хмарочос;
- павутина;
- утапи створення сайту.

Створення сайту умовно можна розділити на такі етапи:

1. Попередній етап розробки сайту. На цьому етапі розв'язуються питання загального характеру. Обговорюється загальна концепція сайту, формулюються та фіксуються цілі створення сайту.
2. Етап проектування сайту. Визначення структури сайту: меню, посилання, розміщення модулів, побудова списку компонентів, що підключаються, тощо.
3. Етап розробки й тестування сайту.
4. Розміщення сайту.
5. Розвиток ресурсу.

Вдалий дизайн в першу чергу має орієнтуватися на користувача, який без особливих зусиль може знайти і використати будь яку інформацію, що міститься на сайті. При цьому повинні бути абсолютно зрозумілими логічна схема сайту і логічний взаємозв'язок між окремими сторінками.

Хороша навігація сайту означає, що користувачі точно знають, де знаходяться, де розташовані елементи сайту і як використовувати ці елементи.

Правильна структура інформації дозволяє користувачам без побоювання продовжувати мандрування сайтом, і бути впевненими в тому, що вони завжди зможуть без зусиль повернутися до раніше переглянутих сторінок [7].

На вдало організованих сайтах відвідувачі можуть вільно переміщатися у пошуках потрібної інформації, не турбуючись про структуру сайту. Піклування про структуру сайту є завданням розробника, а не відвідувачів.

Логічне проектування передбачає організацію інформації на сайті, побудову його структури та навігацію по розділах [8].

Організація даних у вигляді впорядкованої структури повинна повідомляти відвідувача, яку інформацію він може знайти на сайті, і де її шукати. Інформація повинна бути організована так, щоб відвідувач знав, що його чекає на наступній сторінці вже за назвою посилання, на якому він натискає.

Перед тим, як розділяти офіційну сторінку на окремі сторінки, необхідно розробити план. Сайт проектується таким чином, що потрібно добре продумати його загальну структуру, зміст інформації та посилання.

Нижче наведено чотири варіанти структури сайту. Кожна з цих структур має свої недоліки та переваги при проектуванні офіційного Web-сайту. Вибір конкретної структури визначається особливостями завдань, що вирішуються за допомогою Web-сайту.

Стандартна. Основна Web-сторінка містить посилання на інші документи Web-сайту, а документи містять посилання, відповідно, на основну Web-сторінку. Це найпростіший і найчастотніший спосіб організації Web-сайту. Таке подання інформації з однієї сторони полегшує процес взаємодії з відвідувачами, з іншої сторони подібна організація структури сайту не залишає відвідувачам волі вибору, що може їм не сподобатися. Тому на практиці лінійна структура сайту застосовується з деякими видозмінами. На рисунку 2.1. показана схема стандартного способу розділення Web-сторінки.

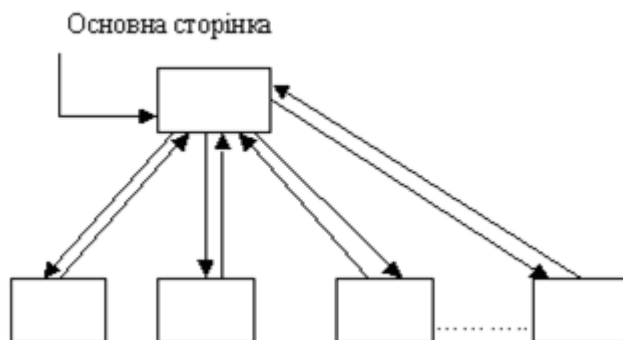


Рисунок 2.1. Стандартний спосіб організації Web-сторінок

Каскад. У цьому випадку посилання у документах задані таким чином, що існує тільки один шлях обходу сторінок Web-сайту. На рисунку 2.2. нижче показаний приклад каскадного сайту.

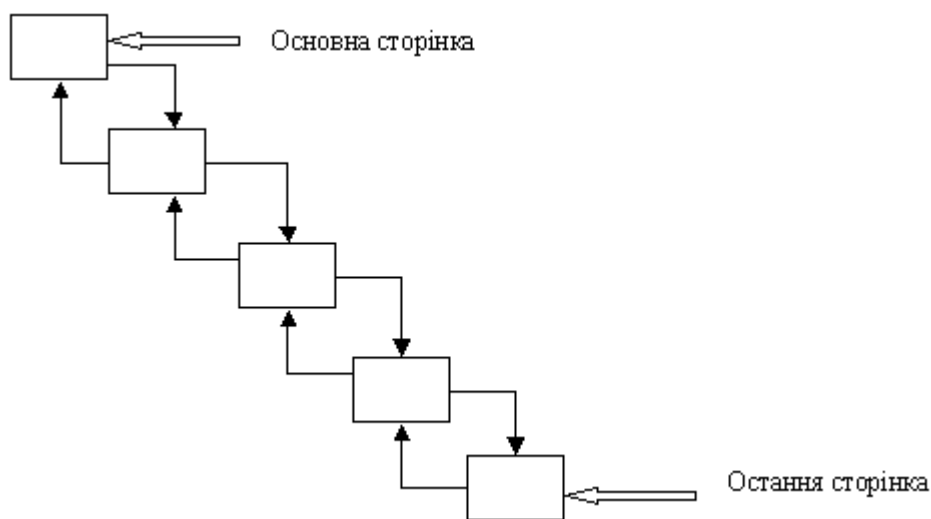


Рисунок 2.2 Каскадний спосіб організації Web-сторінок

При каскадному способі організації сторінок відвідувачі сайту можуть переміщуватись тільки в одному з напрямків, вперед або назад.

Хмарочос. У даній моделі відвідувачі можуть опинитися на деяких сторінках, тільки якщо вони йдуть вірним шляхом. Це нагадує підйом до потрібної кімнати у

великому хмарочосі. Схема, яка наведена на малюнку нижче, дійсно нагадує план хмарочоса.

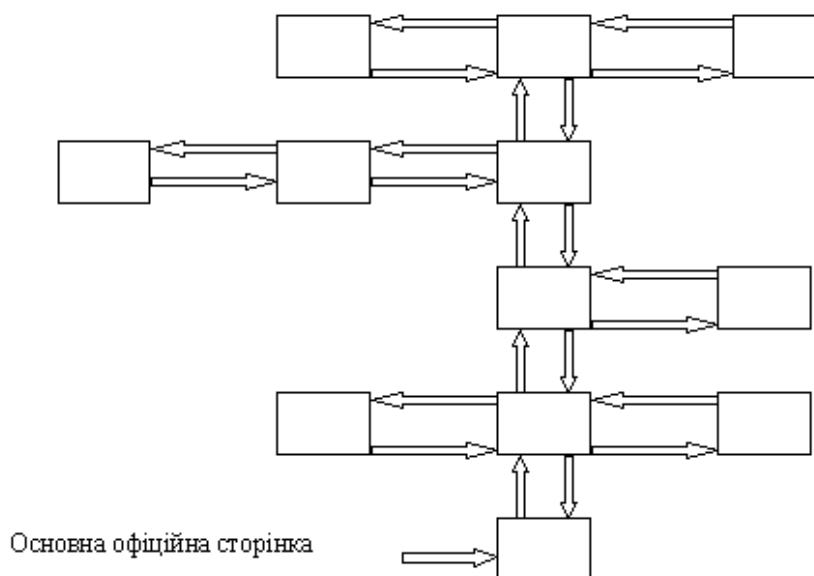


Рисунок 2.3 Структура сайту «хмарочос»

Павутина. У цьому випадку усі сторінки Web-сайту містять посилання на інші сторінки, і користувач може легко перейти з будь-якої сторінки практично на будь-яку іншу. Ця схема може перетворитися у лабіринт, якщо вийде з-під контролю, але вона популярна у тих випадках, коли посиланнями на документи користуються не надто часто. На рисунку 2.4 нижче представлений доволі простий приклад такої організації.

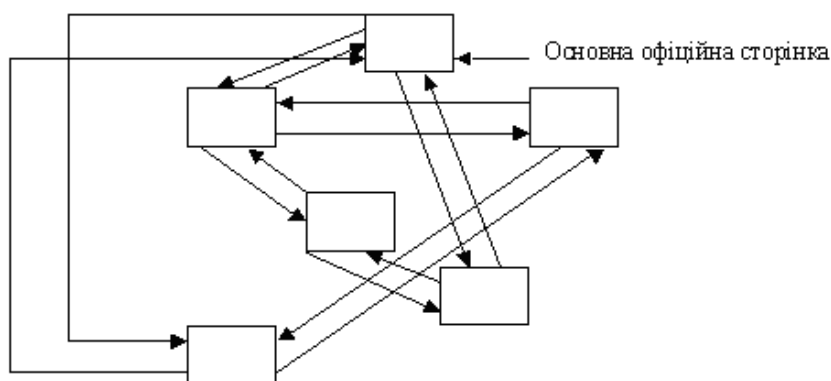


Рисунок 2.4 Структура сайту «павутина»

При організації офіційного Web-сайту рекомендується користуватися гібридним методом, який представляє суміш стандартного методу і методу павутини. Ми можемо опинитись у будь-якому документі безпосередньо з основної офіційної сторінки, при цьому і самі документи також посилаються один на інший. Використання методу “хмарочосу” небажане тому, що воно дезорганізує навігацію користувача у Web-сайті.

Інформацію на Web-сайті необхідно подавати українською мовою, а певні розділи – дублювати іншими мовами, згідно з особливостями регіону (російською, англійською, угорською). Для цього головна сторінка Web-сайту має мати можливість переключення між цими мовами. Меню переключення мов складається із графічних зображень (у форматі GIF), а не з текстових надписів. Це дозволяє здійснювати вибір мови у середовищах, де не підтримується потрібне кодування літер, зокрема українських, наприклад, користувачами Web-сайтів у країнах Європи або Америки.

Обсяги надання інформації в українському та іншомовному варіантах Web-сторінок можуть не співпадати тому, що Web-сайт призначений одночасно для різних верств користувачів в Україні і за кордоном.

Структура є першочерговим кроком у розробці сайту. Вона представляє собою певний каркас, тобто базис, який згодом заповнюється графічним та інформаційним матеріалом.

Найбільш зручною та універсальною є блочна структура веб-сторінок, що передбачає розділення сайту на логічні частини (блоки), до яких окремо застосовується певне графічне оформлення, яке відповідає загальному дизайну всього проекту. Вона є незалежною від різного програмного забезпечення, на якому в подальшому буде відображатися сайт, що надає їй значних переваг над іншими структурами, наприклад, табличною.

Логічну блочну структуру можна реалізувати у вигляді декількох моделей, а саме: лінійної, моделі «грати», «дерево» та «павутина». Використовуються також різні комбінації описаних варіантів.

Лінійна структура сайту дозволяє відвідувачам переходити тільки від попередньої сторінки до наступної, ніяк інакше. Через таку жорсткість та відсутність вибору вона застосовується досить рідко, хоч і є легкою у реалізації. Модель «грати» має і горизонтальні, і вертикальні взаємозв'язки, тобто двонаправлену лінійну структуру, що дає можливість переглядати сторінки за обома критеріями. Структура «дерево» є ієрархічною, і дає можливість керувати глибиною відвідування, а простір вибору обмежується «шириною дерева». Ця структура є найбільш використовуваною та зручною, тому вона буде обрана для розробки заданого сайту. Щодо моделі «павутина», то така організація взаємодії веб-сторінок є заплутаною і важко зрозумілою, але надає користувачам весь обсяг інформації[9].

Головними елементами сайту, на які потрапляє увага відвідувача, є система навігації та меню, назви категорій, гіперпосилання і структура інформації. Вдала навігація означає, що користувач точно знає, де знаходиться, де розташовані елементи сайту і як використовувати ці елементи. Правильна структура інформації дозволяє вільно продовжувати дослідження сайту, залишаючись впевненими в тому, що завжди є можливість повернутися до раніше переглянутих сторінок. Значна зміна елементів навігації при переході від однієї веб-сторінки до іншої відволікає увагу відвідувачів від вмісту сайту, тому її неварто застосовувати. Меню потрібно робити зручним для використання та легко зрозумілим. Надмірне застосування ефектів, задля гарного вигляду, в даному випадку себе не виправдовує.

Розробка заданого сайту базується на описаних методиках, з використанням усіх технологій, принципів та основ побудови структур сайтів. Проаналізувавши всі підходи та вибравши найбільш підходящі методи рішення даної задачі був вибраний найоптимальніший із варіантів побудови структури сайту – каскадний. Згідно з ними, структура сайту з вказаними блоками та зв'язками зображена на Рисунку 2.5.

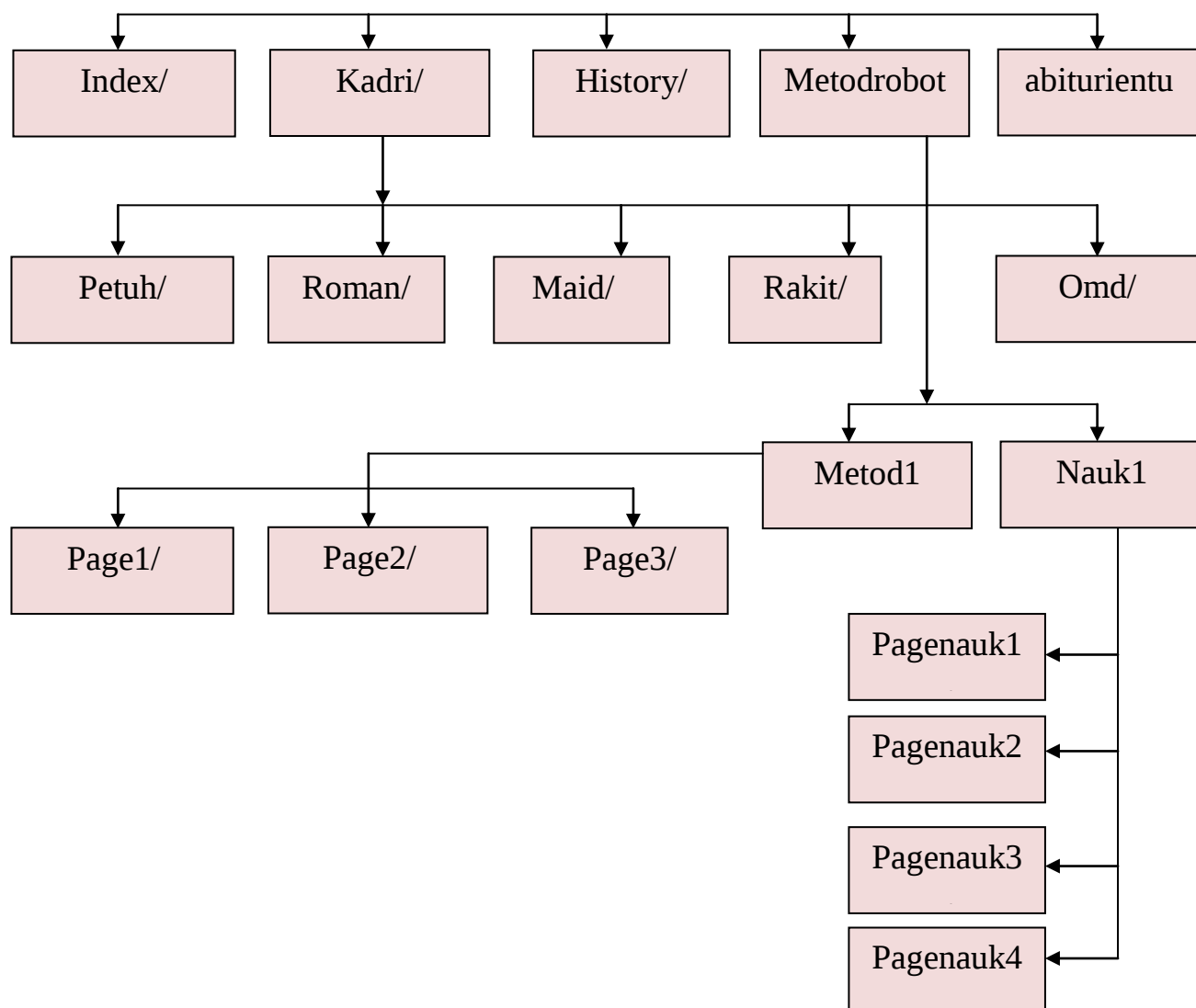


Рисунок 2.5 – Структура веб-сайту

Зі складеної структури сайту видно, що вона відповідає деревовидній моделі, яка характеризується глибиною та шириною побудованого дерева.

Подамо короткий опис назв блоків. Головна сторінка сайту в нашому випадку називається index/. Сторінка історії містить інформацію в короткому варіанті про історію кафедри програмного забезпечення – history/. Кадровий склад кафедри відображається на сторінці kadri/. Веб-сторінка про методичну роботу кафедри показана за адресою /metodrobota/. Вона розбита на дві складові, які описані нижче. Сторінка з інформацією абітурієнту знаходиться за адресою abiturientu/

Так як із сторінки кадрового складу можна перейти до інформації про будь-якого викладача, то вона містить посилання ще на список викладачів кафедри, а саме: Petuh/, Roman/, Maid/, ..,OMD/.

Сторінка методичної роботи розбита на дві складових, які в свою чергу також розбиті на підскладові сторінки. Перші дві під категорії мають назви Metok1 і Nauk1. Для зручності і структурованості роботи, обидві сторінки є нащадкома головної сторінки – Робота, з якої можна перейти по посиланню на ту сторінку, яка найбільше цікавить користувача в даний момент. Однією з переваг даної методики є те, що на батьківській сторінці можна розмістити детальну інформацію.

Оскільки сайт написаний на дуже зручній і сучасній системі управління контентом, то перелік і позиція сторінок головного меню може змінюватися дуже швидко і без надмірних затрат. Для цього потрібно виконати невеликий перелік маніпуляцій в адміністративній панелі сайту, що обмежується кількома кліками на відповідних кнопках керуванням порядком сторінки.

Згідно з описаною вище структурою меню веб-сайту має вигляд, зображений на рисунку 2.6.

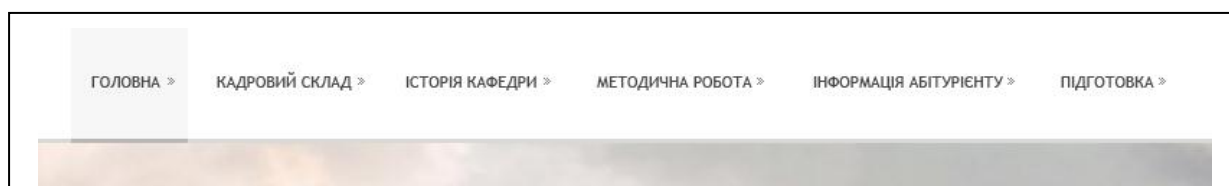


Рисунок 2.6 – Меню веб-сайту

Наведемо загальний вигляд головної сторінки у браузері разом із навігаційним меню на рисунку 2.7. Структура сайту виконана з урахуванням усіх сучасних тенденцій для побудови сайтів, з урахуванням потреб в SEO оптимізації та юзабіліті для кінцевого користувача, а також зручності роботи з кодом для програміста, який займається проектуванням або буде займатися його модифікацією.



Рисунок 2.7 – Загальний вигляд головної сторінки сайту

Інформаційне забезпечення сайту є одним із найважливіших етапів розробки веб-сторінок. Якщо створення структури та оформлення передбачає привертання уваги користувачів до зовнішнього вигляду сайту, то контент робить ставку безпосередньо на обсяг і якість отриманої відвідувачем інформації.

Наповнення контентом сайту сьогодні є одним з невід’ємних заходів з підтримки працездатності ресурсу. Інформація час від часу обов’язково повинна оновлюватися, вдосконалюватися. Це дає змогу ресурсу довше залишатись більш привабливим для користувачів. Інформаційне наповнення сайту включає в себе написання цікавих матеріалів, поділ їх на розділи, додавання новин корисних повідомлень. Крім текстового контенту, сайт повинен містити графічні, фото та відео матеріали. Чим наочнішими вони будуть, тим привабливішим

залишатиметься ресурс для користувачів. Повинна оновлюватися інформація про викладачів кафедри і дані про методичну роботу. Все це буде легко робити в адмін панелі сайту.

2.2 Розробка інформаційного забезпечення головної сторінки

Інформація, що міститься на головній сторінці сайту, є певною візитною карткою продукту, який створюється. Користувач, аналізуючи її складає загальне враження про відвідуваний ресурс, а отже і про всю кафедру програмного забезпечення інституту, робить висновки про відповідність текстового та графічного матеріалу власним вимогам, а також визначається з вибором: продовжувати пошук необхідної інформації на поточному сайті чи шукати більш вдалі ресурси. Це значить, що кафедра буде втрачати потенційних своїх учнів. Пам'ятаючи про це, потрібно відповідально ставитися до контенту головної сторінки, який має бути сучасним і не залежати від думок викладачів, які нічого не розуміють в даній темі. Текст повинен бути чітким, лаконічним, зрозумілим, змістовним, невеликого обсягу, в кращому випадку має супроводжуватись ілюстраціями, малюнками, відео-матеріалом. Варто зберігати логічну послідовність викладу інформації, дотримуватись її структури. Актуальним є розміщення останніх новин кафедри, цікавих фактів і подій, які відбуваються або проводяться за участі кафедри, обговорюваних проблем, новинок у обраній темі, змін у методичній роботі. Вони більш вірогідно можуть привернути увагу відвідувача, ніж звичайна інформація, типу тупо накиданого тексту, який ніхто не читає, так як це зроблено на старому ресурсі. Слід пам'ятати, що контент головної сторінки принципово залежить від тематики створюваного сайту, яка задає орієнтири для розробника.

Кафедра програмного забезпечення інституту інформаційних технологій та комп'ютерної інженерії заснована 17 квітня 1989 року і з тих пір з успіхом готує висококваліфікованих інженерів-програмістів.

Засновником кафедри і її першим завідувачем був доктор технічних наук, професор, академік Академії інженерних наук України Стахов Олексій Петрович. Основою викладацького складу кафедри стала група висококваліфікованих фахівців, які працювали на кафедрі обчислювальної техніки та спеціалізувались у галузі прикладної математики, обчислювальних систем та програмного забезпечення ЕОМ.

З 1995 року кафедру очолив д. т. н., професор Петух А.М., під керівництвом якого створений науковий напрям по розробці методів і засобів підвищення інтерактивності інформаційної взаємодії. В цій науковій школі захищено 8 кандидатських дисертацій та одна докторська. Провідні спеціалісти з цього напрямку в теперішній час працюють на кафедрі. Професор Петух А. М. є членом підкомісії з напрямку програмна інженерія науково-методичної комісії МОН України. Результати наукових досліджень завідувача кафедри професора Петуха А.М. щодо альтернативного представлення величин демонструвались на міжнародних виставках винаходів [10].

2.3 Розробка інформаційного забезпечення сторінки про методичну роботу

Інформація для сторінки про методичну роботу є безпосередньо тією частиною веб-сайту, що повинна повністю розкривати відповідну тему, містити корисні, актуальні, ретельно відібрані, логічно зв'язані дані про роботу кафедри, які б цілком задовольняли користувача. З огляду на це, потрібно з особливою увагою віднестись до розробки текстового наповнення, а краще його просто взяти з старого сайту кафедри. В нашому випадку на сторінці про методичну роботу кафедри найбільш доцільно розмістити інформацію про те, чим займається кафедра, що вони випускають і які останні праці були розроблені. Далі наведено частину текстового контенту для сторінки методичної роботи.

Кафедра програмного забезпечення постійно вдосконалює навчальний процес, видає навчальні посібники та методичну літературу, впроваджує нові програмні продукти, розробляє оригінальні пакети прикладних програм. Всі

дисципліни, що читаються, в достатній мірі забезпечені методичною літературою з практичних занять, лабораторних практикумів, самостійної роботи студентів, курсового і дипломного проектування.

В навчальному процесі широко використовуються новітні технології навчання, зокрема елементи дистанційної форми навчання. Викладачами кафедри проводиться робота по створенню інформаційно-методичного забезпечення (включає навчальні і робочі програми, тексти лекцій, методичні вказівки до лабораторних робіт, курсового проектування і самостійної роботи, рейтингову систему оцінювання тощо) по всіх дисциплінах робочих навчальних планів, які розміщуються в інформаційному ресурсі дистанційної освіти «E-learning server».

Професор кафедри, д.т.н. Романюк О. Н. у 2012 році відвідав Китайську народну республіку, де прочитав цикл лекцій для студентів Цзинанського педагогічного університету та університету радіоелектроніки та телебачення, ознайомився з їх методиками навчання. [11].

2.4 Розробка інформаційного забезпечення сторінки викладацького складу

Текстове наповнення сторінки викладацького складу повинно бути мінімальним, і максимально точним і лаконічним, з точки зору огляду на те, що тут головним акцентом слід залишити конкретну інформацію про викладача. Вони несуть найбільшу частину інформації в цьому розділі. Текст виконує основну функцію, тобто детально описує персону викладача з усіма його досягненнями, займаною посадою, всіма вченими званнями.

Інформаційний контент сторінки викладацького складу, в нашому випадку, має вигляд переліку усіх викладачів кафедри, що описані на сайті. А саме: ПСТУХ Анатолій Михайлович, РОМАНЮК Олександр Никифорович, МАЙДАНЮК Володимир Павлович, РАКИТЯНСЬКА Ганна Борисівна, КРУПОДЬОРОВА Людмила Михайлівна, ВЛАСЮК Василь Харитонович, ДЕНИСЮК Алла Василівна, ВОЙТКО Вікторія Володимирівна, МОСКВИЧОВА Олена Віталіївна, КУЗЬМІНА Наталя Федорівна, КУЗЬМІНА Наталя Федорівна,

РЕЙДА Олександр Миколайович, ЧЕРНОВОЛИК Галина Олександрівна, КУЗЬМІН Євгеній Володимирович, ОБІДНИК Микола Дем'янович, ОБІДНИК Микола Дем'янович, РОМАНЮК Оксана Володимирівна, КОЦЮБІВСЬКА Катерина Іванівна

2.5 Розробка інформаційного забезпечення сторінки абітурієнту

Сторінка абітурієнту є важливим елементом кожного веб-сайту, який представляє кафедру інституту. Вона передбачає об'ємний та детальний текстовий контент, що містить усю необхідну і доцільну інформацію, потрібну абітурієнту в момент вступу на кафедру для подальшого навчання і отримання крутих професійних навиків, які в майбутньому забезпечать його виживання в цьому жорстокому світі. Наприклад, логічним є розміщення контактних даних (телефону, пошти та адреси) прийомної комісії, або детального переліку усіх потрібних документів, які постійно вимагають при вступі. Також може бути описані умови вступу на дану кафедру та детально описані результати, який буде досягнуто студентом по закінченню навчання. Така сторінка, як правило, безпосередньо не стосується теми веб-сайту. Вона стосується кафедри і вступу на навчання

Далі наведено приклад текстового контенту, розміщеного на сторінці про нас. Прийом здійснюється на денну та заочну форму навчання як за держзамовленням, так і за контрактом. Ліцензований обсяг прийому зі спеціальності складає 60 осіб

Спеціалістів з програмного забезпечення випускає кафедра програмного забезпечення (ПЗ). Наші випускники працюють на великих, малих та спільних підприємствах, обчислювальних центрах, науково-дослідних інститутах та банках України. СНД та інших країн.

За результатами 4-х років навчання випускники одержують ступінь бакалавра з напрямку підготовки «Програмна інженерія» (кваліфікація – 3121 фахівець з розробки та тестування програмного забезпечення). Ще через рік

навчання можна одержати диплом спеціаліста зі спеціальності 7.05010301 «Програмне забезпечення систем» (кваліфікація 2132.2 «Інженер-програміст»). Найбільш здібні до наукової роботи студенти продовжують навчання і отримують вчену ступінь магістра програмного забезпечення систем. Наукові напрями підготовки магістрів: засоби машинної графіки; системи цифрової обробки сигналів та зображень; розробка нових форм представлення інформації.

2.6 Розробка ER-моделі бази даних

Модель сутність-зв'язок (ER-модель) (англ. entity-relationship model, ERM) - модель даних, що дозволяє описувати концептуальні схеми предметної області.

ER-модель використовується при високорівневого (концептуального) проектування баз даних. З її допомогою можна виділити ключові сутності і позначити зв'язки, які можуть встановлюватися між цими сутностями.

ER-модель зручна при проектуванні інформаційних систем, баз даних, архітектур комп'ютерних застосунків та інших систем (моделей). За допомогою такої моделі виділяють найсуттєвіші елементи (вузли, блоки) моделі і встановлюють зв'язки між ними.

Існує ряд моделей для представлення знань. Одним з найзручніших інструментів уніфікованого представлення даних, незалежного від реалізуючого його програмного забезпечення, є модель "сутність-зв'язок" (*entity - relationship model, ER - model*).

Модель розробляється у візуальному середовищі проектування бази даних - MySQL Workbench. MySQL Workbench — інструмент для візуального проектування баз даних, що інтегрує проектування, моделювання, створення й експлуатацію БД в єдине безшовне оточення для системи баз даних MySQL. Є наступником DBDesigner 4 з FabForce.

ER-модель бази даних зображена на рисунку 2.8.

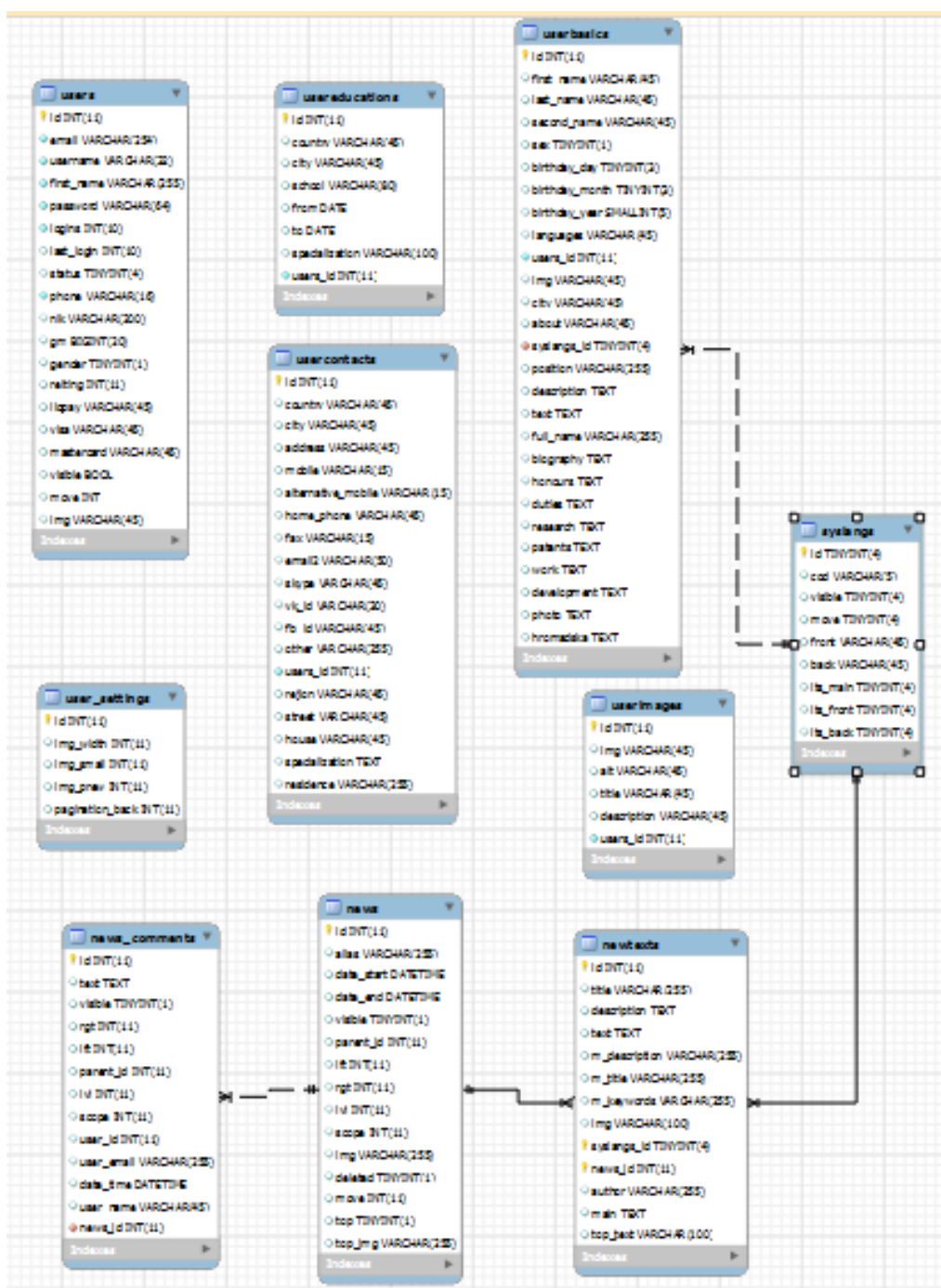


Рисунок 2.8 - ER-модель бази даних сайту

Після створення моделі бази даних, її необхідно синхронізувати з створеною заготовкою бази даних на сервері. Дана заготовка створюється в середовищі PhpMyAdmin, а експортується із середовища MySQL Workbench за допомогою команди База даних / Синхронізувати, після чого вибирається потрібна база на сервері.

2.7 Висновки

В даному розділі було проаналізовано принципи реалізації сайту. Було визначено наступні структури: стандартна (лінійна), каскадна, хмарочосна, павутинна (комбінована). Для реалізації сайту було обрано павутинну структуру. Також було визначено основні принципи, що будуть застосовуватись при проектуванні інформаційного наповнення сайту і засобів навігації.

Було визначено типи інформаційного забезпечення та коротко охарактеризовано їх. Доцільним вважається розміщення на сайті загальної інформації про кафедру, про історію та роботу цієї кафедри у різних сферах та інформацію для абітурієнтів, що повинна сприяти полегшенню процесу збору інформації для вступу в університет, інформацію про кадровий склад з детальною інформацією про викладачів, що працюють на кафедрі. Детально описаний і реалізований модуль новин, що забезпечує додавання, редагування та видалення новин, також він забезпечує виведення останніх новин на головній сторінці сайту.

В розділі було розроблено функціональну структуру клієнтської та адміністративної частин сайту відповідно до вимог та принципів розробки web-сайту .

Також було розроблено інтерфейс та логіку роботи сайту, які повністю реалізують функції, що на них покладені.

Для більш змістовного наповнення було використано статичні графічні зображення, які надають контенту сайту різноманітності.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ САЙТУ

Наступним важливим, можна сказати – основним, етапом розробки веб-сайту буде розробка програмного забезпечення сайту. Для початку, оберемо зручне середовище розробки. Також проведемо варіантний аналіз програмних засобів реалізації та систем керування базами даних. Обравши найбільш підходящий варіант для реалізації поставлених завдань, реалізуємо основні модулі веб-сайту.

3.1 Вибір середовища розробки веб-сайту

На сьогодні існує величезна кількість систем створення веб-додатків, які підрозділяють на три різні категорії:

- текстові редактори HTML-коду (т.з. «text-based HTML editors»: Alleycode HTML Editor, Aptana, Arachnophilia, BBEdit і т.п.);
- редактори WYSIWYG («What You See Is What You Get» або «що ви бачите, то ви і отримаєте» – досить потужні і зручні засоби створення і обробки веб-додатків: Adobe Contribute "Dreamweaver Lite", Adobe Dreamweaver, Adobe GoLive, Amaya, Blockstar, Bluevoda, HotDog, iWeb, Media Lab, SiteGrinder Sandvox, SeaMonkey Composer, Softpress Freeway, RapidWeaver, WorldWideWeb і т.п.) [12].
- текстові редактори (Emacs, gedit, Notepad, TextEdit, UltraEdit, vi і т.п.);

Для реалізації веб-сайту було обрано інтегроване середовище розробки NetBeans IDE, так як воно є найпопулярнішою системою розробки веб-додатків. NetBeans IDE — вільне інтегроване середовище розробки (IDE) для мов програмування Java, JavaFX, C/C++, PHP, JavaScript, HTML5, Python, Groovy. Середовище може бути встановлене і для підтримки окремих мов, і у повній конфігурації. Середовище розробки NetBeans за умовчанням підтримує розробку для платформ J2SE і J2EE.

Поширюється у сирцевих текстах під ліцензіями GPLv2 і CDDL. Проект NetBeans IDE підтримувався і спонсорувався фірмою Sun Microsystems і після придбання Sun — Oracle, проте розробка NetBeans ведеться незалежно співтовариством розробників (NetBeans Community) і компанією NetBeans.Org.

NetBeans IDE доступна для платформ Microsoft Windows, GNU/Linux, FreeBSD, і Solaris (як SPARC, так x86). Для інших платформ доступна можливість зібрати NetBeans самостійно із сирцевих текстів.

За якістю і можливостям останні версії NetBeans IDE змагається з найкращим інтегрованими середовищами розробки для мови Java, підтримуючи рефакторинг, профілювання, виділення синтаксичних конструкцій кольором, автодоповнення мовних конструкцій на льоту, шаблони коду та інше.

3.2 Варіантний аналіз і обґрунтування вибору програмних засобів для реалізації веб-сайта

Ще одним важливим підготовчим етапом є вибір мови програмування для розробки веб-сайту. Для створення веб-додатків використовуються різноманітні технології і мови програмування, оскільки кожна з мов реалізує свою можливість, що визвані способом реалізації програм на даній мові. Кожна з них має свої преевоги і недоліки, тому вибір мови повинен залежати від поставленої задачі і в результаті прорахування навантаження. Найпопулярнішими мовами для задач подібного типу є:

- ASP;
- ASP .NET;
- Java script;
- Perl;
- PHP.

3.2.1 Аналіз можливостей ASP та ASP .NET

ASP .NET – технологія створення веб-додатків і веб-сервісів від компанії Майкрософт. Вона є складовою частиною платформи Microsoft NET і розвитком старішої технології Microsoft ASP [13].

ASP .NET зовні багато в чому зберігає схожість із старішою технологією ASP, що дозволяє розробникам відносно легко перейти на ASP.NET. У той же час внутрішній устрій ASP .NET істотно відрізняється від ASP, оскільки вона заснована на платформі .NET і, отже, використовує всі нові можливості, що надаються цією платформою, яка своєю структурою зручно забезпечує всі можливості роботи [14].

Порівняно з ASP, ASP .NET має ряд переваг:

- компільований код виконується швидше, більшість помилок відловлюється ще на стадії розробки;
- значно поліпшена обробка помилок часу виконання, з використанням блоків try .. catch;
- користувацькі елементи управління (controls) дозволяють виділяти часто використовувані шаблони, такі як меню сайту;
- використання метафор, вже застосовуються в Windows-додатках, наприклад, таких як елементи керування та події;
- можливість поділу візуальної частини та бізнес-логіки з різних файлів («code behind»);
- розширювана модель обробки запитів, розширена подієва модель, розширювана модель серверних елементів керування;
- наявність master-сторінок для завдання шаблонів оформлення сторінок.
- вбудована підтримка AJAX [15].

3.2.2 Аналіз можливостей JavaScript

JavaScript – назва реалізації стандарту мови програмування ECMAScript компанії Netscape, базується на принципах прототипного програмування. Найпоширеніше і найвідоміше застосування мови – написання сценаріїв для веб-сторінок, але, також, використовується для впровадження сценаріїв керування об'єктами вбудованими в інші програми.

Незважаючи на схожість назв, JavaScript та мова програмування Java мають дуже мало спільного. Права на назву JavaScript належать компанії Sun Microsystems [16].

JavaScript має низку властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів, підтримка об'єктів в ній відрізняється від традиційних мов ООП. Крім того, JavaScript має ряд властивостей, притаманних функціональним мовам – функції як об'єкти першого рівня, об'єкти як списки, каррінг (currying), анонімні функції, замикання (closures), що додає мові додаткову гнучкість [14].

JavaScript має C-подібний синтаксис, але в порівнянні з мовою C має такі корінні відмінності:

- об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів;
- функції як об'єкти першого класу;
- обробка винятків;
- автоматичне приведення типів;
- автоматичне прибирання “сміття”;
- анонімні функції [7].

JavaScript містить декілька вбудованих об'єктів: Global, Object, Error, Function, Array, String, Boolean, Number, Math, Date, RegExp. Крім того, JavaScript містить набір вбудованих операцій, які, строго кажучи, не обов'язково є

функціями або методами, а також набір вбудованих операторів, що управляють логікою виконання програм. Синтаксис JavaScript в основному відповідає синтаксису мови Java (тобто в кінцевому рахунку успадкований від C), але спрощений порівняно з ним, щоб зробити мову сценаріїв легкою для вивчення. Так, приміром, декларація змінної не містить її типу, властивості також не мають типів, а декларація функції може стояти в тексті програми після неї [13].

3.2.3 Аналіз можливостей Perl

Perl – високорівнева, інтерпретована, динамічна мова програмування загального призначення. Perl було розроблено у 1987 році Ларі Воллом, як скриптову мову для Unix, метою якої було полегшити процес обробки текстів файлів журналів. Маючи схожий з C синтаксис, реалізуючі регулярні вирази, будучи безтиповим, як Бейсік, з великим числом внутрішніх системних функцій, можливістю виклику зовнішніх програм і підключенням бібліотек, Perl забезпечував зручність при обробці тексту та маніпуляції з файлами. Це і було найбільш необхідно в Web (Perl так само зручний і для написання системних утиліт: існують великі системи, написані на цій мові). Але, не дивлячись на всі ці переваги, все ж таки Perl є мовою загального призначення, і має ряд недоліків:

- необхідність налаштування Web-сервера для запуску скриптів (зазвичай створюється папка CGI-BIN, і в Web-сервері ставиться дозвіл на виконання скриптів, що знаходяться в цій папці і нижче;
- складнощі з налагодженням CGI-скриптів. Скрипт можливо відлагодити, лише запускаючи його напряму (зазвичай запустити на сервері скрипт і подивитися висновок інформації про помилки та попередження можна через Telnet / SSH, а такої можливості досить часто веб-програміст просто позбавлений);
- незручності, що виникають при необхідності відправлення E-Mail-повідомлення з CGI скрипта на Perl. В даному випадку доводиться викликати програму Sendmail, передаючи їй в STDIN необхідні

заголовки листа і, відповідно, тіло листа, що не дуже зручно (з огляду на те, що необхідно знати дорогу до Sendmail на сервері, та доведеться зайвий раз звертатися в технічну підтримку).

Не дивлячись на перераховані недоліки, Perl - добре підходить для Web-програмування [15].

3.2.4 Аналіз можливостей PHP

PHP (Hypertext Preprocessor) – скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок. PHP підтримується переважною більшістю хостинг-провайдерів.

Головним чинником мови PHP є практичність. PHP повинен надати програмісту інструмент для швидкого і ефективного вирішення поставлених завдань. Практичний характер PHP обумовлений п'ятьма важливими характеристиками:

- традиційність;
- простота;
- ефективність;
- безпека;
- гнучкість.

Мова PHP здаватиметься знайомою програмістам, що працюють в різних областях. PHP – мова, що поєднує достоїнства Perl, C і спеціально націлена на роботу в Інтернеті, мова з універсальним і зрозумілим синтаксисом. Вона може бути вбудована безпосередньо в html-код сторінок, які, у свою чергу будуть коректно оброблятися PHP-інтерпретатором. Велика різноманітність функцій PHP позбавляє від написання багаторядкових, призначених для користувача функцій на C або Pascal.

Дуже важлива перевага RHP полягає в його «движку». «Движок» RHP не є ні компілятором, ні інтерпретатором. Він є транслуючим інтерпретатором. Такий «движок» RHP дозволяє обробляти сценарії з достатньо високою швидкістю [12].

RHP надає в розпорядження розробників та адміністраторів гнучкі та ефективні засоби безпеки, які умовно поділяються на дві категорії: засоби системного рівня та засоби рівня програми.

У RHP реалізовані механізми безпеки, що знаходяться під управлінням адміністраторів; при правильному налаштуванні RHP це забезпечує максимальну свободу дій і безпеку. RHP може працювати в так званому безпечному режимі (safe mode), який обмежує можливості застосування RHP користувачами по ряду важливих показників. Наприклад, можна обмежити максимальний час виконання та використання пам'яті (неконтрольована витрата пам'яті негативно впливає на швидкодію сервера).

У стандартний набір функцій RHP входить ряд надійних механізмів шифрування. RHP також сумісний з багатьма додатками незалежних фірм, що дозволяє легко інтегрувати його з захищеними технологіями електронної комерції (e-commerce). Інша перевага полягає в тому, що вихідний текст сценаріїв RHP не можна переглянути у браузері, оскільки сценарій компілюється до його відправлення за запитом користувача [30].

Оскільки RHP є вбудовуванню (embedded) мовою, вона відрізняється винятковою гнучкістю по відношенню до потреб розробника. Хоча RHP зазвичай рекомендується використовувати у поєднанні з HTML, він з таким же успіхом інтегрується і в JavaScript, WML, XML та інші мови. Крім того, добре структуровані програми RHP легко розширюються в міру необхідності (втім, це відноситься до всіх основних мов програмування). Поріг входження цієї мови є досить низьким, що дозволяє великій кількості програмістів використовувати її, затративши дуже мало часу на процес навчання.

3.2.5 Обґрунтування вибору засобу реалізації веб-сайту

Отже, проаналізувавши web-технології програмування було вирішено використовувати мову програмування PHP для написання сценаріїв.

Хоча PHP є досить молодого мовою, вона знайшла таку популярність серед web-програмістів, і на даний момент є мало чи не найпопулярнішою мовою для створення web-додатків (скриптів).

Код PHP дуже схожий на той, який зустрічається в типових програмах на C або Pascal, що помітно знижує початкові зусилля при вивченні PHP.

Головні особливості PHP – практичність, легкість у використанні, ефективність, продуктивність та гнучкість [11].

Прийняття стратегії Open Source і безкоштовне розповсюдження початкових текстів PHP надало неоціненну послугу користувачам. До того, співтовариство користувачів PHP є свого роду «колективною службою підтримки», і в популярних електронних конференціях можна знайти відповіді на найскладніші питання.

3.3 Варіантний аналіз систем керування базами даних

Робота з базами даних є одною з найважливіших складових програмування сайту динамічного типу. Бази даних для сайтів використаємо з метою зберігання різної інформації. Спрощено бази даних представляють собою певний набір взаємозалежних таблиць. Розміри таблиць в базах даних різні, їх кількість довільна. Саме в базах даних буде зберігатися на сервері необхідна для роботи сайту інформація, наприклад, інформація про клієнтів, каталог товарів, статистичні дані тощо.

До найпопулярніших баз даних, що використовують при програмуванні сайтів відносять SQL, MySQL, Oracle Database, PostgreSQL та ін. Для того, щоб обрати СУБД, найбільш підходящу для розробки сайту кафедри, проаналізуємо переваги та недоліки кожної з них [14].

3.3.1 Аналіз можливостей SQL

Для побудови запитів до баз даних широко застосовується SQL (Structured Query Language) - “мова структурованих запитів”. За допомогою SQL може здійснюватися додавання, видалення, редагування записів в таблицях баз даних, вибірка даних у відповідності з різними умовами, сортування даних і багато іншого.

Основними перевагами SQL є:

- Незалежність від конкретної СУБД. Незважаючи на наявність діалектів і відмінностей в синтаксисі, в більшості своїй тексти SQL-запитів, що містять, DDL і DML, можуть бути досить легко перенесені з однієї СУБД в іншу. Існують системи, розробники яких спочатку закладалися на застосування щонайменше кількох СУБД (наприклад: система електронного документообігу Documentum може працювати як з Oracle, так і з Microsoft SQL Server і IBM DB2). Природно, що при застосуванні деяких специфічних для реалізації можливостей такої переносимості добитися вже дуже важко.

- Наявність стандартів. Наявність стандартів і набору тестів для виявлення сумісності і відповідності конкретній реалізації SQL загальноприйнятому стандарту, що тільки сприяє «стабілізації» мови. Правда, варто звернути увагу, що сам по собі стандарт місцями занадто формалізований і роздутий в розмірах, наприклад, Core-частину стандарту SQL:2003 включає понад 1300 сторінок тексту [15].

- Декларативність. За допомогою SQL програміст описує тільки те, які дані потрібно витягнути або модифікувати. Те, яким чином це зробити, вирішує СУБД безпосередньо при обробці SQL-запиту. Проте не варто думати, що це повністю універсальний принцип – програміст описує набір даних для вибірки або модифікації, проте йому при цьому корисно уявляти, як СУБД розбиратиме текст його запиту. Особливо критичні такі моменти стають при роботі з великими базами даних і зі складними запитамі – чим складніше сконструйований запит,

тим більше він допускає варіантів написання, різних за швидкістю виконання, але тих самих за набором даних.

SQL має і недоліки:

- невідповідність реляційної моделі даних;
- складність;
- відступи від стандартів;
- складність роботи з ієрархічними структурами.

3.3.2 Аналіз можливостей MySQL

СУБД MySQL - вільна система управління базами даних , одна з найбільш часто вживаних в програмуванні сайтів. СУБД MySQL підтримує велику кількість існуючих типів таблиць (InnoDB, MyISAM і т. д.), а завдяки відкритій архітектурі і GPL-ліцензуванню, в СУБД MySQL постійно з'являються нові типи таблиць. Управління базами даних за допомогою MySQL дуже зручне, що зробило дану систему затребуваною і популярною.

Переваги сервера MySQL наступні:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн.;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки [11].

Проте, основною перевагою СУБД MySQL є те, що використання MySQL є безкоштовним.

Проте сервера MySQL має і недоліки:

- нереалізована підтримка транзакцій. Натомість пропонується використовувати LOCK/UNLOCK TABLE;
- відсутня підтримка зовнішніх (foreign) ключів;
- відсутня підтримка тригерів і збережених процедур;

- відсутня підтримка представлень (VIEW).

Зазначені недоліки не є критичними при розробці малих і середніх інформаційних систем для робочих груп [12].

Таким чином, СУБД MySQL є однією з найпоширеніших реляційних систем управління базами даних. Вона широко використовується для створення web-сторінок.

3.3.3 Аналіз можливостей Oracle Database

Об'єктно-реляційна система управління базами даних компанії Oracle - Oracle Database - працює на Windows, Unix, Linux, MacOS. Oracle Database, на відміну від MySQL, наприклад, має більш широку сферу застосування. СУБД Oracle має високу продуктивність, широкий функціонал, унікальні технології (RAC, RAS і т. д.). У програмуванні сайтів для невеликих і середніх компаній застосовується досить рідко через свою високу вартість. До того ж, досить складно знайти хостинг з підтримкою даної СУБД. Переваги:

- продуктивність;
- функціональна потужність;
- надійність;
- масштабованість;
- захищеність;
- відкритість;
- можливість підтримання системи з високими вимогами щодо захисту даних, безперервності роботи систем, що працюють в інтернеті і інтранеті.

У програмуванні сайтів для невеликих і середніх компаній застосовується досить рідко через свою високу вартість, що в результаті призводить до нерентабельності проектів таких розмірів. До того ж, досить складно знайти хостинг з підтримкою даної СУБД [9].

3.3.4 Аналіз можливостей PostgreSQL

PostgreSQL – широко поширена система керування базами даних з відкритим програмним кодом. Прототип був розроблений в Каліфорнійському університеті Берклі 1987р. під назвою POSTGRES, після чого активно розвивався і доповнювався. Порівняно до інших проектів з відкритим кодом, такими як Apache, FreeBSD або MySQL, PostgreSQL не контролюється якоюсь однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використувати цю СКБД та впроваджувати у неї найновіші досягнення.

Вільна система управління базами даних PostgreSQL існує в редакціях для Linux, Solaris/OpenSolaris, Win32, Win x86-64, Mac OS X, FreeBSD, QNX 4.25, QNX 6. Базується на мові SQL. Серед переваг PostgreSQL виділяють підтримку БД практично необмеженого розміру, наявність надійних механізмів реплікації, легку розширюваність, підтримку великого набору вбудованих типів даних та багато іншого.

Серед переваг PostgreSQL виділяють підтримку БД практично необмеженого розміру, наявність надійних механізмів реплікації, легку розширюваність, підтримку великого набору вбудованих типів даних та багато іншого [10].

Основними функціональними можливостями PostgreSQL:

- підтримка всіх задекларованих SQL запитів, оптимізація запитів;
 - можливість одночасної роботи багатьох користувачів;
 - каскадне маніпулювання даними;
 - транзакції;
 - внутрішні процедурні мови, включаючи природну мову, названу PL/pgSQL.
- Є можливість використовувати Perl, Python або TCL — як вбудовані процедурні мови;
- гнучкий API. Ці інтерфейси містять Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++ та Pike [9].

3.3.5 Обґрунтування вибору систем керування базами даних

Отже, для того щоб обрати СКБД, яка б як найкраще підходила саме для виконання задач при розробці сайту кафедри програмного забезпечення, визначимо, які вимоги має задовольняти обрана СКБД [14].

Система керування базою даних має відповідати таким вимогам:

- надійність;
- швидкість;
- простота;
- зручність використання;
- вартість.

Проаналізувавши переваги та недоліки СКБД можна зробити наступні висновки:

- мова структурованих запитів SQL не підходить для реалізації поставлених завдань, так як вона не відповідає реляційній моделі даних та є складною у використанні [8].
- Oracle Database має багато переваг, які є ключовими у виборі СКБД для розробки сайту кафедри програмного забезпечення, проте вона рідко використовується для невеликих та середніх сайтів компаній, так як має високу вартість.

Отже, для забезпечення функціональності веб-сайту буде доцільно використовувати MySQL.

3.4 Розробка бази даних засобами MySQL

Даний сайт будується на основі фреймворка Кохана, який містить готові модулі для роботи з базами даних. Архітектура даного модуля відповідає шаблону HMVC, що забезпечує зручну і наглядну структуру файлів і каталогів нашого сайт. Це дає зручність у формуванні зв'язків між таблицями бази даних і

файлів моделей. Розглянемо основні кроки при роботі з базами даних на цьому фреймворці.

3.4.1 Налаштування модуля database

Сайт , на якому присутня рідко змінювана інформація (наприклад, сайт-візитка) зазвичай організовують за допомогою декількох статичних сторінок. Також як джерело для зберігання інформації можна використовувати файли. Саме тому в Кохані спочатку відключена бібліотека для роботи з базою , але вона звичайно ж включена в збірку в якості офіційного модуля. Налаштування роботи цього модуля нічим не відрізняється від налаштування інших модулів, що знаходяться в папці `modules` . Для підключення модуля потрібно просто розкоментувати рядок з його назвою у файлі `bootstrap.php`:

```
/**
 * Enable modules. Modules are referenced by a relative or absolute path.
 */
Kohana::modules(array(
    ...
    'database' => MODPATH.'database', // Database access
    ...
));
```

Щоб використовувати базу даних , її потрібно звичайно ж створити . Як це зробити описується в наступних пунктах . Наступне , що потрібно зробити - відредагувати конфігураційний файл , де необхідно прописати налаштування доступу до свіжоствореної бази . Але оскільки в папці `modules` правити нічого не можна , для цього у нас є папка `application` , то конфігураційний файл потрібно туди скопіювати. Продублювавши файл `database.php` з «`modules / database / config /`» в «`application / config /`» і відкрити його. На даний момент необхідно тільки з'єднання `default` , тобто ось ця частина коду:

```
'default' => array
```

```
(
    'type'          => 'mysql',
    'connection' => array(
        'hostname'   => 'localhost',
        'database'   => 'kohana',
        'username'   => FALSE,
        'password'   => FALSE,
        'persistent' => FALSE,
    ),
    'table_prefix' => "",
    'charset'      => 'utf8',
    'caching'      => FALSE,
    'profiling'    => TRUE,
),
```

Нижче є ще одне з'єднання - `alternate` , яке також можна використовувати. Взагалі таких з'єднань можна створити скільки завгодно, якщо планується використовувати кілька баз даних. Тоді, на відміну від дефолтної бази, прийдеться вказувати її назву в запитах .

Але повернемося до налаштувань з'єднання `default` . Параметр `type` у першому рядку масиву - тип СУБД. У нашому випадку це « `mysql` ». Оскільки ми обрали саме цю систему управління , тому тут нічого міняти непотрібно. Слідом йде параметр `connection` , що відповідає за налаштування з'єднання з БД. Тут потрібно внести: назва хоста, назва бази даних, ім'я користувача, пароль користувача, чи підтримувати постійне з'єднання (булевий тип, за замовчуванням - `FALSE`, тобто не підтримувати) .

3.4.2 Розробка баз даних

Отже для реалізації веб-сайту було обрано скрипкову мову програмування PHP та СКБД MySQL. MySQL складається з двох частин: серверної і клієнтської. Сервер MySQL постійно працює на комп'ютері. Клієнтська програма, якою в даному випадку є PHP, посилає серверу MySQL SQL-запити через механізм сокетів (тобто за допомогою мережевих засобів), сервер їх обробляє і

запам'ятовує результат. Тобто скрипт (клієнт) вказує, яку інформацію він хоче отримати від сервера баз даних. Потім сервер баз даних посилає відповідь (результат) клієнтові (скрипту).

Для роботи з MySQL будемо використовувати PHP-функції, що призначені для роботи з СКБД. Також для обробки даних будуть використовуватись такі можливості PHP, як створення змінних, класів, використання логічних операцій порівняння даних, створення та перебір елементів масиву тощо.

Для розділення інформації, яка міститься на сайті та програми, яка її видає, використовуються бази даних MySQL. Вся інформація сайту буде зберігатися в базі даних, а скрипт PHP буде здійснювати вибірку необхідної інформації для конкретної веб-сторінки.

Для управління базами даних використаємо програму **PHPMyAdmin**. **PHPMyAdmin** – це сукупність скриптів написаних на PHP, що забезпечують повноцінну, в тому числі віддалену роботу з базами даних MySQL, через веб-інтерфейс. Так як phpMyAdmin дозволяє у багатьох випадках обійтися без безпосереднього введення команд SQL, то робота з базами даних є просою та швидкою. PHPMyAdmin встановлюється із базовим пакетом Denwer (доступ до нього здійснюється через адресу в адресному рядку браузера – <http://localhost/Tools/phpMyAdmin/>).

Структура MySQL трирівнева: бази даних - таблиці - записи. Для початку, в кореневому каталозі phpMyAdmin створимо нову базу даних, яка міститиме таблиці даних веб-сайту кафедри програмного забезпечення.

У рядку «Створити нову БД» введемо назву бази даних «kafedra» і обиремо кодування utf8_general_ci. Таке кодування дозволяє зберігати в таблицях даних символи кирилиці (при використанні іншого типу кодування, текстова інформація веб-сторінки може відображатись некоректно).

Отже, після того, як створена база даних «kafedra», наповнимо її таблицями, які будуть містити інформацію зі сторінок веб-сайту.

У рядку «Створити нову таблицю в БД kafedra» введемо назву таблиці «pages» та вкажемо чотири поля таблиці.

Створення таблиці pharماسу наведено на рисунку 3.1.

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	id	int(11)			No	None	auto_increment	[Icons]
☐	alias	varchar(255)	utf8_general_ci		Yes	NULL		[Icons]
☐	status	int(11)			Yes	NULL		[Icons]
☐	parent_id	int(11)			Yes	NULL		[Icons]
☐	lft	int(11)			Yes	NULL		[Icons]
☐	rgt	int(11)			Yes	NULL		[Icons]
☐	lvl	int(11)			Yes	NULL		[Icons]
☐	scope	int(11)			Yes	NULL		[Icons]
☐	img	varchar(255)	utf8_unicode_ci		Yes	NULL		[Icons]
☐	is_main	tinyint(4)			Yes	NULL		[Icons]
☐	visible	tinyint(4)			Yes	NULL		[Icons]
☐	blocked	tinyint(4)			Yes	NULL		[Icons]
☐	move	int(11)			Yes	NULL		[Icons]
☐	date_time	datetime			Yes	NULL		[Icons]

Рисунок 3.1 – Створення таблиці «Pages»

Дана таблиця буде містити перелік динамічних сторінок сайту.

Задамо назви полів:

1. «id». Дане поле будемо використовувати як первинний ключ (primary auto incremental key). Встановимо цілочисловий тип поля int. Дане поле буде унікальним і буде зростати при додаванні записів;
2. «alias». Дане поле зберігає назву сторінки транслітом, для подальшої побудови адреси;
3. «status». Поле яке вказує статус сторінки;
4. «parent_id». Вказує батьківську сторінку, одна із компонентів структури NestetSets.
5. «lft». Вказує на лівого потомка сторінки, одна із компонентів структури NestetSets.
6. «rgt». Вказує на правого потомка сторінки, одна із компонентів структури NestetSets.
7. «lvl». Зберігає значення рівня сторінки в структурі багаторівневої ієрархії, одна із компонентів структури NestetSets.
8. «scope». Забезпечує порядок виведення в межах своєї структури;
9. «img». Зберігає назву зображення для поточної сторінки;

10. «is_main». Зберігає показник того, чи є дана сторінка головною сторінкою сайт;
11. «visible». Показник для визначення чи потрібно цю сторінку відображати в меню;
12. «blocked». Показник сторінки, який вказує чи можна переходити на цю сторінку за її адресою.
13. «move». Забезпечує виведення елементів у вказаному порядку.
14. «date_time». Поле для збереження дати і часу створення сторінки.

Запит MySQL-серверу на створення таблиці має такий вигляд:

```
CREATE TABLE `pages` (  
  `id` INT(11) NOT NULL AUTO_INCREMENT,  
  `alias` VARCHAR(255) NULL DEFAULT NULL COLLATE 'utf8_general_ci',  
  `status` INT(11) NULL DEFAULT NULL,  
  `parent_id` INT(11) NULL DEFAULT NULL,  
  `lft` INT(11) NULL DEFAULT NULL,  
  `rgt` INT(11) NULL DEFAULT NULL,  
  `lvl` INT(11) NULL DEFAULT NULL,  
  `scope` INT(11) NULL DEFAULT NULL,  
  `img` VARCHAR(255) NULL DEFAULT NULL COMMENT 'картинка для цієї  
сторінки, яка не залежить від мови' COLLATE 'utf8_unicode_ci',  
  `is_main` TINYINT(4) NULL DEFAULT NULL COMMENT 'вказує на те, що ця  
сторінка є індексною',  
  `visible` TINYINT(4) NULL DEFAULT NULL COMMENT 'відображає сторінку  
у меню',  
  `blocked` TINYINT(4) NULL DEFAULT NULL COMMENT 'заборона доступу  
до сторінки',  
  `move` INT(11) NULL DEFAULT NULL,  
  `date_time` DATETIME NULL DEFAULT NULL,  
  PRIMARY KEY (`id`)  
)  
COMMENT='головна таблиця динамічних сторінок, без залежності від мови'  
COLLATE='utf8_unicode_ci'  
ENGINE=InnoDB  
AUTO_INCREMENT=13;
```

Заповнимо поля таблиці необхідними даними. Створену таблицю наведемо на рисунку 3.3.

	id	alias	status	parent_id	lft	rgt	lvl	scope	img	картинка для цієї сторінки, яка не закриває відмови	is_main	вказує на те, що ця сторінка є індексною	visible	відображає сторінку у меню	blocked	заборона доступу до сторінки	move	date_time
☐	1		NULL	0	1	2	1	1	NULL			1		0	0	0	1	NULL
☐	2	Poslugi	NULL	0	1	2	1	2	7jh8fev6psiogwdt.jpg			0		0	0	0	3	NULL
☐	3	questions	NULL	0	1	10	1	3	0vsaahq1ytcipgnj.jpg			0		0	0	0	5	NULL
☐	5	contacts	NULL	0	1	2	1	5	NULL			0		0	0	0	7	NULL
☐	7	pro_nas	NULL	0	1	2	1	6	NULL			0		0	0	0	2	2013-04-03 13:53:25
☐	9	questions/cat/6	NULL	3	2	3	2	3	NULL			0		0	0	0	9	2013-05-19 23:32:57
☐	10	questions/cat/7	NULL	3	4	5	2	3	NULL			0		0	0	0	10	2013-05-19 23:33:45
☐	11	questions/cat/8	NULL	3	6	7	2	3	NULL			0		0	0	0	11	2013-05-19 23:34:10
☐	12	questions/cat/9	NULL	3	8	9	2	3	NULL			0		0	0	0	12	2013-05-19 23:34:28

☐ Check All / ☐ Uncheck All With selected: ☐ ☐ ☐

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Рисунок 3.3 – Таблиця «pages»

Створимо користувача, який зможе працювати з базою даних. Для цього в phpMyAdmin заходимо в пункт «Привілегии» і вибираємо «Добавить нового пользователя». Надаємо йому всі права, для того, щоб мати змогу виконувати будь-які дії зі створеною базою даних. Створення користувача бази даних наведено на рисунку 3.4.

Добавить нового пользователя

Информация логина

Имя пользователя:

Хост:

Пароль:

Подтверждение:

Глобальные привилегии

Примечание: привилегии MySQL задаются по-английски

☐ Отметить все ☐ Снять отметку со всех

Данные	Структура	Администрирование
☒ SELECT	☒ CREATE	☒ GRANT
☒ INSERT	☒ ALTER	☒ SUPER
☒ UPDATE	☒ INDEX	☒ PROCESS
☒ DELETE	☒ DROP	☒ RELOAD
☒ FILE	☒ CREATE TEMPORARY TABLES	☒ SHUTDOWN
		☒ SHOW DATABASES
		☒ LOCK TABLES
		☒ REFERENCES
		☒ EXECUTE
		☒ REPLICATION CLIENT
		☒ REPLICATION SLAVE

Предел ресурсов

Замечание: Установка этих опций в 0 (ноль) удаляет лимит.

MAX QUERIES PER HOUR

MAX UPDATES PER HOUR

MAX CONNECTIONS PER HOUR

Рисунок 3.4 – Створення нового користувача бази даних

3.5 Розробка модуля управління новинами

Модуль управління новинами є важливим компонентом даного сайту, так як він призначений для створення у користувачів сайту відчуття, що даний сайт кафедри є динамічним і «живим», що постійно відбуваються якісь заходи, процеси на кафедрі і в університеті. Даний модуль є типовим для всього прокту і його структура майже нічим не відрізняється від усіх остальных, за винятком деяких деталей, які викликані його призначенням.

Новина — оперативне інформаційне повідомлення, яке містить суспільно важливу та актуальну інформацію, що стосується певної сфери життя суспільства загалом чи окремих його груп. В журналістиці - окремий інформаційний жанр, який характеризується стислим викладом ключової інформації щодо певної події, яка сталася нещодавно.

Модуль — функціонально завершений фрагмент програми, оформлений у вигляді окремого файлу з сирцевим кодом або його іменованої частини (наприклад, Active Oberon), призначений для використання в інших програмах. Модулі дозволяють розбивати складні задачі на менші відповідно до принципу модульності. Зазвичай проектується таким чином, щоб надати програмістам зручну для багаторазового використання функціональність (інтерфейс).

Структурно модуль розбитий на дві частини: клієнтську і адміністративну. Кожна з них відповідає за своє функціональне навантаження і має різний ступінь доступу. Кожна з наведених частин відповідно розбита на свою структурні одиниці у відповідності по шаблона проектування HMVC. HMVC (англ. Hierarchical model–view–controller) - Ієрархічні Модель-Вид-Контролер, одне з розширень архітектурного паттерна MVC, що дозволяє вирішити деякі проблеми масштабованості програм, що мають класичну MVC-архітектуру. На рисунку 3.5 наведено структуру папок даного модуля з відповідними файлами в ньому.

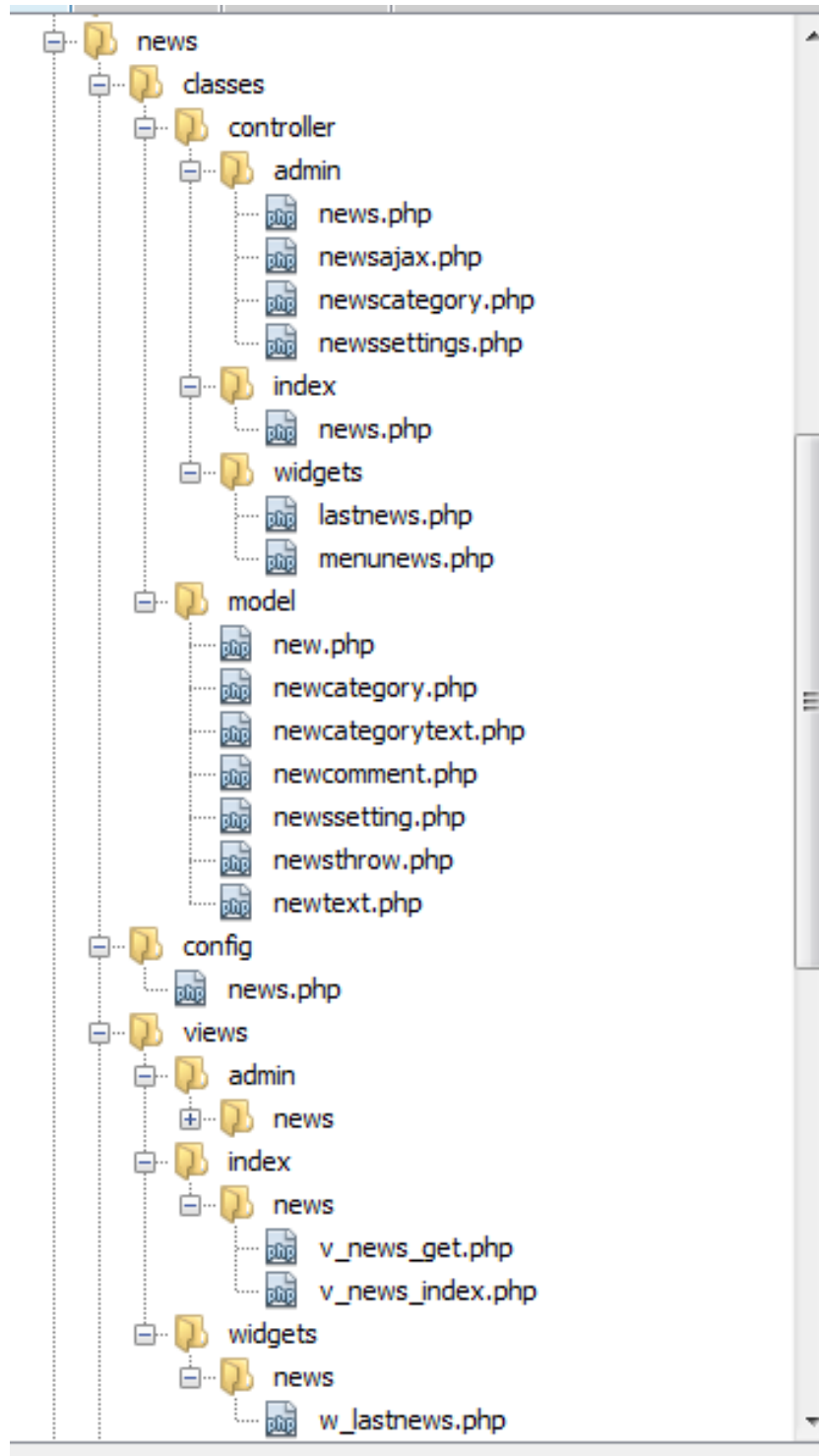


Рисунок 3.5 – Структура папок модуля новин

У відповідності до шаблону проектування MVC даний модуль містить контролери, моделі і види, кожні з яких розміщені у відповідній папці. В папці controller розміщені класи, якій відповідають за логіку роботи модуля для адміністративно і клієнтської частини, також винесений окремо блок управління

відметами. Контроллери виступають як проміжковий рівень між моделями і видами. Структуру класів наведено в спрощеній UML діаграмі на рисунку 3.6.

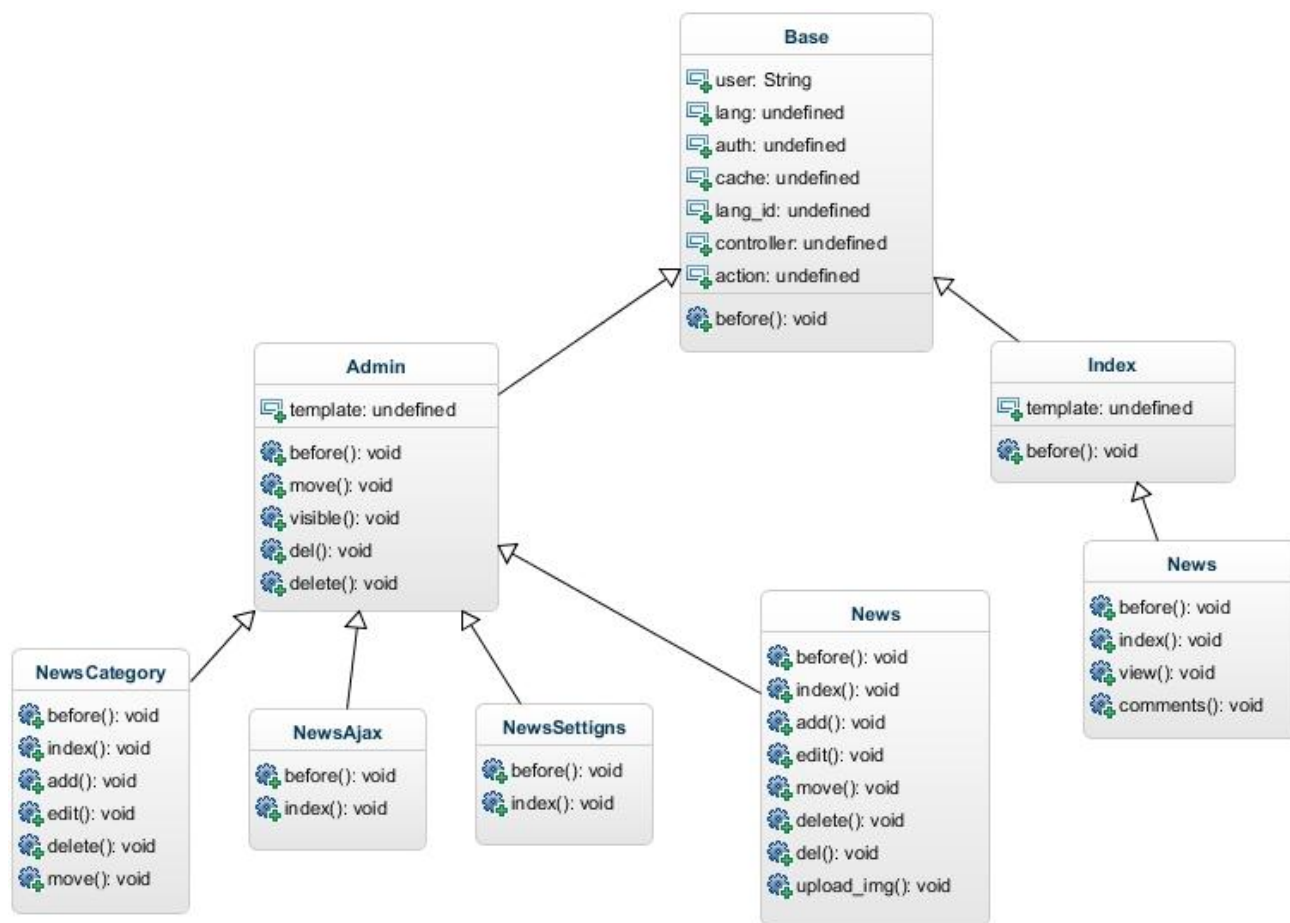


Рисунок 3.6 – Спрощена UML діаграма класів модуля новин

У відповідності до вибраної архітектури, необхідно реалізувати моделі, які будуть виступати у ролі посередника між скриптом і базою даних. База даних повинна відповідати усім поставленим вимогам і будуватись у відповідності до моделей. Побудуємо ER-діаграму таблиць бази даних, яка буде описувати структуру і відношення таблиці. Для побудови використаємо програму MySQL Workbench, і після побудови моделі синхронізуємо її з базою даних. На рисунку 3.7 показана діаграма поделі.

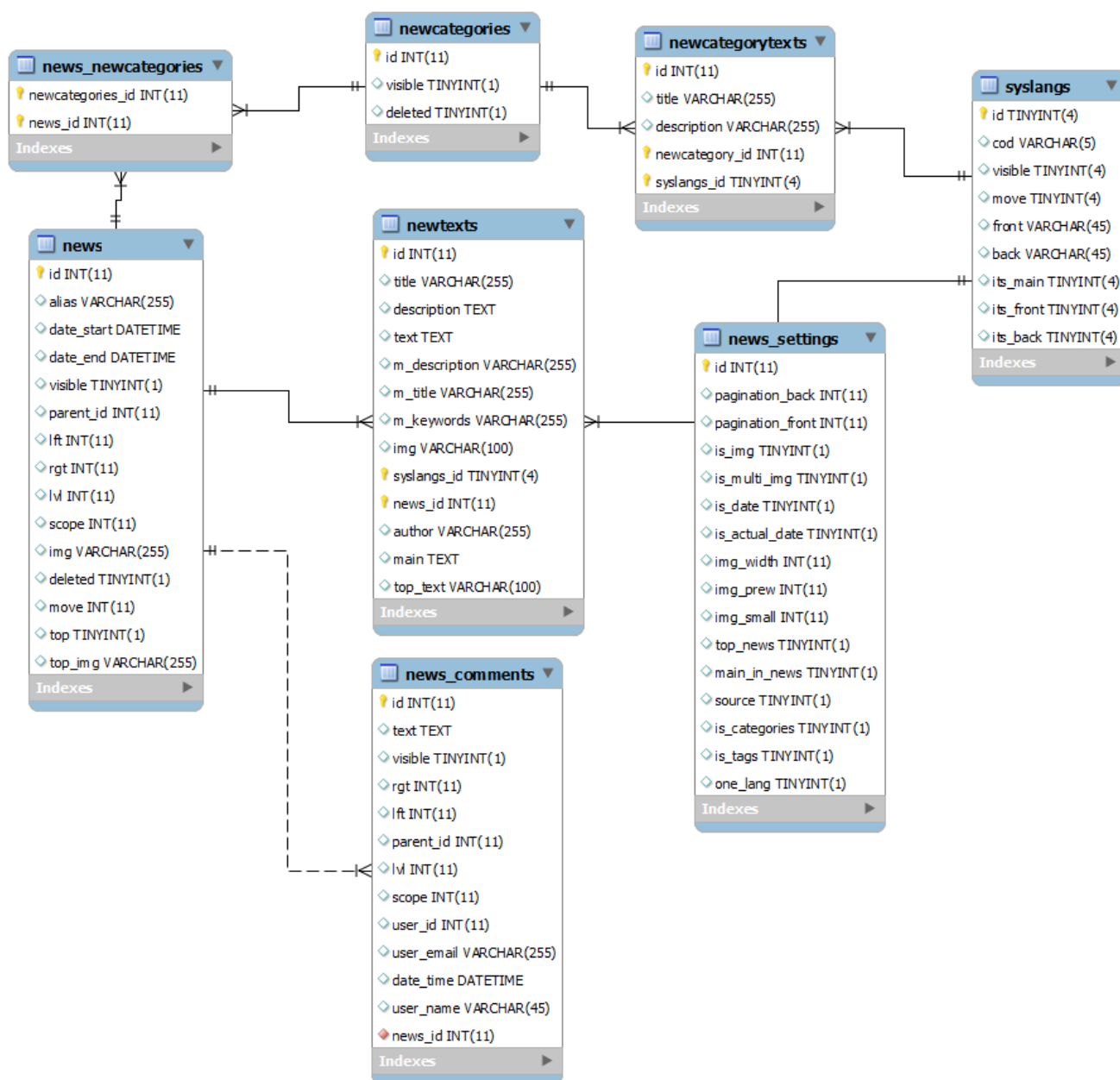


Рисунок 3.7 – ER-діаграма таблиць бази даних модуля

Наданій моделі показані усі необхідні таблиці для функціонування даного модуля. Їхню структуру можна описати кількома словами: є головна таблиця новин, де зберігаються дані про новину, не прив'язані до мови; є таблиця збереження категорій новин, їх ще можна назвати рубриками. Оскільки кожна новина може належати декільком рубрикам і навпаки, то між ними повинен бути зв'язок багато до багатьох, оскільки він важко реалізовується, то заміняється на два зв'язки один до багатьох, і з створенням додаткової проміжкової таблиці.

Оскільки новини можуть бути на декількох мовах, то для цього створюються додаткові таблиці, де зберігаються дані, які прив'язані до якоїсь конкретної мови. Дані таблиці зв'язані з таблицями новин і категорій, а також з таблицею мов, оскільки можна зберігати новини лише на тих мовах, які є в системі, і при видаленні якоїсь мови з системи, каскадно повинні видалитись усі записи, які використовують дану мову.

Вихідний код модуля наведено в додатку А з детальними поясненнями у вигляді коментарів.

3.6 Висновки

В даному розділі було розглянуто основні принципи розробки динамічних сайтів. Проведений варіантний аналіз мов програмування та СКБД, що використовуються при розробці веб-сайтів.

Також було розроблено засобами MySQL базу даних та таблиці, які використовувались для обробки за допомогою скриптів php.

Використовуючи PHP і MySQL на основі фреймворка Kohana був розроблений модуль управління новинами на сайті. Для нього створенні відповідні таблиці в базі даних, реалізовані скріпти, що забезпечують повне функціонування модулю і відповідати усім поставленим йому завданням, а саме додавання, редагування, видалення новин і окремих складових.

4 ОХОРОНА ПРАЦІ

4.1 Характеристика професії та приміщення

Загальну характеристику професії «Оператор комп'ютерного набору» наведено у вигляді таблиці[1-2].

Таблиця 4.1 – Загальна характеристика професії «Оператор комп'ютерного набору».

Назва	Опис
1	2
Розділ	3 Фахівці
Галузі застосування знань	Цей розділ вміщує професії, що вимагають знань в одній чи більше галузях природознавчих, технічних і гуманітарних наук. Професійні завдання полягають у виконанні спеціальних робіт, пов'язаних із застосуванням положень та використанням методів відповідних наук.
Кваліфікаційні вимоги	До цього розділу належать професії, яким відповідає кваліфікація за дипломом чи іншим відповідним документом: • молодшого спеціаліста; • бакалавра; • спеціаліста, що проходить післядипломну підготовку (стажування, інтернатуру, клінічну ординатуру тощо); спеціаліста (на роботах з керування складними технічними комплексами чи їх обслуговування).
Код професії	4112
Код за ДКХПП	1, 18

Продовження таблиці 4.1.

Завдання та обов'язки в сфері охорони праці	• Приймає участь у роботі комісії по комплексній перевірці охорони праці в ввірених підрозділах підприємства. Перевіряє виконання заходів по покращенню і оздоровленню умов праці приписів контролюючих органів, організацію навчання і інструктажів робітників, приймає оперативні заходи по усуненню виявлених недоліків. • Приймає участь у проведенні контролю стану охорони праці у відділі. • Здійснює комплексні і цільові перевірки стану охорони праці у встановлені терміни.. • Здійснює контроль своєчасного виконання перспективних поточних, оперативних планів роботи з охорони праці в ввірених підрозділах. • Здійснює контроль за дотриманням керівниками ділянок, вимог правил і норм безпеки тощо. • Працівник зобов'язаний знати та дотримуватися правил особистої гігієни: приходити на роботу в чистому одязі і взутті; постійно стежити за чистотою тіла, рук, волосся; мити руки з милом після відвідування туалету, дотику з забрудненими предметами, після закінчення роботи. • Підготувати робочу зону для безпечної роботи: перевірити оснащеність робочого місця, справність обладнання, електропроводки на видимі пошкодження. При несправності повідомити безпосередньому керівникові. Перевірити зовнішнім оглядом достатність освітлення і справність вимикачів і розеток тощо. • Забороняється зберігати на своєму робочому місці пожежо та вибухонебезпечні речовини.
--	---

Об'єктом дослідження в бакалаврській роботі обрано відділ програмного забезпечення м. Вінниці. Приймаємо, що приміщення відділу знаходиться на шостому поверсі адміністративної будівлі. Характеристики приміщення наступні:

- загальна площа – 30 м²;
- висота – 2,5 м;
- об'єм приміщення – 75 м³;
- приміщення має 2 вікна (довжина 1,2 м., висота 1,4 м.);

- кількість працюючих у приміщенні – 6 осіб;
- кількість робочих місць, обладнаних ЕОМ – 6.

В такому випадку, на одного працюючого в приміщенні припадає: $30/6=5$ (м²/ос.) робочої площі.

Згідно із СНиП 2.09.04 – 87[2] на кожного працюючого в управлінських приміщеннях повинно припадати не менше 4 (м²/чол.) робочої площі. Відповідно до НПОАП 0.00-1.28-10 [3] на одного працівника, який застосовує ЕОМ має припадати не менше 6(м²/чол.), а об'єм повітря в приміщенні має бути не менше 20 (м³/чол.)(у нашому випадку $(75/6=12,5$ м³). Висота приміщення – не менше 2,5 м. Отже, можна зробити висновок, що нормативу забезпечення працюючих площею та об'ємом повітря в відділі недотримано.

У приміщенні розміщено 6 письмових столів, 6 комп'ютерів, одна шафа для документів та один холодильник.

За небезпекою ураження електричним струмом приміщення відділу програмного забезпечення можна віднести до приміщень без підвищеної небезпеки ураження електричним струмом працюючих. Схему приміщення наведено на рисунку 4.1. За типовими нормами належності вогнегасників НАПБ Б.03.001–2004 для приміщення відділу програмного забезпечення варто обрати один вуглекислотний вогнегасник ВВК-3,5. Тип пожежної сигналізації – тепловий автоматичний сповіщувач, що реагує на певний поріг температури в повітрі.

Схему приміщення наведена на рисунку 4.1 [9].

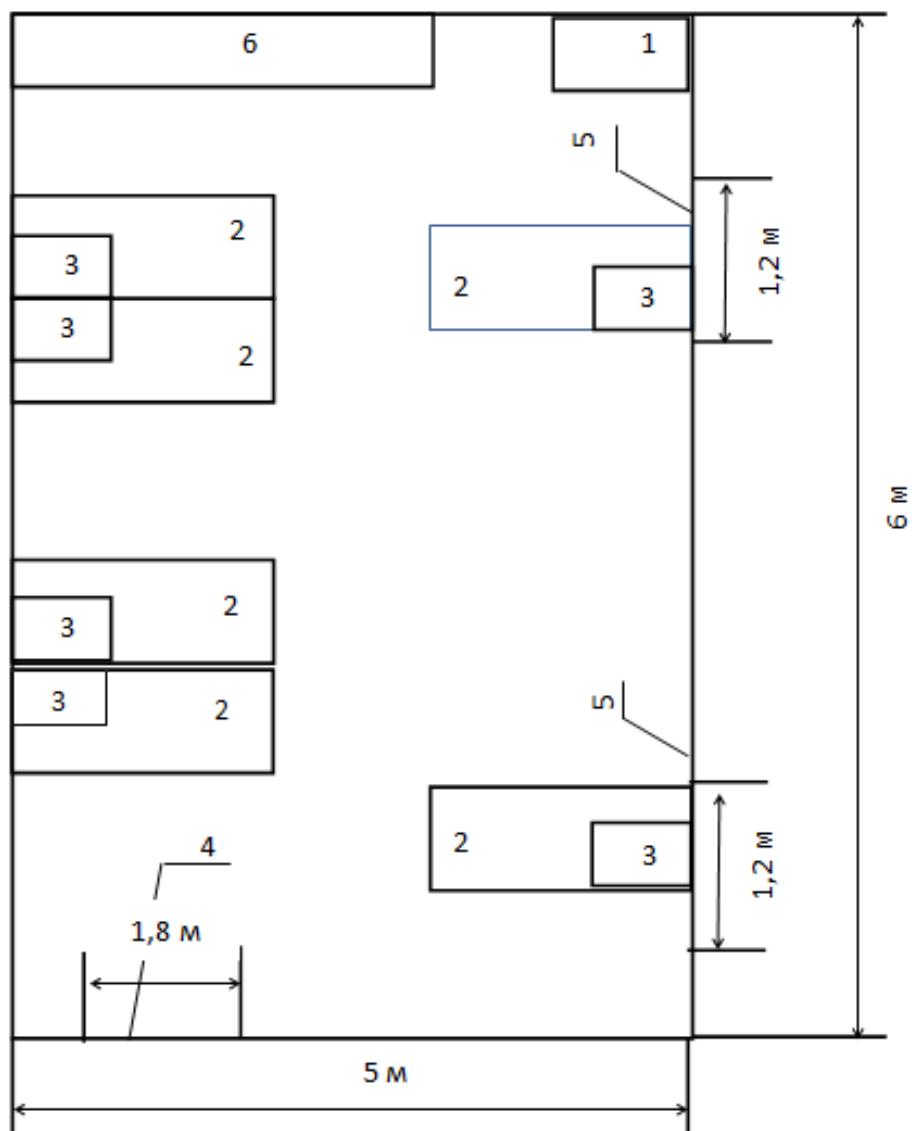


Рисунок 4.1 – Схема приміщення

- де: 1 – холодильник;
 2 – робочі місця (столи);
 3 – персональні комп'ютери;
 4 – дверний отвір;
 5 – вікна (розмір вікна 1,2×1,4 м.);
 6 – шафа.

4.2 Аналіз стану умов охорони праці в приміщенні

Аналіз умов робочого місця проводиться в теплий період року. Нижче наведемо вихідні дані аналізу:

- робоче місце є непостійним для робітника,
- виконуються роботи Легка Іб;
- температура – 20°C;
- вологість – 70%;
- швидкість руху повітря – 0,45 м/с;
- концентрація оксиду вуглецю – 22 мг/м³,
- озону – 0,23 мг/м³;
- оксиду азоту – 7,6 мг/м³;
- Формальдегіду – 0,557мг/м³.

Згідно санітарних норм мікроклімату виробничих приміщень ДСН 3.3.6.042–99[5] категорії робіт л Легка Іб наведено такі норми мікроклімату в приміщенні (таблиця 4.2).

Таблиця 4.2 – Опимальні та допустимі характеристики мікроклімату [5,6]

Період року	Категорія робіт	Робоче місце	Температура повітря, °C	Відносна вологість, не більше %	Швидкість руху повітря, м/с	Концентрація шкідливих речовин оксид вуглецю (во), озон (о), оксид азоту (оа), формальдегід (ф) (не більше мг/м³)			
						во	о	оа	ф
Теплий	Іб	Н	Оптимальні норми			20	0,1	2	0,5
			21-23	40-60	0,1				
			Допустимі норми						
			17-25	75%	0,1				

Отже, робимо висновок, що нормативам не відповідають такі показники: швидкість руху повітря (вище норми), концентрація оксиду вуглецю,

формальдегіду, озону та оксиду азоту в повітрі. Все це чинить шкідливий вплив на робітників приміщенні, тому умови праці потрібно негайно нормалізувати.

Для оцінки ефективності природного освітлення наведемо схему приміщення для розрахунку (рисунок 4.2-4.3).

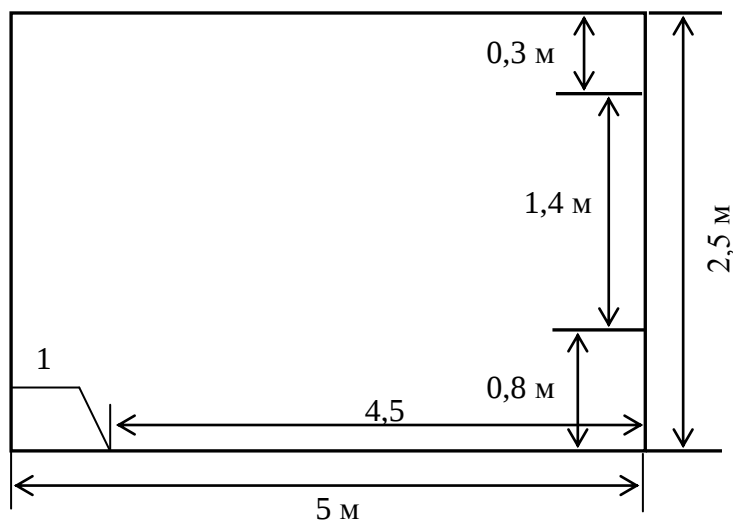


Рисунок 4.2 – Схема приміщення, вид збоку

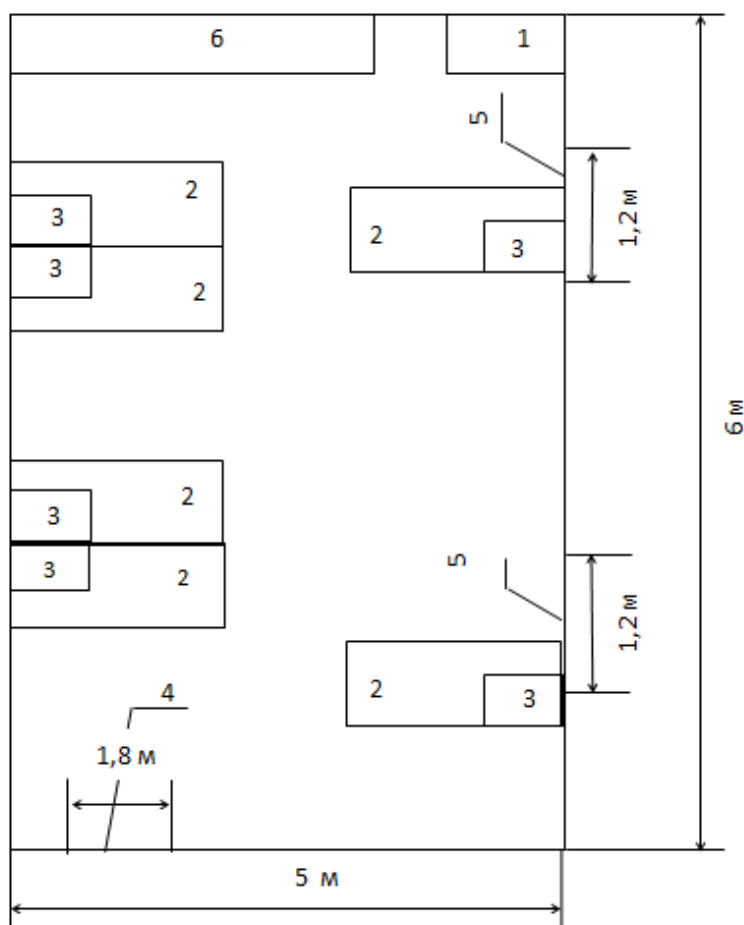


Рисунок 4.3 – Схема приміщення, вид зверху

Нормоване значення коефіцієнта природного освітлення (КПО) визначається у відсотках, і знаходиться за формулою: $e_N = e_n \cdot m_N$ [11], де e_N – нормоване значення КПО, e_n – значення КПО за ДБН В.2.5–28–2006 «Природне і штучне освітлення», m_N – коефіцієнт світлового клімату.

Значення КПО беремо з ДБН В.2.5–28–2006 «Природне і штучне освітлення». Воно буде становити 1,5. Прийmemo, що вікна нашого приміщення виходять на північ. Тоді коефіцієнт світлового клімату буде дорівнювати 0,9.

Знаходимо нормоване значення КПО.

$$e_N = 1,5 \cdot 0,9 = 1,35 \quad (4.1)$$

Для робочих місць, де застосовується ЕОМ, значення КПО має становити не менше 1,5%. Для визначення достатності природного [11] освітлення потрібно розрахувати фактичне значення КПО, використовуючи формулу

$$e_{\Phi} = \frac{100 \cdot S_B \cdot \tau_3 \cdot r_1}{S_n \cdot \eta_B \cdot K_3 \cdot K_{б\text{уд}}} \quad (4.2)$$

Площа усіх вікон у приміщенні буде становити $S_B = 3,36 \text{ м}^2$.

Площа підлоги приміщення $S_n = 30 \text{ м}^2$.

Визначимо загальний коефіцієнт світлопроникності віконного прорізу τ_3 .

$$\tau_3 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4 \cdot \tau_5 \quad [11]$$

$\tau_1 = 0,65$, $\tau_2 = 0,9$, $\tau_3 = 1$, $\tau_4 = 1$, $\tau_5 = 1$. Отже, $\tau_3 = 0,585$.

Розрахуємо світлову характеристику вікна η_B за даними, наведеними в таблиці. Відношення довжини приміщення L до його глибини $B = 6/5 = 1,2$.

Відношення глибини приміщення B до висоти від рівня робочої поверхні до верхнього краю вікна $h = 5/1,4 = 3,57$.

Отже, $\eta_B = 18$.

Оскільки відношення відстані між протилежними будівлями до висоти карнизу протилежного будинку становлять 1 м, то $K_{б\text{уд}} = 1,4$.

Приймаємо, що коефіцієнт запасу $K_3 = 1,3$.

Для розрахунку коефіцієнта r_1 необхідно розрахувати такі параметри:

1) середньозважений коефіцієнт відбиття;

2) відношення глибини приміщення до висоти від рівня умовної робочої поверхні до верху вікна;

3) відношення відстані до розрахункової точки від зовнішньої стіни до глибини приміщення;

4) відношення довжини приміщення до його глибини.

Розрахуємо середньозважений коефіцієнт відбиття за формулою 4.3:

$$P_{\text{ср}} = \frac{P_{\text{стелі}} \cdot S_{\text{стелі}} + P_{\text{стін}} \cdot S_{\text{стін}} + P_{\text{підлоги}} \cdot S_{\text{підлоги}}}{S_{\text{стелі}} + S_{\text{стін}} + S_{\text{підлоги}}} \quad (4.3)$$

де: $P_{\text{стелі}}$, $P_{\text{стін}}$, $P_{\text{підлоги}}$ – відповідні коефіцієнти відбиття;

$S_{\text{стелі}}$, $S_{\text{стін}}$, $S_{\text{підлоги}}$ – відповідні площі поверхонь.

Розрахуємо середньозважений коефіцієнт відбиття $P_{\text{ср}}$. Оскільки стеля свіжопобілена, то $p_{\text{стелі}} = 0,8$. Стіни обклеєні темними шпалерами, то $p_{\text{стін}} = 0,15$.

Підлога пофарбована у оливково-зелений колір, то $p_{\text{підлоги}} = 0,2$. Підрахуємо також площу стін. $S_{\text{стін}} = 50 \text{ м}^2$.

$$P_{\text{ср}} = \frac{0,8 \cdot 30 + 0,15 \cdot 50 + 0,2 \cdot 30}{30 + 50 + 30} = 0,34$$

Отже, $P_{\text{ср}} = 0,34$.

Відношення глибини приміщення B до висоти від рівня умовної робочої поверхні до верху вікна: $5/1,4 = 3,57$.

Відношення відстані до розрахункової точки від зовнішньої стіни 1 до глибини приміщення: $4,5/5 = 0,9$.

Відношення довжини приміщення L до його глибини B : $6/5 = 1,2$.

Враховуючи знайдені вище співвідношення знаходимо, що коефіцієнт, який враховує відбиття світла від внутрішніх поверхонь буде дорівнювати $r_1 = 1,85$.

Тепер розрахуємо фактичне значення природного освітлення.

$$e_{\text{ф}} = \frac{100 \cdot 3,36 \cdot 0,585 \cdot 1,85}{30 \cdot 18 \cdot 1,3 \cdot 1,4} = \frac{363,636}{982,8} = 0,37$$

Як показало дослідження, фактичне значення природного освітлення майже в 3 рази нижче нормованого ($0,37 < 1,35$), то природне освітлення в приміщенні відділу недостатнє і потрібно його покращити.

Для освітлення відділу застосовуються люмінесцентні лампи потужністю 60 Вт. Система освітлення – загальна. Кількість світильників – 2 штук, а кількість ламп у світильнику – 2 штуки. Отже, значення освітленості згідно встановлених нормативів повинне становити не менше 300 люкс (за ДБН В. 2.5–28–2006) [7].

Схему розміщення світильників у приміщенні відділу наведено на рисунку 4.4.

Визначимо фактичне значення освітлення (E_f). Фактичне значення штучного освітлення (E_f) розраховуємо за формулою[4.4]:

$$E_f = \frac{F_l \cdot \eta_B \cdot N \cdot n}{S \cdot k \cdot z} \quad (4.4)$$

де F_l – світловий потік однієї лампи, лм;

η_B – коефіцієнт використання світлового потоку. Для світильників, які використовуються в адміністративних будівлях для традиційних розмірів приміщення і кольорового оздоблення, може набувати значення в межах $\eta_B = 0,4 - 0,6$;

N – кількість світильників у приміщенні, яке досліджується, шт.;

n – кількість ламп у кожному світильнику, шт.;

S – площа приміщення, м²;

k – коефіцієнт запасу.

z – коефіцієнт нерівномірності освітлення: для ламп розжарювання $z = 1,15$; для люмінесцентних ламп $z = 1,1$.

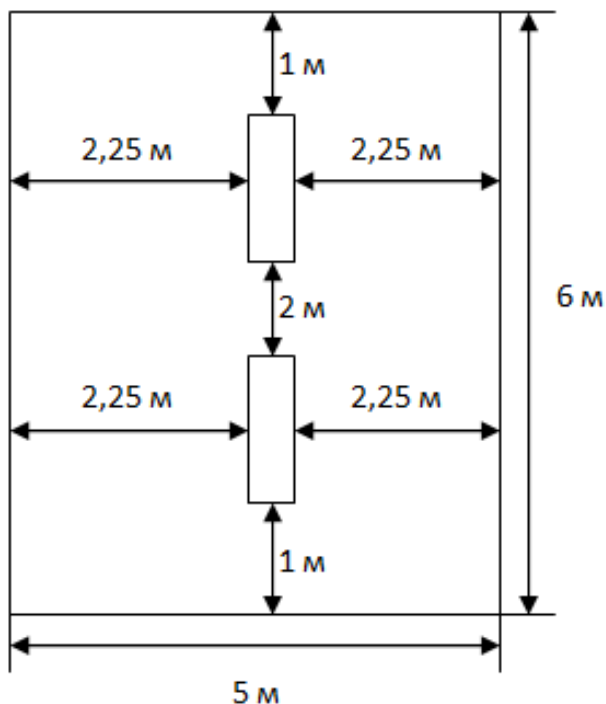


Рисунок 4.4 – Розміщення світильників

Розрахуємо фактичне значення освітлення E_{ϕ} .

Приймемо середнє значення коефіцієнта використання світлового потоку.

$\eta_{\text{в}} = 0,5$.

Кількість світильників $N = 2$. Кількість ламп у світильнику $n = 2$.

Площа приміщення $S = 30 \text{ м}^2$.

Коефіцієнт запасу $K = 1,5$.

Коефіцієнт нерівномірності освітлення $z = 1,1$.

Згідно з таблицею 11 вибираємо тип лампи – ЛДУ 80 (світловий потік буде становити 3740 лм) Підставимо дані значення для обчислення фактичного значення освітлення.

$$E_{\phi} = \frac{3740 \cdot 0,5 \cdot 2 \cdot 2}{30 \cdot 1,5 \cdot 1,1} = 151$$

Згідно з отриманим значенням освітлення можна сказати, що приміщення освітлюється погано за допомогою штучного освітлення ($151 < 300$).

4.3 Розробка заходів щодо поліпшення умов праці

Під час виконання бакалаврської роботи були отримані певні результати. Занесемо їх у таблицю 4.3.

Таблиця 4.3 – Розробка заходів покращення стану охорони та умов праці

	Виявлені недоліки	Заходи покращення умов праці
	Швидкість вітру значно перевищує норму	Потрібно вжити заходи, щодо реорганізації системи штучної та природної вентиляції
	Концентрація оксиду вуглецю, озону та оксиду азоту перевищує граничну допустиму концентрацію	
	Недостатність природного освітлення в приміщенні	Використовувати в приміщенні світлі штори. Застосовувати при необхідності штучне освітлення. Використовувати матеріали з максимальним коефіцієнтом відбиття світла від внутрішніх поверхонь приміщення (світлі шпалери та підлога).
	Низьке значення фактичного штучного освітлення	Застосовувати лампи більшої потужності, збільшити їхню кількість.

4.4 Висновки до розділу 4

У ході виконання бакалаврської роботи було охарактеризовано професію «Оператор комп'ютерного набору». Проаналізувавши вхідні параметри, було наведено загальну характеристику приміщення, яке використовується з метою

набору комп'ютерного тексту. Також було зображено схему приміщення з усіма необхідними позначеннями.

На основі вхідних даних (температура повітря, відносна вологість, швидкість руху повітря та ін.) було проаналізовано стан метеорологічних умов на робочому місці. Було з'ясовано, концентрація яких саме шкідливих речовин перевищує гранично-допустиму концентрацію.

Взявши за основу розміщення вікон, їхню кількість, характеристики а також деякі параметри приміщення було проведено оцінку ефективності природного освітлення. Взявши до уваги тип і характеристики ламп, виконано оцінку ефективності штучного освітлення.

Усі рекомендації, стосовно поліпшення умов мікроклімату, тощо, були внесені у спеціальну таблицю.

5 ТЕСТУВАННЯ РОБОТИ ВЕБ-САЙТУ

У процесі створення сайту, на етапі розробки, дуже важливо перевірити працездатність сайту перед його розміщенням на хостингу. Для перевірки працездатності сайту проводиться спеціальне тестування.

Процес тестування стосується як програмної частини, так і частини користувацького інтерфейсу сайту.

Даний етап роботи є надзвичайно важливим, так як допомагає визначити всі помилки та недоробки, що виникли у процесі розробки сайту.

5.1 Тестування адмін-панелі сайту

В процесі тестування було здійснено такі перевірки:

- додавання, редагування та видалення новин;
- додавання та видалення користувачів сайту;
- формування динамічних сторінок;
- додавання зображень до контенту сайту.

Адмін-панель сайту кафедри наведено на рисунку 5.1

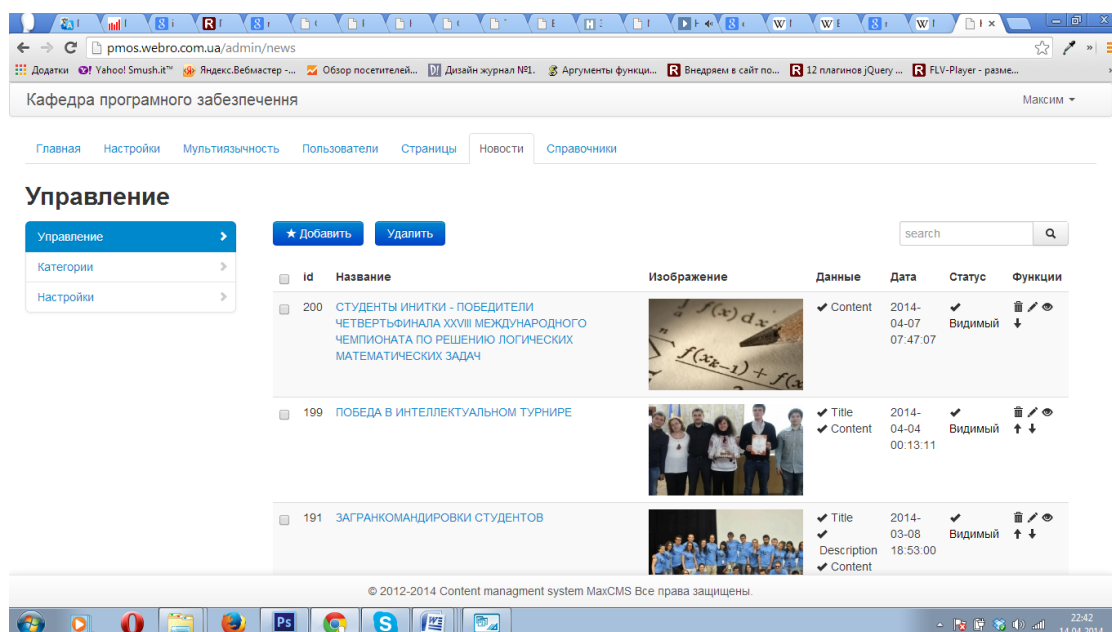


Рисунок 5.1 – Адмін-панель сайту кафедри

5.2 Тестування клієнтської частини сайту

Перевіримо правильність виведення інформації про викладачів для клієнтської частини. Приклад виведення даних наведено на рисунку 5.2.

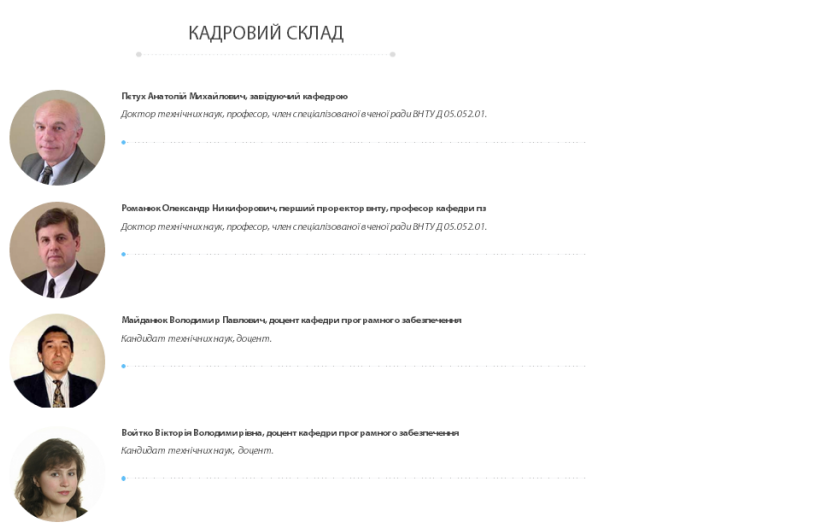


Рисунок 5.2 – Виведення даних для клієнтської частини

5.3 Тестування користувацького інтерфейсу сайту

Під час тестування сайт перевіряється на відповідність технічному завданню, перевіряються його технічні характеристики.

В процесі тестування було здійснено такі перевірки:

- переглянено сайт на різних моніторах. Переглядаючи сайт на моніторах різних розмірів і при різному дозволі, дизайн сайту на комп'ютерах потенційних відвідувачів не змінився, не з'явилась горизонтальна смуга прокручування на маленьких екранах, елементи дизайну при різному дозволі не змістились й т.д.
- переглянено сайт в різних браузерерах. Переглядаючи сайт в різних інтернет-браузерах і їхніх версіях, перевірено кросбраузерність сайту.

Для перевірки кросбраузерності було обрано чотири найпоширеніших браузера: Internet Explorer, Firefox, Opera та Google Chrome. Переглядаючи даний сайт на різних браузерах впевнились у коректності відображення інформації різними браузерами;

- перевірено швидкість завантаження сайту. Сайт завантажуються досить швидко, що робить його зручним у використанні;
- перевірено працездатність гіперпосилань на сайті. Під час перевірки не було виявлено посилань, які не працюють або ведуть не на ту сторінку.
- перевірено коректність відображення кольорів на сайті при різних налаштуваннях колірної палітри монітора, а також відображення шрифтів, анімації та графічних зображень;
- перевірено властивості кожної сторінки сайту: заголовки, ключові слова, описи та інші мета-теги;
- перевірено зміст кожної сторінки на відповідність вихідним матеріалам замовника й правопису на кожній сторінці.

Отже в ході тестування користувацького інтерфейсу сайту було виявлено і усунуто певні недоліки сайту, які могли б вплинути на коректність роботи та зручність даного сайту. Тепер веб-сайт працює коректно, однаково виглядає на моніторах різних розмірів, однаково відображається в найпопулярніших браузерах, зберігаючи при цьому правильне розміщення елементів дизайну (кольори, шрифти, анімації та графічні зображення тощо). Всі гіперпосилання даного сайту робочі, працюють правильно. Висока швидкість завантаження сторінки.

5.4 Тестування роботи бази даних

Перевіримо правильність налаштування зв'язку з базою даних. Для цього розглянемо елементарний приклад роботи скрипта завантаження сторінки «Кадровий склад», за допомогою якого в циклі виводиться перелік всіх

викладачів кафедри. Приклад використання розробленого скрипта наведено на рисунку 5.3.

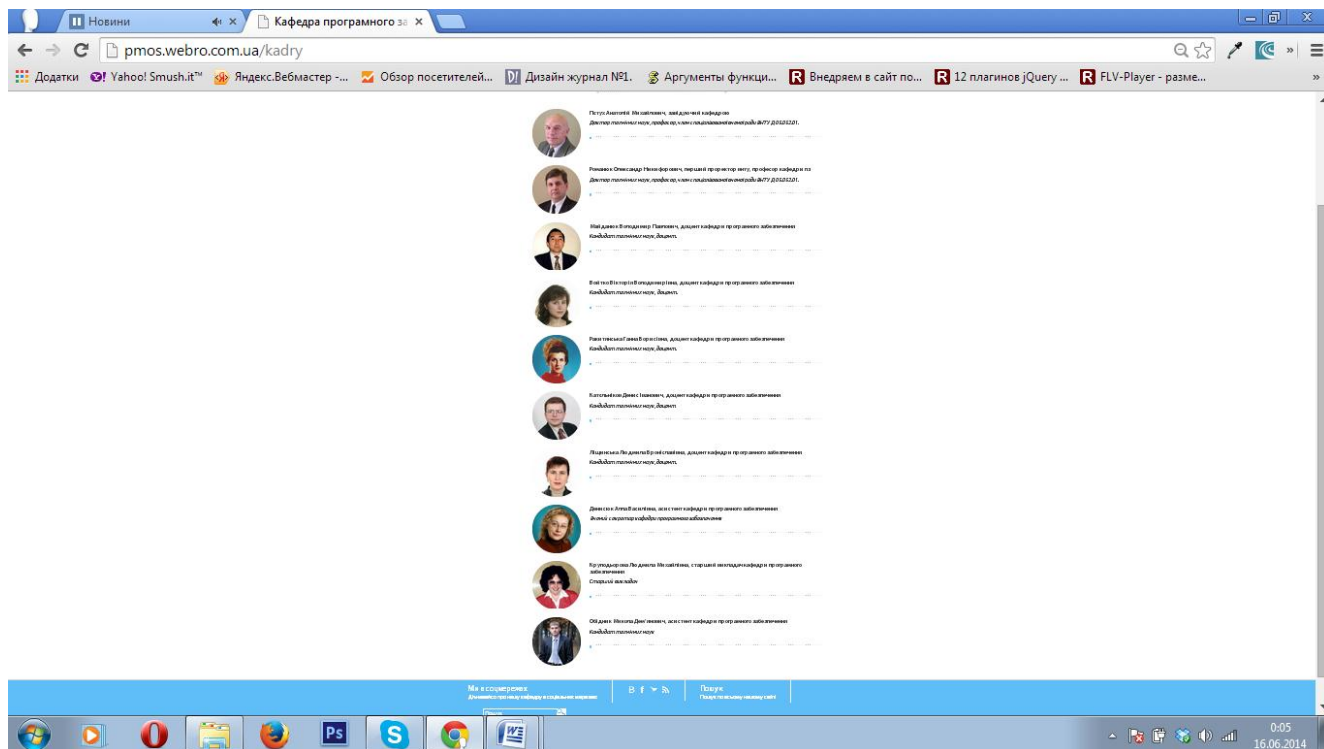


Рисунок 5.3 – Робота скрипта завантаження сторінки «Кадровий склад»

Як бачимо з рисунку 5.3, дані витягнуті з бази даних “kafedra” відображаються коректно, що свідчить про правильне налаштування з’єднання з базою даних.

За таким ж принципом було створено блок новин, який виводить останні новини кафедри, інформуючи всіх відвідувачів про останні новини кафедри на рисунку 5.4.



Рисунок 5.4 – Приклад роботи скрипта завантаження новин

5.5 Висновки

Отже в даному розділі була здійснена перевірка працездатності сайту перед його розміщенням на хостингу. Тестування веб-сайту проведено у двох напрямках: тестуванні користувацького інтерфейсу сайту та адміністративної частини.

В ході тестування користувацького інтерфейсу сайту було виявлено і усунуто певні невеличкі недоліки сайту. Тепер веб-сайт працює коректно, однаково виглядає на моніторах різних розмірів, однаково відображається в найпопулярніших браузерях, зберігаючи при цьому правильне розміщення елементів дизайну (кольори, шрифти, анімації та графічні зображення тощо). Всі гіперпосилання даного сайту робочі, працюють правильно. Швидкість завантаження сторінки висока.

При тестуванні адміністративної частини, було перевірено правильність роботи з базою даних, протестовано модулі реєстрації, управління користувачами, динамічними сторінками та новинами. В ході тестування не було виявлено збоїв роботи розроблених модулів. Після проведеного тестування можна зробити висновок, що веб-сайт відлагоджений і готовий до розміщення на хостингу.

ВИСНОВКИ

У бакалаврській дипломній роботі розроблено динамічний сайт кафедри програмного забезпечення. Проведено аналіз розвитку сучасних web-технологій та обґрунтовано здійснений вибір засобів реалізації сайту. Здійснено детальний аналіз предметної області, визначено суть технічної проблеми, актуальність питання, проаналізовано сучасний стан та перспективи розвитку інформаційних сайтів-візиток кафедр навчальних закладів. Обґрунтовано необхідність розробки сайту.

Також у роботі проаналізовано принципи реалізації сайту та розроблено функціональну структуру веб-сайту. Для успішної розробки програмного продукту розроблено адмін-панель сайту, велику кількість модулів для роботи з контентом, розроблена загальна логічна структура сайту.

У роботі було розглянуто основні принципи розробки динамічних сайтів. Проведений варіантний аналіз мов програмування та СКБД, що використовуються при розробці веб-сайтів.

Використовуючи PHP і MySQL, було розроблено модуль динамічних сторінок, сторінок інформації про викладачів, контактну сторінку, сторінки списків і перегляду новин.

Розроблений сайт містить загальну інформацію про історію кафедри, наукову та методичну роботи, детальну інформацію про викладачів та переліки останніх новин кафедри.

Вся інформація, викладена на сайт, є достовірною та узгодженою з викладачами кафедри програмного забезпечення.

ПЕРЕЛІК ПОСИЛАНЬ

1. Веб-сайт [електронний ресурс] // Режим доступу:
<http://www.webtec.com.ua/uk/articles/index/view/2011-05-05/web-site> – Назва з екрану.
2. Компанія Nexo [електронний ресурс] // Режим доступу: <http://nexo-sa.com/en/> – Назва з екрану.
3. Компанія QSC [електронний ресурс] // Режим доступу: <http://qsc.com/> – Назва з екрану.
4. Компанія CrownAudio [електронний ресурс] // Режим доступу:
<http://www.crownaudio.by/> – Назва з екрану.
5. Б. Хоган HTML5 и CSS3. Веб-разработка по стандартам нового поколения– 4-е изд. – СПб.: Издательский дом «Питер», 2010. – 260 с.
6. Питер Лабберс, Брайан Олберс, Фрэнк Салим HTML5 для профессионалов. Мощные инструменты для разработки современных веб-приложений – СПб.: Издательский дом «Питер», 2011. – 272 с.
7. Flat design [електронний ресурс] // Режим доступу: <http://fltdsgn.com/> – Назва з екрану.
8. Скевоморфизм [електронний ресурс] // Режим доступу:
<http://ru.wikipedia.org/wiki/Скевоморфизм> – Назва з екрану.
9. Wordpress [електронний ресурс] // Режим доступу:
<http://mywordpress.ru/about/> – Назва з екрану.
10. Drupal [електронний ресурс] // Режим доступу: <http://drupal.ua/> – Назва з екрану.
11. Про Joomla [електронний ресурс] // Режим доступу: <http://joomla-master.org/> – Назва з екрану.
12. MaxCMS! [електронний ресурс] // Режим доступу:
<http://ru.wikipedia.org/wiki/MaxCMS!> – Назва з екрану.
13. Сравнительная характеристика трех CMS: Drupal, MaxCMS и Wordpress [електронний ресурс] // Режим доступу:
<https://sites.google.com/site/sravnitelnyeharakteristik/sravnitelnaya-harakteristika->

- treh-cms-drupal-joomla-i-wordpress – Назва з екрану.17. Евгений Жданов. Правильный выбор цвета. – К.: Факт, 2004 – 470 с.
14. Kohana [електронний ресурс] // Режим доступу: <http://kohanaforum.org.ua/> – Назва з екрану.
15. Романюк О.Н. Веб-дизайн і комп'ютерна графіка. Навчальний посібник / О.Н. Романюк, Д.І. Кательніков, О.П. Косовець. – В.: ВНТУ, 2007. - 147 с.
16. Буда А. Г. Методичні вказівки до оформлення курсових проектів (робіт) у Вінницькому національному технічному університеті /Уклад. Г.Л. Лисенко, А.Г. Буда, Р.Р. Обертюх. В.: ВНТУ, 2006. – 60 с.

ДОДАТКИ

ДОДАТОК А. ВИХІДНИЙ ТЕКСТ МОДУЛЯ УПРАВЛІННЯ НОВИНАМИ

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Controller_Admin_News extends Controller_Admin {
```

```
    public function before() {
```

```
        parent::before();
```

```
        $this->settings = Kohana::$config->load('news');
```

```
        $this->path = Request::initial()->url();
```

```
        $this->controller = Request::current()->controller();
```

```
    }
```

```
    public function action_index() {
```

```
        $this->template->scripts[] = 'adminmedia/js/bootstrap-typeahead.js';
```

```
        $this->template->scripts[] = 'adminmedia/js/bootstrap-tooltip.js';
```

```
        $count = ORM::factory('new')->count_all();
```

```
        $pagination = Pagination::factory( array(
```

```
            'total_items' => $count,
```

```
            'items_per_page' => $this->settings->pagination_back,
```

```
        ))
```

```
        ->route_params( array(
```

```
            'controller' => $this->controller,
```

```
            'action' => Request::current()->action(),
```

```
        ));
```

```
        $news = ORM::factory('new')
```

```
            ->limit($pagination->items_per_page)
```

```
            ->offset($pagination->offset)
```

```
            ->order_by('move', 'DESC')
```

```
            ->find_all()->as_array();
```

```
        $news = array_values($news);;
```

```
        $menu = Widget::load('menusub', array('param'=>'admin'));
```

```
        $search = Widget::load('search', array('param'=>'admin', 'contr'=>$this->controller));
```

```
        $content = View::factory('admin/news/v_news_index')
```

```

->bind('news', $news)
->bind('menu', $menu)
->bind('search', $search)
->bind('controller', $this->controller)
->bind('path', $this->path)
->bind('lang_id', $this->lang_id)
->bind('settings', $this->settings)
->bind('pagination', $pagination);

// ВЫВОД В ШАБЛОН
$this->template->block_center = array($content);

// ВЫВОД В ШАБЛОН
// $this->template->block_center = array($content);
}

public function action_edit() {
    $this->template->scripts[] = 'adminmedia/js/tiny_mce/tiny_mce.js';
    $this->template->scripts[] = 'adminmedia/js/tiny_mce/my_init.js';
    $this->template->scripts[] = 'adminmedia/js/jquery.MultiFile.pack.js';
    $this->template->scripts[] = 'adminmedia/js/upload.js';
    $this->template->scripts[] = 'http://code.jquery.com/ui/1.10.0/jquery-ui.js';
    $this->template->styles[] = 'http://code.jquery.com/ui/1.10.0/themes/base/jquery-ui.css';
    $id = (int) $this->request->param('id');
    $news = ORM::factory('new', $id);
    $langs = ORM::factory('syslang')->where('visible', '=', 1)->find_all();

    if(!$news->loaded()){
        $this->request->redirect('admin/news');
    }
    $data = $news->as_array();
    if ($this->settings->is_categories){
        $categories = ORM::factory('newcategory')->find_all()->as_array();
    }
}

```

```

$cats = array();
foreach ($categories as $cat){
    $cat_lang = $cat->texts->where('syslangs_id', '=', $this->lang_id)->find();
    $cats[$cat->id] = $cat_lang->title;
}
$data['cat'] = $news->newscategories->find_all()->as_array();
}

$dani = array();
$img = array();
foreach($langs as $lang){
    $value = $news->newstext->where('syslangs_id', '=', $lang->id)->find()->as_array();
    $dani[$lang->cod][$lang->cod.'_title'] = $value['title'];
    $dani[$lang->cod][$lang->cod.'_m_title'] = $value['m_title'];
    $dani[$lang->cod][$lang->cod.'_m_description'] = $value['m_description'];
    $dani[$lang->cod][$lang->cod.'_m_keywords'] = $value['m_keywords'];
    $dani[$lang->cod][$lang->cod.'_text'] = $value['text'];
    $dani[$lang->cod][$lang->cod.'_top_text'] = $value['top_text'];
    $dani[$lang->cod][$lang->cod.'_author'] = $value['author'];
    $dani[$lang->cod][$lang->cod.'_description'] = $value['description'];
    $dani[$lang->cod][$lang->cod.'_main'] = $value['main'];
    $dani[$lang->cod][$lang->cod.'_img'] = $value['img'];
    $img[$lang->cod] = $value['img'];
}

// Редактирование
if (isset($_POST['submit'])) {
    // if (is_writable('/var/www/ratiolegal/data/mod-tmp/')) echo 'write'; else echo
    'not_write';
    // print_r($_FILES);die();
    foreach($langs as $lang){
        $dani[$lang->cod] = Arr::extract($_POST, array($lang->cod.'_title', $lang-
        >cod.'_m_title',
            $lang->cod.'_m_description', $lang->cod.'_m_keywords', $lang->cod.'_text',
            $lang->cod.'_description',

```

```

        $lang->cod.'_author', $lang->cod.'_main', $lang->cod.'_top_text'));
        if(!$dani[$lang->cod][$lang->cod.'_title']) $dani[$lang->cod][$lang->cod.'_title'] =
Date('Y-m-d H:i:s');
    }
    //      print_r($dani);die();
    $data = Arr::extract($_POST, array('alias', 'date_start', 'date_end', 'visible', 'tag', 'gis',
'cat', 'top', 'court', 'execute'));
    $data['date_end'] = date('Y-m-d H:i:s', strtotime($data['date_end']));
    $data['date_start'] = date('Y-m-d H:i:s', strtotime($data['date_start']));
    if(!$data['alias']){
        $useful = new Model_Useful();
        //беремо головну мову сайту, витягуємо введену користувачем назву сторінки
на цій мові, і переводимо її в трансліт
        $main_lang = $useful->GetMainLanguage($langs);
        $alias = Arr::extract($_POST, array($main_lang.'_title', 'ru_title'));
        $data['alias'] = $useful->translit($alias[$main_lang.'_title']);
    }

    try {
        //работа с изображениями
        if (!empty($_FILES['images']['name'][0])){
            foreach($_FILES['images']['tmp_name'] as $image){
                $filename = $this->upload_img($image, $id);
                $data['img'] = $filename;
            }
            if (file_exists('media/uploads/news/'.$id.'/'.$news->img))
unlink('media/uploads/news/'.$id.'/'.$news->img);
                if (file_exists('media/uploads/news/prew/'.$id.'/'.$news->img))
unlink('media/uploads/news/prew/'.$id.'/'.$news->img);
                if (file_exists('media/uploads/news/small/'.$id.'/'.$news->img))
unlink('media/uploads/news/small/'.$id.'/'.$news->img);
                //      if (file_exists('media/uploads/page/prew/'.$pages->img))
unlink('media/uploads/page/prew/'.$pages->img);

```

```

//          if (file_exists('media/uploads/page/small/'.$pages->img))
unlink('media/uploads/page/small/'.$pages->img);
    }

    if ($data['top'] && !empty($_FILES['top_img']['name'])){
        $data['top_img'] = $this->upload_slider($_FILES['top_img']['tmp_name'], $id);
    }
    $news->values($data);
    $news->save();

    if ($this->settings->is_categories){
        if ($data['cat']){
            $news->remove('newscategories');
            $news->add('newscategories', $data['cat']);
        }
    }
    foreach($slangs as $lang){
        $values = array();
        $values['title'] = $dani[$lang->cod][$lang->cod.'_title'];
        $values['m_title'] = $dani[$lang->cod][$lang->cod.'_m_title'];
        $values['m_description'] = $dani[$lang->cod][$lang->cod.'_m_description'];
        $values['m_keywords'] = $dani[$lang->cod][$lang->cod.'_m_keywords'];
        $values['description'] = $dani[$lang->cod][$lang->cod.'_description'];
        $values['main'] = $dani[$lang->cod][$lang->cod.'_main'];
        $values['text'] = $dani[$lang->cod][$lang->cod.'_text'];
        $values['top_text'] = $dani[$lang->cod][$lang->cod.'_top_text'];
        $values['author'] = $dani[$lang->cod][$lang->cod.'_author'];
        $values['news_id'] = $id;
        $values['syslangs_id'] = $lang->id;
        //работа с изображениями
        if (!empty($_FILES[$lang->cod.'_images']['name'][0])){
            foreach($_FILES[$lang->cod.'_images']['tmp_name'] as $image){
                $filename = $this->upload_img($image, $id);
                $values['img'] = $filename;
            }
        }
    }

```

```

        if (file_exists('media/uploads/news/'.$id.'/'.$img[$lang->cod]))
unlink('media/uploads/news/'.$id.'/'.$img[$lang->cod]);
        if (file_exists('media/uploads/news/prew/'.$id.'/'.$img[$lang->cod]))
unlink('media/uploads/news/prew/'.$id.'/'.$img[$lang->cod]);
        if (file_exists('media/uploads/news/small/'.$id.'/'.$img[$lang->cod]))
unlink('media/uploads/news/small/'.$id.'/'.$img[$lang->cod]);
    }
    $newstexts = $news->newstext->where('syslangs_id', '=', $lang->id)->find();
    if (!$newstexts->loaded()){
        $newstexts = ORM::factory('newtext');
    }
    $newstexts->values($values);
    $newstexts->save();

}

$this->request->redirect('admin/news');
}
catch (ORM_Validation_Exception $e) {
    $errors = $e->errors('validation');
}
}

$content = View::factory('admin/news/v_news_edit')
->bind('id', $id)
->bind('errors', $errors)
->bind('tags', $tags)
->bind('cats', $cats)
->bind('langs', $langs)
->bind('main_lang', $this->langsettings->frontend)
->bind('dani', $dani)
->bind('settings', $this->settings)
->bind('path', $this->path)
->bind('controller', $this->controller)

```

```

->bind('gis', $gis)
->bind('court', $court)
->bind('execute', $execute)
->bind('data', $data);

// Вывод в шаблон
$this->template->page_title .= ' :: Редактировать';
$this->template->block_center = array($content);
}

public function action_add() {
    $this->template->scripts[] = 'adminmedia/js/tiny_mce/tiny_mce.js';
    $this->template->scripts[] = 'adminmedia/js/tiny_mce/my_init.js';
    $this->template->scripts[] = 'adminmedia/js/jquery.MultiFile.pack.js';
    $this->template->scripts[] = 'adminmedia/js/upload.js';
    $this->template->scripts[] = 'http://code.jquery.com/ui/1.10.0/jquery-ui.js';
    $this->template->styles[] = 'http://code.jquery.com/ui/1.10.0/themes/base/jquery-ui.css';
    $langs = ORM::factory('syslang')->where('visible', '=', 1)->find_all();

    if ($this->settings->is_categories){
        $categories = ORM::factory('newcategory')->find_all()->as_array();
        $cats = array();
        foreach ($categories as $cat){
            $cat_lang = $cat->texts->where('syslangs_id', '=', $this->lang_id)->find();
            $cats[$cat->id] = $cat_lang->title;
        }
        // $data['cat'] = $news->newscategories->find_all()->as_array();
    }

    if (isset($_POST['submit'])) {

        $dani = array();
        foreach($langs as $lang){

```

```

        $dani[$lang->cod] = Arr::extract($_POST, array($lang->cod.'_title', $lang->cod.'_m_title', $lang->cod.'_m_description',
            $lang->cod.'_m_keywords', $lang->cod.'_text', $lang->cod.'_description',
            $lang->cod.'_author', $lang->cod.'_main', $lang->cod.'_top_text'));
        if(!$dani[$lang->cod][$lang->cod.'_title']) $dani[$lang->cod][$lang->cod.'_title'] =
Date('Y-m-d H:i:s');
    }
    //      print_r($dani);die();
    $data = Arr::extract($_POST, array('alias', 'date_start', 'date_end', 'visible', 'tag', 'gis',
'cat', 'top', 'execute', 'court'));
    $data['date_end'] = date('Y-m-d H:i:s', strtotime($data['date_end']));
    if ($data['date_start']){
        //      $date_time = strtotime(mktime(date('H', time()), date('i', time()), 0, date('m',
strtotime($data['date_start'])), date('d', strtotime($data['date_start'])), date('Y',
strtotime($data['date_start']))));
        $date_time = date('Y-m-d H:i:s', mktime(date('H', time()), date('i', time()), 0, date('m',
strtotime($data['date_start'])), date('d', strtotime($data['date_start'])), date('Y',
strtotime($data['date_start']))));
        //      echo $date_time;die();
        $data['date_start'] = $date_time;//date('Y-m-d H:i:s', strtotime($data['date_start']));
    }
    else $data['date_start'] = date('Y-m-d H:i:s', time());
    if(!$data['alias']){
        $alias = Arr::extract($_POST, array($this->langsettings->translit_cod.'_title',
'ru_title');
        $data['alias'] = $this->usefull->translit($alias[$this->langsettings->translit_cod.'_title']);
    }

    $news = ORM::factory('new');
    try {
        //работа с изображениями
        $news->values($data);
        $news->save();
    }

```

```

        if (!empty($_FILES['images']['name'][0])){
            foreach($_FILES['images']['tmp_name'] as $image){
                $filename = $this->_upload_img($image, $news->pk());
                //                $data['img'] = $filename;
            }
        }
        if ($data['top'] && !empty($_FILES['top_img']['name'])){
            $top_img = $this->_upload_slider($_FILES['top_img']['tmp_name'], $news-
>pk());
        }
        $move = ORM::factory('new', $news->pk());
        $move->move = $move->id;
        if (!empty($_FILES['images']['name'][0])) $move->img = $filename;
        if ($data['top'] && !empty($_FILES['top_img']['name'])) $move->top_img =
        $top_img;

        $move->save();

        foreach($langs as $lang){
            $newstexts = ORM::factory('newtext');
            $values = array();
            $values['title'] = $dani[$lang->cod][$lang->cod.'_title'];
            $values['m_title'] = $dani[$lang->cod][$lang->cod.'_m_title'];
            $values['m_description'] = $dani[$lang->cod][$lang->cod.'_m_description'];
            $values['m_keywords'] = $dani[$lang->cod][$lang->cod.'_m_keywords'];
            $values['description'] = $dani[$lang->cod][$lang->cod.'_description'];
            $values['text'] = $dani[$lang->cod][$lang->cod.'_text'];
            $values['top_text'] = $dani[$lang->cod][$lang->cod.'_top_text'];
            $values['news_id'] = $news->pk();
            $values['syslangs_id'] = $lang->id;
            //работа с изображениями
            if (!empty($_FILES[$lang->cod.'_images']['name'][0])){
                foreach($_FILES[$lang->cod.'_images']['tmp_name'] as $image){
                    $filename = $this->_upload_img($image, $news->pk());

```

```

        $values['img'] = $filename;
    }
}
$newstexts->values($values);
$newstexts->save();
}
$this->request->redirect('admin/news');
}
catch (ORM_Validation_Exception $e) {
    $errors = $e->errors('validation');
}
}
$content = View::factory('admin/news/v_news_add')
    ->bind('errors', $errors)
    ->bind('data', $data)
    ->bind('dani', $dani)
    ->bind('cats', $cats)
    ->bind('settings', $this->settings)
    ->bind('main_lang', $this->langsettings->frontend)
    ->bind('langs', $langs);

// ВЫВОД В ШАБЛОН
$this->template->page_title .= '::__'.__('Add');
$this->template->block_center = array($content);
}

public function action_delete() {
    $id = (int) $this->request->param('id');
    $path = $this->request->param('path');
    if(!$id) {
        $this->request->redirect('admin/news');
    }

    $page = ORM::factory('new', $id);
    if(!$page->loaded()) $this->request->redirect($path);

```

```

        $this->usefull->del_dir_with_images('media/uploads/news/'.$id);
//    $this->_push_history($page, 2);
    $page->delete();
    $this->request->redirect($path);
}

public function _upload_img($file, $id, $ext = NULL, $directory = NULL){
    if($directory == NULL)
    {
        $directory = 'media/uploads/news/'.$id;
    }
    if($ext == NULL)
    {
        $ext = 'jpg';
    }
    if (!file_exists($directory)) mkdir ($directory);
    if (!file_exists($directory.'/small/')) mkdir ($directory.'/small/');
    if (!file_exists($directory.'/prew/')) mkdir ($directory.'/prew/');
    $symbols = '1234567890qwertyuiopasdfghjklzxcvbnm';
    $filename = substr(str_shuffle($symbols), 0, 16);
    //задаються розміри картінки, зокрема її ширина.
    $im = Image::factory($file);
    $min_res = $im->height < $im->width ? $im->height : $im->width;
//    if ($im->height < $im->width)
//        $im->resize(NULL, 161)->crop($this->settings->img_prew, 161);
//    else
        $im->crop($min_res, $min_res)->resize($this->settings->img_prew)->crop($this->settings->img_prew, 161);
        $im->save("$directory/prew/$filename.$ext");

    $im = Image::factory($file);
//    if ($im->height < $im->width)
//        $im->resize(*$this->settings->img_small*/NULL, 112)->crop($this->settings->img_small, 112);
//    else

```

```

$im->crop($min_res, $min_res)->resize($this->settings->img_small)->crop($this-
>settings->img_small, 112);

```

```

$im->save("$directory/small/$filename.$ext");

```

```

$im = Image::factory($file);

```

```

if ($im->width > $this->settings->img_width)

```

```

    $im->resize($this->settings->img_width);

```

```

$im->save("$directory/$filename.$ext");

```

```

return "$filename.$ext";

```

```

}

```

```

public function _upload_slider($file, $id, $ext = NULL, $directory = NULL){

```

```

    if($directory == NULL)

```

```

    {

```

```

        $directory = 'media/uploads/news/'.$id;

```

```

    }

```

```

    if($ext == NULL)

```

```

    {

```

```

        $ext = 'jpg';

```

```

    }

```

```

    if (!file_exists($directory.'/slider/')) mkdir ($directory.'/slider/');

```

```

    $symbols = '1234567890qwertyuiopasdfghjklzxcvbnm';

```

```

    $filename = substr(str_shuffle($symbols), 0, 16);

```

```

    //задаються розміри картінки, зокрема її ширина.

```

```

    $im = Image::factory($file);

```

```

    $im->resize(637)->crop(637, 240);

```

```

    $im->save("$directory/slider/$filename.$ext");

```

```

    return "$filename.$ext";

```

```

}

```

```

public function action_del(){

```

```

    $data = Arr::extract($_POST, array('del_all', 'path'));

```

```

    foreach($data['del_all'] as $key=>$value){

```

```

        $pages = ORM::factory('new', $value);

```

```

        if(!$pages->loaded()){
            $this->request->redirect('admin/'.$this->controller);
        }
//        $this->_delete_page($pages);
//        $this->_push_history($pages, 2);
        $this->usefull->del_dir_with_images('media/uploads/news/'.$value);
        $pages->delete();
    }
    $this->request->redirect($data['path']);
}

```

```

public function action_move(){
    $id = $this->request->param('id');
    $path = $this->request->param('path');
    $items = explode('-', $id);
    $this->usefull->move_sibling('new', $items[0], $items[1]);
    $this->request->redirect($path);
}

```

```

public function action_visibility(){
    $id = (int) $this->request->param('id');
    $path = $this->request->param('path');
    $this->usefull->visible('new', $id);
    $this->request->redirect($path);
}

```

```

public function action_search(){
    $this->template->scripts[] = 'adminmedia/js/bootstrap-typeahead.js';
    $this->template->scripts[] = 'adminmedia/js/bootstrap-tooltip.js';
    $data = Arr::extract($_GET, array('search'));
    $keys = array();
    if ($data['search']){
        $multis = ORM::factory('newtext')
            ->where('syslangs_id', '=', $this->langsettings->backend)

```

```

        ->and_where('title', 'like', '%'.$data['search'].'%')
        ->find_all();
    foreach($multis as $multy){
        $keys[] = $multy->news_id;
    }
}

$find = count($keys);
$pagination = Pagination::factory( array(
    'total_items' => $find,
    'items_per_page' => 2, // $this->settings->pagination_back,
    'view' => 'pagination/floating',
))
->route_params( array(
    'controller' => $this->controller,
    'action' => Request::current()->action(),
));
/* @var $count type - count of pages*/
if ($find){
    $multis = ORM::factory('new')
        ->where('id', 'in', $keys)
        ->limit($pagination->items_per_page)
        ->offset($pagination->offset)
        ->order_by('move', 'DESC')
        ->find_all()->as_array();
    $news = array_values($multis);
}

$count = ORM::factory('new')->count_all();
$menu = Widget::load('menusub', array('param'=>'admin'));
$search = Widget::load('search', array('param'=>'admin', 'contr'=>$this->controller));
$content = View::factory('admin/news/v_news_index')
    ->bind('news', $news)
    ->bind('find', $find)
    ->bind('count', $count)

```

```
->bind('menu', $menu)
->bind('search', $search)
->bind('controller', $this->controller)
->bind('path', $this->path)
->bind('lang_id', $this->lang_id)
->bind('settings', $this->settings)
->bind('pagination', $pagination);

// Вывод в шаблон
// $this->template->title = 'Статьи::Поиск';
$this->template->page_title .= '::Поиск';
$this->template->block_center = array($content);
}
}
```

ДОДАТОК Б. ЛІСТИНГ ГОЛОВНОГО КОНТРОЛЕРА САЙТУ

```

<?php defined('SYSPATH') or die('No direct script access. ');
/*
 * Базовый класс главной страницы
 */
class Controller_Index extends Controller_Base {
    public $template = 'index/v_base';    // Базовый шаблон

    public function before() {
        parent::before();
        if ($this->langsettings->location){
            if (isset($_SERVER['HTTP_ACCEPT_LANGUAGE'])){
                $ACCEPT_LANGUAGE = strtolower(substr($_SERVER['HTTP_ACCEPT_LANGUAGE'], 3, 2));
                if ($ACCEPT_LANGUAGE == 'us') $ACCEPT_LANGUAGE = 'en';
                if ($ACCEPT_LANGUAGE && !Cookie::get('lang_front')) $this->lang = $ACCEPT_LANGUAGE;
            }
            if (Cookie::get('lang_front')) $this->lang = Cookie::get('lang_front');
        }
        else
            $this->lang = Cookie::get('lang_front') ? Cookie::get('lang_front') : $this->langsettings->front_cod; // присвоюємо буквенний код мови фронтенду

        I18n::lang($this->lang); //встановлюємо мову фронтенду.
        // echo i18n::lang();
        // echo $this->lang;
        $front = ORM::factory('syslang')->where('cod', '=', $this->lang)->find();
        $this->lang_id = $front->id;//$this->langsettings->frontend;
        // echo $this->lang_id;
        if($this->settings->site_block == TRUE){
            if($this->settings->site_block_404 == TRUE){
                throw new HTTP_Exception_404('Запрашиваемая категория не найдена');
                return;
            }
        }
    }
}

```

```

    }
    $response = new Response;
    $response->status('404');
    die('site is blocked');
}

$stopmenu = Widget::load('topmenu');
$footermenu = Widget::load('footermenu');
$login = Widget::load('login');
$registration = Widget::load('registration');
$langchange = Widget::load('lang', array('param'=>'anchor'));
$indexSearch = Widget::load('indexSearch');
$lang_front = $this->lang;

$this->template->scripts = array('media/js/jquery.js',
                                'media/js/less.js',
                                'media/js/script.js',
//                                'adminmedia/js/fancybox/jquery.fancybox-1.3.4.pack.js',
//                                'adminmedia/js/fancybox/jquery.easing-1.3.pack.js',
//                                'adminmedia/js/fancybox/jquery.mousewheel-3.0.4.pack.js',
                                'media/css/modal/js/bootstrap.min.js',
                                'media/js/dotdotdot.js',
//                                'adminmedia/js/bootstrap-typeahead.js',
                                );
$this->template->styles = array('media/css/modal/css/bootstrap.min.css',
                                //'media/css/bootstrap-responsive.min.css',
                                //'media/css/style.css',
                                'adminmedia/js/fancybox/jquery.fancybox-1.3.4.css'
                                );
$this->template->less = array('media/less/newstyle.less');
// Вывод в шаблон
$this->template->footermenu = $footermenu;
$this->template->slider = NULL;//$slider;
$this->template->ask = NULL;//$slider;

```

```

$this->template->auth = $this->auth;
$this->template->user = $this->user;
$this->template->login = $login;
$this->template->registration = $registration;
$this->template->lang = $langchange;
$this->template->indexSearch = $indexSearch;
$this->template->controller = $this->controller;
$this->template->top_menu = $topmenu;
//$this->template->footer_menu = $footermenu;
$this->template->lang_front = $lang_front;
$this->template->slider = NULL;

//    $this->template->block_left = array();
    $this->template->block_right = array();
    $this->template->block_center = array();

    $page = ORM::factory('page')->where('alias', '=', $this->controller)->find();
    $page = $page->pagetext->where('syslangs_id', '=', $this->lang_id)->find()->as_array();
    // Вывод в шаблон
    $this->template->title = $page['m_title'];
    $this->template->keywords = $page['m_keywords'];
    $this->template->description = $page['m_description'];
//    echo URL::site(NULL, TRUE);

}

}

```

ДОДАТОК В. ЛІСТИНГ ГОЛОВНОГО ШАБЛОНУ

```

<!DOCTYPE HTML>
<html lang="<?=$lang_front;?>">
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta charset="utf-8">
    <meta name="SKYPE_TOOLBAR"
content="SKYPE_TOOLBAR_PARSER_COMPATIBLE" />
    <meta content="telephone=no" name="format-detection">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <?foreach ($styles as $file_style):?>
        <?=html::style($file_style)?>
    <?endforeach?>

    <?foreach ($less as $file_less):?>
        <?=html::less($file_less)?>
    <?endforeach?>

    <?foreach ($scripts as $file_script):?>
        <?=html::script($file_script)?>
    <?endforeach?>

    <title><?=$site_name?> | <?=$title?></title>
    <meta name="description" content="<?=$description;?>" />
    <meta name="keywords" content="<?=$keywords;?>" />
    <link rel="icon" type="image/vnd.microsoft.icon" href="/media/img/favicon.ico" />
    <link rel="SHORTCUT ICON" href="/media/img/favicon.ico" />
    <link rel="alternate" type="application/rss+xml" title="<?=$site_name;?>"
href="<?=URL::base(TRUE, TRUE);?>feed/" />
    <meta name="author" content="Maxim Rudniy, http://maxim.webro.com.ua">

    <!--[if gte IE 9]>
        <style type="text/css">

```

```

.gradient {
    filter: none;
}
</style>
<![endif]-->
<!--[if lt IE 9]>
    <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->

</head>
<body>
    <!--[if lt IE 8]>
        <div style=" margin:10px auto 0px auto; padding:20px; background:#DDDDDD;
border:1px solid gray; width:980px; font-size:14px;">
            Уважаемый Пользователь!</br>
            Вы используете <span class="red">устаревший WEB-браузер</span>.</br>
            Предлагаем Вам установить и использовать последние версии WEB-браузеров,
            например:<br/>
            <ul>
                <li>Google Chrome <a
href="https://www.google.com/chrome">https://www.google.com/chrome</a></li>
                <li>Mozilla Firefox <a
href="http://www.mozilla.org/ru/firefox/new/">http://www.mozilla.org/ru/firefox/new/</a></li>
                <li>Opera <a
href="http://www.opera.com/download/">http://www.opera.com/download/</a></li>
            </ul>
            Последние версии WEB-браузеров доступны для установки на сайтах разработчиков
            и содержат улучшенные свойства безопасности, повышенную скорость работы, меньшее
            количество ошибок. Эти простые действия помогут Вам максимально использовать
            функциональность сайта, избежать ошибок в работе, повысить уровень безопасности.
        </div>
    <![endif]-->

    <div class="wrapper">

```

```

<header>
  <div class="top">
    <div class="wrap">
      <div class="support left">
        <span><i class="icon-phone"></i><?=__('mob (067) 655-55-55');?></span>
        <span><i class="icon-mail"></i><?=__('kaf_mail');?></span>
        <span><i class="icon-time"></i><?=__('rozpisanie');?></span>
      </div>
      <?=$lang;?>
      <span class="enter right">
        <?if(!$auth->logged_in('login')):?>
          <?=HTML::anchor('register', __('Registration'), array('id'=>'registre_anchor'));>
          <span></span>
          <?=HTML::anchor('#myModal', __('Enter'), array('id'=>'enter_anchor',
'class'=>'modal_enter', 'data-toggle'=>'modal'));>
          <!--<span class="modal_enter" id="enter_anchor"><?=__('Enter');?></span>-->
          <?else:?>
            <?=$basic = $user->basic->find();?>
            <?=HTML::anchor('logout', __('Exit'), array('class'=>'logout',
'id'=>'registre_anchor'));>
            <span></span>
            <?if ($basic->first_name && $basic->first_name) echo HTML::anchor('account',
__, __('Welcome').', <b>'.$basic->first_name.' '.$basic->last_name.'</b>', array('class'=>'account',
'id'=>'enter_anchor'));
            else echo HTML::anchor('account', __('Welcome').', <b>'.$user-
>username.'</b>', array('class'=>'account', 'id'=>'enter_anchor'));>
            <?endif;?>
          </span>
        </div>
        <div class="clear"></div>
      </div>
      <div class="wrap">
        <div class="row-fluid">
          <div class="logo">
            <a href="/" class="kaf"><?=__('kafedra');?></a>

```

```

        <div class="pz"><a href="/"><?=__('software');?></a></div>
        <div class="soft"><a href="/"><?=__('software_second');?></a></div>
    </div>
</div>
</div><!--.wrap-->

    <div class="navbar">
        <div class="wrap">
            <?=$top_menu;?>
        </div><!--.wrap-->
    </div>

</header>
    <?=$slider;?>
    <div class="content wrap">
<!-- Центральный блок-->
    <div class="vmist">
        <!-- Левый блок-->
        <? if (isset($block_left)):?>
            <aside>
                <? foreach ($block_left as $lblock):?>
                    <?=$lblock?>
                <?endforeach?>
            </aside>
        <?endif?>
        <!-- /Левый блок-->
        <article <?if (isset($block_left)) echo 'class="cut_article"';?>>
        <? if (isset($block_center)):?>

            <!--<h2 class="page_title"><?=$page_title?></h2>-->
            <? foreach ($block_center as $cblock):?>
                <?=$cblock?>
            <?endforeach?>

        <?endif?>

```

```

</article>
<div class="clear"></div>
</div>
    <!-- /Центральный блок-->
</div>
<footer>
    <div class="wrap soc_seti">
        <div class="left">
            <h2><?=__('We are in social networks');?></h2>
            <h3><?=__('Find out about our department in social networks');?></h3>
        </div>
        <div class="left links">
            <a href="http://vk.com/fcsukraine" class="soc vk" target="_blank"></a>
            <a href="https://www.facebook.com/FCSUkraine" class="soc fb"
target="_blank"></a>
            <a href="https://twitter.com/FCSU_official" class="soc tw" target="_blank"></a>
            <a href="/feed" class="soc rss" target="_blank"></a>
        </div>
        <div class="left">
            <h2><?=__('search_top');?></h2>
            <h3><?=__('all_search');?></h3>
        </div>
        <div class="left last">
            <?=$indexSearch;?>
        </div>

    <div class="clear"></div>

</div><!--.wrap-->
<div class="footer_menu">
    <div class="wrap">
        <?=$footermenu;?>
        <div class='right copyright'>
            &copy;<?=$Date('Y', time());?> <a
href="http://webro.com.ua"><?=__('copyright');?></a> webro.com.ua

```

```

        </div>
        <div class='clear'></div>
    </div>
</div>
</footer>
</div>
<!-- Modal -->
<div id="myModal" class="modal hide fade" tabindex="-1" role="dialog" aria-
labelledby="myModalLabel" aria-hidden="true">

    <div class="modal-body">
        <div class='ent'><?=__('Enter');?></div>
        <button class="btn" data-dismiss="modal" aria-hidden="true"></button>
        <div class="rez_submit"></div>
        <?=Form::open('login');?>
        <?=Form::input('username', "", array('placeholder'=>__('Login'), 'id'=>'mod_login'));?>
        <?=Form::password('password', "", array('placeholder'=>__('password1'),
'id'=>'mod_pass'));?>
        <div class='links'>
            <?=HTML::anchor('register', __('Registration'));?>
            <?=HTML::anchor('restorpass', __('Forgot your password').'?');?>
        </div>
        <?=Form::submit('submit', __('voiti'), array('id'=>'mod_sub'));?>
        <?=Form::close();?>
        <div class='clear'></div>
    </div>

</div>
<script>
    $('modal_enter').modal(options);
    // $('modal_enter').click(function(e){e.preventDefault();});
</script>
</body>
</html>

```

ДОДАТОК Г

ІЛЮСТРАТИВНИЙ МАТЕРІАЛ ДО ЗАХОСТУ БАКАЛАВРСЬКОЇ
ДИПЛОМНОЇ РОБОТИ

Завідувач кафедри ПЗ, д.т.н., професор _____ А.М. Петух

Науковий керівник, к.т.н., доцент _____ В.В. Войтко

Рецензент, д.т.н., професор _____ Роїк О. М.

Виконавець, студент групи ІПІ-12мс _____ Рудний М. М.