

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ім. Ігоря Сікорського**

Навчально-науковий комплекс «Інститут прикладного системного аналізу»
(повна назва інституту/факультету)

Кафедра Системного проектування
(повна назва кафедри)

«До захисту допущено»
Завідувач кафедри

_____ А.І.Петренко
(підпис) (ініціали, прізвище)

“ ____ ” _____ 2017 р.

Дипломна робота

на здобуття ступеня бакалавра

з напрямку підготовки _____ 6.050101 Комп'ютерні науки
(код і назва)

на тему: Розробка та впровадження бази методичних посібників на фреймворку

Виконав: студент 4 курсу, групи ДА-31
(шифр групи)

_____ Молоденков Вадим Петрович _____
(прізвище, ім'я, по батькові) (підпис)

Керівник _____ доцент к.т.н. Цурін О.П. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант економічний _____ доцент к.е.н. Рощина Н.В. _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Нормоконтроль _____ старший викладач Бритов О.А. _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Київ – 2017 року

**Національний технічний університет України
«Київський політехнічний інститут»
ім. Ігоря Сікорського**

Інститут (факультет) ННК «Інститут прикладного системного аналізу
(повна назва)

Кафедра Системного проектування
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки 6.050101 Комп'ютерні науки
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ А.І.Петренко
(підпис) (ініціали, прізвище)

« ___ » _____ 2017 р.

ЗАВДАННЯ

на дипломну роботу студенту

Молоденкову Вадиму Петровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та впровадження бази методичних посібників на фреймворку

керівник роботи Цурін Олег Пилипович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « ___ » _____ 20__ р. № _____

2. Термін подання студентом роботи 06.06.2017

3. Вихідні дані до роботи

1. Мова програмування - PHP, WEB-фреймворк – Yii
2. База даних MySQL
3. Форма реалізації – web-сайт на базі MVC архітектури.

4. Зміст роботи

1. Провести розгляд і аналіз сучасних WEB-фреймворків
2. Провести опис вимог до сайту “Бази методичних посібників”, зробити вибір фреймворку для реалізації сайту, описати принципи роботи з обраним фреймворком.
3. Розробити сайт “База методичних посібників” за допомогою обраного фреймворку.
4. Провести функціонально-вартісний аналіз програмного продукту.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

1. UML діаграма послідовностей архітектури MVC
2. Порівняльна характеристика сучасних WEB-фреймворків
3. Інструкція по встановленню WEB-фреймворку

6. Консультанти розділів роботи*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Рощина Н.В., доцент.		

* Консультантом не може бути зазначено керівника дипломної роботи.

7. Дата видачі завдання 01.02.2017

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка

Студент

(підпис)

(ініціали, прізвище)

Керівник роботи

(підпис)

(ініціали, прізвище)

АНОТАЦІЯ

бакалаврської дипломної роботи Молоденкова Вадима Петровича на тему: «Розробка та впровадження бази методичних посібників на фреймворку».

Дана дипломна робота присвячена вивченню можливостей сучасних фреймворків у створенні сайтів. В рамках дипломної роботи був проведений аналіз популярних фреймворків. Розглянуто переваги і недоліки використання фреймворків, а також розглянуто процес встановлення фреймворку. Було здійснено вибір фреймворку для реалізації сайту «Методичних посібників».

Фінальним етапом роботи над дипломною роботою стало створення сайту «Методичних посібників»

Загальний обсяг роботи: 103 сторінки, 33 рисунків, 9 таблиць, 20 посилань.

Ключові слова: MVC, фреймворк, Front-End, Back-End, CSS.

ANNOTATION

to the bachelor thesis of by Molodenkov Vadym On a theme “ Development and implementation of the database methodical books on framework”.

This course is devoted to studying the possibilities of modern frameworks to create web-sites. Within the thesis analyzed the popular frameworks. The advantages and disadvantages of using frameworks and considered the installation process framework. Was chosen a framework for creating website methodologies materials.

The final stage of master work was the creation of website "Methodical materials "

The total volume of 103 pages, 33 figures, 9 tables, 20 references.

Keywords: MVC, Framework, Front-End,Back-End,CSS.

Перелік умовних позначень, символів, скорочень і термінів

БД	база даних
СУБД	система управління базами даних
ГОСТ	« государственный стандарт » (міждержавний стандарт СНД)
СНД	Співдружність Незалежних Держав
ПЗ	програмне забезпечення
API	Application Programming Interface
URL	Uniform Resource Locator
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
MVC	Model-View-Controller

ЗМІСТ

ВСТУП	10
1 АНАЛІЗ СУЧАСНИХ WEB-ФРЕЙМВОРКІВ.....	12
1.1 Аналіз Back-end фреймворків.....	15
1.1.1 Переваги використання Back-end фреймворку	25
1.1.2 Недоліки використання Back-end фреймворку	26
1.1.3 Шляхи та області використання	26
1.2 Аналіз Front–End фреймворків	27
1.2.1 Переваги і недоліки застосування Front-end фреймворків	31
1.3 Висновки до розділу 1	33
2 ВИМОГИ ДО САЙТУ “БАЗИ МЕТОДИЧНИХ ПОСІБНИКІВ”. ВИБІР СИСТЕМИ УПРАВЛІННЯ БАЗАМИ ДАНИХ. ВИБІР ФРЕЙМВОРКУ ДЛЯ РЕАЛІЗАЦІЇ. ОГЛЯД РОБОТИ З ОБРАНИМ ФРЕЙМВОРКОМ.....	34
2.1 Вимоги до сайту “Бази методичних посібників”	34
2.2 Аналіз структури формування опису методички	36
2.3 Вибір системи управління базами даних для реалізації сайту “База методичних посібників”	37
2.3.1 Вибір архітектури бази даних	39
2.4 Вибір Back – End фреймворку для реалізації сайту “ База методичних посібників”	44
2.5 Yii як програмний каркас для створення web–додатків	47
2.5.1 Структура Yii	47
2.5.2 Ключові компоненти додатку Yii	48
2.5.3 Реалізація MVC в Yii.....	50
2.5.4 Типовий запит Yii.....	52
2.5.5 Життєвий цикл додатку Yii	54
2.5.6 Процес розробки додатку з використанням Yii	54
2.6 Висновки до розділу 2	56
3 РОЗРОБКА САЙТУ ЗА ДОПОМОГОЮ Yii	57
3.1. Вибір програмного забезпечення	57
3.2 Розробка дизайну сайту.....	57
3.3 Розробка таблиць бази даних.....	59
3.3.1 Створення БД за допомогою PhpMyAdmin	60
3.3.2 Розробка таблиці сутності «Автори».....	61
3.3.3 Розробка таблиці сутності «Публікації»	63

3.3.4 Розробка таблиці зв'язку «Автор_Публікація»	64
3.4 Встановлення Yii на локальний сервер	65
3.5 Налаштування Yii	67
3.5.1 Налаштування з'єднання з базою даних	67
3.5.2 Активація кодогенератора Gii	68
3.5.3 Встановлення доповнення Yiiastrap	69
3.5.4 Налаштування системи доступу на основі ролей	69
3.6 Реалізація архітектури MVC сайту	70
3.6.1 Генерація моделей сайту	71
3.6.2 Генерація контролерів та представлень сайту	72
3.6.3 Зміна дизайну header та footer сайту	74
3.6.4 Зміна дизайну контентної частини сайту	74
3.7 Висновки до розділу 3	77
4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ	78
4.1 Постановка задачі техніко-економічного аналізу	79
4.1.1 Обґрунтування функцій програмного продукту	80
4.1.2 Варіанти реалізації основних функцій	80
4.2 Обґрунтування системи параметрів ПП	84
4.2.1 Опис параметрів	84
4.2.2. Кількісна оцінка параметрів	84
4.2.3 Аналіз експертного оцінювання параметрів	87
4.3 Аналіз рівня якості варіантів реалізації функцій	92
4.4 Економічний аналіз варіантів розробки ПП	94
4.5 Вибір кращого варіанта ПП техніко-економічного рівня	99
4.6 Висновки до розділу 4	99
ВИСНОВКИ	101
ПЕРЕЛІК ПОСИЛАНЬ	103

ВСТУП

Розробка веб-сайтів є перспективним напрямком в розвитку інформаційних технологій, адже глобальна мережа Інтернет охоплює з кожним роком все більше і більше користувачів. В Інтернеті знаходяться мільйони веб-сайтів різного спрямування. Створення та підтримка яких є важкою працею.

Розвиток Інтернету нерозривно пов'язано з проектуванням сайтів. Масова поява сайтів спровокувала проблему їх якості. Популярність створення веб-ресурсів сприяла розробці різних систем і програм, які спрощують процес написання сайту. Також вони допомагають підвищити ефективність роботи, а також дозволяють розробнику сфокусуватися на основною логікою програми. Такі технології, як PHP, Java, Microsoft.Net, MySQL, Oracle, Microsoft SQL Server і розроблені на їх основі фреймворки - це каркаси системи або підсистеми, що можуть включати допоміжні програми, мови сценаріїв - і все, що полегшує розробку й об'єднання різних компонентів [1].

Фреймворки за останній час набрали популярність, і стали базовою платформою для розробки WEB-додатків. Іншими словами можна сказати, що вони забезпечують основну структуру програми. Використання фреймворків, дозволяє економити велику кількість часу, зменшити навантаження на процес розробки, позбавляючи від проблеми повторюваного коду, і швидко створювати додатки.

Проаналізувавши інформацію з мережі Інтернет, можна побачити що існує сотні фреймворків для створення web-додатків. Тому досить складно зробити вибір фреймворку, так як кожен з них має велику кількість привабливих функцій та доповнень. А неправильний вибір фреймворку може стати основною причиною невдачі проекту. Порівнюючи фреймворки слід обрати такі критерії порівняння, що будуть найкраще демонструвати

переваги та недоліки кожного. Також потрібно оцінити які фреймворки є найбільш простими та дозволяють створити веб-сайт швидше за все, а які надають сайту найбільше можливостей й гнучкості [2].

Результатом роботи стане список критеріїв порівняння, які будуть демонструвати які технології краще використовувати в яких ситуаціях, а також опис основних можливостей .

1 АНАЛІЗ СУЧАСНИХ WEB-ФРЕЙМВОРКІВ

Фреймворк - це програмне забезпечення, що полегшує розробку і об'єднання різних компонентів великого програмного проекту. Ця платформа підходить для створення сайтів, бізнес-додатків і веб-сервісів. На відміну від бібліотек, які об'єднують набір підпрограм близької функціональності, фреймворк містить у собі велику кількість різних за призначенням бібліотек. Широкого вжитку також набуло слово «каркас», а деякі автори використовують його в якості основного, взагалі не беручи до уваги англomовний аналог. Можна також говорити про каркасний підхід як про підхід до побудови програм, де будь-яка конфігурація програми будується з двох частин: перша (постійна частина) - каркас, в якому конфігурації не пов'язані між собою і містяться гнізда, в яких розміщується друга частина (змінна) - змінні модулі або точки розширення [3].

Для того щоб відповісти на питання, що має входити до фреймворка, щоб останній розглядався як WEB-технологія, необхідно розглянути сучасні види фреймворків:

- Фреймворки програмної системи;
- Фреймворки додатків;
- Фреймворки концептуальної моделі.

Фреймворк програмної системи - це каркас системи або підсистеми, що може включати допоміжні програми, мови сценаріїв - все, що полегшує розробку й об'єднання різних компонентів. Фреймворк відрізняється від поняття бібліотеки тим, що бібліотека може бути використана в програмному продукті просто як набір підпрограм з близькою функціональністю, не впливаючи на архітектуру програмного продукту і не накладаючи на неї ніяких обмежень. У той час як фреймворк диктує правила побудови архітектури додатку, задаючи на початковому етапі розробки поведінку за

замовчуванням, каркас, який потрібно буде розширювати і змінювати відповідно до зазначених вимог. До цього виду фреймворків відносяться і фреймворки для WEB.

Фреймворк додатку має стандартну структуру. Зі зростанням необхідності у графічних інтерфейсах користувача з'явилася і необхідність у фреймворках додатків. З їх допомогою простіше розробляти засоби для автоматичного створення графічних інтерфейсів. Для створення фреймворку додатків використовують об'єктно-орієнтоване програмування. Перший подібний фреймворк написала компанія Apple для Macintosh. Спочатку він був створений за допомогою Паскаль, а з часом його було переформатовано у C++. Фреймворк концептуальної моделі - це абстрактне поняття даної структури для визначення способів вирішення конкретної проблеми [4].

Також необхідно з'ясувати, що ж саме являють собою WEB фреймворки. WEB фреймворки - це каркас, призначений для створення динамічних веб- сайтів, мережевих додатків, сервісів або ресурсів. Він спрощує розробку і позбавляє від необхідності написання рутинного коду. Багато фреймворків спрощують доступ до баз даних, полегшують розробку інтерфейсу, а також зменшують дублювання коду.

Виділяють п'ять типів веб-фреймворків: Request-based, Component-based, Hybrid, Meta and RIA-based. Request-based - фреймворки, які безпосередньо обробляють вхідні запити. Збереження стану відбувається за рахунок серверних сесій. Приклади: Django, Ruby on Rails, Struts, Grails.

Component-based - фреймворки, які абстрагують обробку запитів всередині стандартних компонентів і самостійно стежать за станом. Дані каркаси мають багато спільного зі стандартними програмними графічними інтерфейсами. Приклади: JSF, Tapestry, Wicket.

Hybrid-based - фреймворки, які комбінують Request-based та Component-based фреймворки, беручи під свій контроль всі дані і логічний потік в заснованій на запиті моделі. Розробники мають повний контроль над URL, формами, параметрами, cookies і pathinfos. Однак замість того, щоб відобразити дії і контролери безпосередньо до запиту, гібридні фреймворки забезпечують об'єктну модель компонентів, яка поводиться тотожно в багатьох різних ситуаціях, таких як окремі сторінки, перервані запити, подібні порталу фрагменти сторінок та інтегровані віджети. Компоненти можуть розподілятися окремо і ефективно інтегруватися в інші проекти. Приклади: RIFE.

Meta-based – фреймворки, що мають ряд базових інтерфейсів для загального обслуговування і основу, яка легко розширюється з метою інтегрування компонентів і служб. Приклад: Keel.

RIA-based (фреймворки для розробки Rich Internet Applications (RIA) – фреймворки, що служать для розробки повноцінних додатків, які запускаються всередині браузера. Приклад: Flex. Найбільш поширеними є Request-based і Component-based веб-фреймворки. Зібравши і проаналізувавши інформацію, мною було виділено такі характерні компоненти web фреймворків:

- Шаблонизатор. Відповідає за незалежність верстки від програмного коду.
- Роутер. Розпізнає URL, за яким відбулося звернення до сервера.
- Модуль доступу до бази даних.
- Модуль кешування. Прискорює завантаження сторінок.
- Модуль безпеки. Аутентифікація і авторизація користувачів.
- Файли конфігурації.

WEB фреймворки також можуть керувати сесіями, вести логи, спрощувати використання технології Ajax та ін [4].

1.1 Аналіз Back-end фреймворків

Для Back-end існують сотні фреймворків, що реалізовані на різних мовах програмування, таких як: Java, Rubi, PHP, Phython, а також на технології створення веб-додатків і веб-сервісів від компанії Майкрософт ASP.NET. Далі буде розглянуто найпопулярніші з них.

Yii – це універсальний фреймворк, який може бути задіяний у всіх типах веб додатків. Завдяки його компонентній структурі та відмінній підтримці кешування, фреймворк особливо підходить для розробки таких великих проектів як портали, форуми, CMS, магазини чи RESTful-додатки.

Можливості Yii :

- Низький поріг входження
- Парадигма Модель-вид-контролер
- Інтерфейси DAO і ActiveRecord для роботи з базами даних (PDO)
- Підтримка інтернаціоналізації
- Кешування сторінок і окремих фрагментів
- Перехоплення і обробка помилок
- Введення і валідація форм
- Аутентифікація і авторизація
- Використання AJAX і інтеграція з jQuery
- Генерація базового PHP-коду для CRUD-операцій (скаффолдинг)
- Підтримка тем оформлення для їх легкої зміни
- Можливість підключення сторонніх бібліотек
- Міграції бази даних

- Автоматичне тестування
- Підтримка REST

Недоліки:

- AR не підтримує AR запити
- Не дуже хороший зв'язок в БД багато-до-багатьох(плагін CAdvancedArBehavior випраляє цей недолік)

CakePHP є фреймворком для PHP, який надає розширену архітектуру для розробки, обслуговування і розгортання web-додатків. Він використовує відомий шаблон проектування MVC, як і в об'єктно-реляційних фреймворках. Основною парадигмою CakePHP є збільшення продуктивності розробки, і, як наслідок, допомога програмісту у вигляді зменшенні обсягу написання коду. Спочатку створювався як клон популярного Ruby on Rails, тому багато ідей були запозичені саме звідти.

Можливості:

- сумісність з PHP4 (до версії 1.3 включно) і PHP5
- диспетчер URL із застосуванням регулярних виразів
- генерація всього коду за схемою бази даних (потрібно дотримуватися стандарту іменування стовпців)
- перевірка форм
- компоненти для авторизації, обмеження доступу (ACL), управління сесіями, cookies, подання деревоподібної інформації (у вигляді Nested Sets)
- хелпери(компоненти) для генерації та заповнення форм, поділу на сторінки (paginate), управління кешем, JavaScript(в тому числі і AJAX)
- механізм інтернаціоналізації
- генерація SQL-запитів, в тому числі для таблиць з відносинами один до багатьох і багато до багатьох, ORM

- Scaffolding і генерація CRUD-сторінок для сутностей, Router:mapResources з Put Delete Get Post.
- Автогенератор коду Bake
- Міграції
- Консольна інтеграція, клас Shell і завдання Task
- Плагіни (як окремі програми), компоненти і поведінки
- Підтримка Simple Test

Недоліки:

- Низька продуктивність
- Слабка документація
- Нестійкість до CSRF-атак
- Немає російськомовної спільноти
- Складний в розумінні ACL
- Строгі угоди з іменування
- Низька швидкість розвитку.

Zend framework – це PHP– фреймворк, що створений і підтримується компанією Zend, співробітники якої є безпосередніми авторами мови PHP. Тому він є послідовником традицій і цінностей PHP – базується на простоті, об'єктно- орієнтованих принципах, дружній ліцензії і ретельно тестованому коді із застосуванням Agile методів.

Можливості:

- Всі компоненти повністю орієнтовані на PHP 5 та E_STRICT - сумісні
- Вбудований генератор коду
- Архітектура «використовуй тільки те, що необхідно» з мінімальними залежностями компонентів

- Використовує легко розширюваний шаблон проектування MVC, підтримує макети і PHP-скрипти подання за замовчуванням
- Підтримує безліч різних баз даних, включаючи MariaDB, MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL, SQLite, and Informix Dynamic Server
- Спеціальні класи для створення, відправлення, отримання email з допомогою mbox, Maildir, POP3 та IMAP4
- Гнучка система кешування з підтримкою безлічі сховищ

Недоліки:

- Досить великий
- Дуже повільний без кешування
- Занадто складна архітектура, без глибокого розуміння шаблонів проектування складний в освоєнні
- Слабке російськомовне співтовариство
- Досить повільна швидкість розвитку
- Остання версія вимагає PHP 5.2
- Відсутній ORM.

Symfony2 – PHP фреймворк, який має велику бібліотеку класів, написаний на PHP 5. Архітектура має корисні компоненти та інструменти, призначені для створення складних веб-додатків. Symfony — це вільний каркас, написаний на PHP5, який використовує патерн Model-View-Controller (MVC). Symfony пропонує швидку розробку і керування веб-додатками, що дозволяє легко вирішувати рутинні завдання веб-програміста. Symfony безкоштовний і доступний під ліцензією MIT.

Перваги :

- Підтримує безліч баз даних (MySQL, PostgreSQL, SQLite, або будь-яка інша PDO-сумісна СУБД)

- Вбудовані класи для роботи з email
- Гнучка система шаблонів у поданні
- Вбудований кодогенератор
- Підтримка французького спонсора Sensio
- Дуже гнучкий
- Висока продуктивність

Недоліки:

- Складний в освоєнні
- Підходить тільки для великих проектів
- Відсутність російської документації
- Остання версія 5.3 вимагає PHP
- Немає вбудованої ORM
- Немає російськомовного співтовариства

Ruby on Rails - об'єктно-орієнтований програмний каркас (фреймворк) для створення веб-додатків, написаний на мові програмування Ruby. Ruby on Rails надає каркас модель-вид-контролер (Model-View-Controller) для веб-додатків, а також забезпечує їхню інтеграцію з веб-сервером і сервером бази даних.

Можливості:

- Лаконічний і простий синтаксис, часто зустрічається вплив Ада, Eiffel і Python
- Може обробляти винятки в стилі Java і Python
- Дозволяє переозначувати оператори, які насправді є методами
- Простий та послідовний синтаксис
- Можливість прямих системних викликів
- Миттєва поява змін під час розробки

- Відсутність стадії компіляції
- Повністю об'єктно-орієнтована мова програмування. Всі дані в Ruby є об'єктами в розумінні Smalltalk. Єдиний виняток — керівні конструкції, які в Ruby на відміну від Smalltalk не є об'єктами
- Містить автоматичний прибиральник сміття. Він працює для всіх об'єктів Ruby, в тому числі і для зовнішніх бібліотек
- Створення розширень для Ruby на C дуже просте завдяки збору сміття, та нескладного і зручного API
- Підтримує цикли з повною прив'язкою до змінних
- Підтримує блок коду (код взятий в { ... } або do ... end). Блоки можуть використовуватись в методах або перетворюватись в цикли
- Змінні динамічно типізовані
- Безпосередньо в мові Ruby реалізовано багато шаблонів програмування. Так, наприклад, «одинак» (англ. singleton) може бути реалізований додаванням потрібних методів одному конкретному об'єктові.
- Може динамічно завантажувати розширення, якщо це дозволяє операційна система
- Перенесена на багато платформ. Мова розроблялася на GNU/Linux, але працює на багатьох версіях Unix, DOS, Microsoft Windows (частково, Win32), Mac OS, BeOS, OS/2 і т.д.

Недоліки:

- Низька швидкість роботи
- Проекти на ньому досить громіздкі
- Складний debug коду
- Філософія розвитку багато в чому залежить від "магії" і "припущень"
- Погана російськомовна документація.

Apache Struts — веб-фреймворк з відкритим кодом для розробки Java EE веб-застосунків. Використовує і розширює Java Servlet API, надаючи архітектуру MVC (Модель-Вид-Контролер).

Можливості:

- Простий POJO-based дії
- Простота тестування коду
- Безпека потоку передачі даних
- Підтримка AJAX
- Клієнтські технології AJAX валідації
- Підтримка шаблонів
- Підтримка різних типів результатів
- Легко розширюється за допомогою плагінів
- Підтримка плагінів фреймворку Spring (впровадження залежностей)
- JFreechart плагін (діаграми)
- jQuery плагін (підтримка AJAX, UI віджетів, динамічні таблиці, діаграми)
- Rome плагін

Недоліки:

- Погана російськомовна документація
- Мале російськомовне співтовариство
- Складність вивчення
- Складність оптимізації додатків
- Складність розуміння коду іншими розробниками

ASP.Net MVC - фреймворк для створення веб-додатків, який реалізує шаблон Model-view-controller. Даний фреймворк розроблений компанією Microsoft.

Фреймворк ASP.NET MVC надає наступні можливості:

- Поділ завдань програми (логіка введення, бізнес-логіка і логіка інтерфейсу), широкі можливості тестування і розробки на основі тестування. Всі основні контракти платформи MVC засновані на інтерфейсі і підлягають тестуванню за допомогою макетів об'єкта, які імітують поведінку реальних об'єктів програми. Додаток можна піддавати модульному тестуванню без запуску контролерів у процесі ASP.NET, що прискорює тестування і робить його більш гнучким. Для тестування можливе використання будь-якої платформи, сумісної з .NET Framework.
- Платформа має можливість розширення і доповнення . Компоненти платформи ASP.NET MVC можна легко замінити або налаштувати. Розробник може підключати власний механізм уявлень, політику маршрутизації URL-адрес, серіалізацію параметрів методів, дій та інші компоненти. Платформа ASP.NET MVC також підтримує використання моделей контейнера впровадження залежності (DI) та інверсії елемента керування (IOC). Модель впровадження залежності дозволяє впроваджувати об'єкти в клас, а не очікувати створення об'єкта самим класом. Модель інверсії елемента управління вказує на те, що якщо один об'єкт вимагає інший об'єкт, то перші об'єкти мають отримати другий об'єкт із зовнішнього джерела (наприклад, з файлу конфігурації), що значно полегшує тестування.
- Розширена підтримка маршрутизації ASP.NET. Цей потужний компонент зіставлення URL-адрес дозволяє створювати додатки із зрозумілими URL-адресами, які можна використовувати в пошуку. URL-адреси не повинні містити розширення імен файлів і призначені для підтримки шаблонів іменування URL-адрес, що забезпечують адресацію, оптимізовану для пошукових систем (SEO) і для передачі репрезентативного стану (REST).

- Підтримка використання розмітки в існуючих файлах сторінок ASP.NET (ASPX), елементів управління (ASCX) і головних сторінок (MASTER) як шаблонів представлень. Разом з платформою ASP.NET MVC можна використовувати існуючі функції ASP.NET, наприклад, вкладені головні сторінки, вбудовані вирази, декларативні серверні елементи управління, шаблони, прив'язку даних, локалізацію і т. д.

Недоліки:

- Високий поріг входження
- Відсутність механізму зберігання стану
- Складність створення бібліотек компонентів.

Кожен з вищезгаданих фреймворків має свої переваги та недоліки, тому їх вибір необхідно здійснювати з урахуванням проекту, який буде створюватися. Це є головним критерієм вибору, адже жоден із фреймворків не є універсальним рішенням для будь-якого проекту. Далі наведена таблиця порівняння основного функціоналу вищезгаданих фреймворків [4].

_ Таблиця 1.1 – Порівняльна характеристика популярних сучасних Back-end фреймворків.

Фреймворк							
Критерій	Z end	CakeP HP	Symfony 2	Yii	ASP.NET	ROR	Struts
Мова	P HP	PHP	PHP	PHP	ASP.NET	Ruby	Java
Шаблон проектування MVC	+	+	+	+	+	+	+
Застосування ORM	+	+	+	+	ORM- НЕЗАЛЕЖНА	+	+
Інтеграція з БД	+	+	+	+	***	+	***
Механізми інтернаціоналізації і локалізації	+	+	+	+	***	+	+
Використання шаблонів при створенні інтерфейсів користувача	+	+	***	+	+	+	+
Створення і перевірка форм	+	+	+	+	+	+	+
Управління доступом на основі ролей	+	+	***	+	+	+	HI
Використання AJAX	+	+	***	+	+	+	+
ЧПУ (Friendly URL)	+	+	+	+	+	+	HI
Модульне тестування (unit-testing)	+	+	***	+	+	+	+
Система кешування	+	+	+	+	+	+	HI

1.1.1 Переваги використання Back-end фреймворку

Під час вибору фреймворку можуть виникнути певні труднощі, пов'язані з визначенням завдань, які він може виконувати, та його призначення. Як бачимо, не кожен фреймворк підтримує ORM для роботи з базами даних, має якісну інтеграцію і хорошу документацію. Якщо для створення сайту потрібно знайти зручний і простий в освоєнні фреймворк, то необхідно ретельно підійти до питання його вибору та зважити всі «за» і «проти».

Розглянемо загальні переваги використання фреймворку:

1. гнучкість розробки та розвитку проекту;
2. ефективне використання ресурсів сервера;
3. відкритий код;
4. легкість і надійність web-розробок. фреймворк складається з базових, перевірених, налагоджених функцій і операцій. Він побудований на базі об'єктно-орієнтованого програмування;
5. постійний розвиток і вдосконалення фреймворка;
6. великий обсяг супровідної документації, прикладів з розробки на різних мовах;
7. світова популярність, велика кількість розробників;
8. легкість супроводу проекту надалі, так як розробка із застосуванням фреймворка заснована на певних угодах;
9. фреймворк дозволяє сконцентруватися на вирішенні архітектурних завдань, а не базових, як при розробці без його застосування;
10. фреймворк дозволяє вузько вирішувати поставлену задачу.

Одна з головних переваг фреймворка – зручна розробка нестандартних проектів. Жоден великий нестандартний проект (наприклад, twitter.com, фотобанк з онлайн купівлею фотографій, сайт знайомств і т.д.) не розроблений на готовій CMS – вони для цього не призначені. Усі оригінальні

проекти розробляють на фреймворках. Web-проект, розроблений за допомогою фреймворку, розвивається динамічно. При зміні вимог змінюється і сайт, достатньо лише замінити окремий блок (модуль), створити новий розділ або внести новизну в дизайні [5].

1.1.2 Недоліки використання Back-end фреймворку

Недоліки використання фреймворку досить умовні і незначні порівняно з перевагами:

- 1 файл = 1 клас;
- багато коду не використовується і лежить мертвим вантажем в проекті;
- складність в освоєнні;
- готових модулів і компонентів, які міг би встановити клієнт, в мережі Інтернет немає ані безкоштовних, ані платних. Всі доробки необхідно замовляти у розробників;
- розробка функціоналу, стандартного для сайтів на CMS, обійдеться дорожче і займе більше часу.

1.1.3 Шляхи та області використання

Для того щоб скористатися всіма можливостями фреймворка потрібен чималий багаж знань в розробці додатків. РНР-фреймворки можуть допомогти усунути дуже часту помилку при програмуванні додатків, а саме повторення коду, а також систематизувати процес розробки. Фреймворк є потужним інструментом для такої мови програмування, що швидко розвивається, як РНР, який допоможе організувати ваш код.

Кожна людина має різні вподобання і потреби. Для одного розробника використання фреймворків може допомогти у прискоренні процесу

програмування, а для іншого це може здатися марною тратою часу. У більшості випадків це залежить від рівня професіоналізму, але, загалом, фреймворки призначені, щоб заощадити час і абстрагуватися від рутинних завдань.

В основному, фреймворки застосовуються для розробки проектів складніше, ніж дво- або тресторінковий сайт з текстовими сторінками [7].

1.2 Аналіз Front–End фреймворків

Для створення адаптивного дизайну використовуються CSS фреймворки. В даний час існують десятки таких фреймворків, кожний з яких має свої переваги й недоліки : деякі з них мають гарний функціонал , але велику вагу. Далі буде розглянуто найпопулярніші з них.

Twitter Bootstrap - це безкоштовний набір інструментів з відкритим вихідним кодом, призначений для створення веб-сайтів та веб-застосунків, він містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Bootstrap спрощує розробку динамічних веб-сайтів і веб-застосунків.

Він є одним із найвідоміших CSS-фреймворків: на ньому створено мільйони проектів і в найближчому майбутньому буде створено не менше. Можливо, причина надзвичайної популярності криється в ранньому старті, або, можливо, не останню роль зіграло шляхетне походження (проект створювався в компанії Twitter). У будь-якому випадку Bootstrap не здає позиції і сьогодні вважається одним із найпопулярніших фреймворків.

На мою думку, головною в цьому фреймворку є сітка. Так, остання версія Bootstrap підтримує адаптивну 12-ти колонкову сітку. Мобільні пристрої теж ні в якому разі не забуті. Починаючи з третьої версії фреймворку, діє підхід “Спочатку мобільні”(спочатку розробляється версія

сайту для мобільних пристроїв).

UI набір елементів в Bootstrap одним із найбільших. Є елементи керування на всі випадки життя, а якщо чогось не вистачає, то в більшості випадків необхідне доповнення легко відшукати в мережі. Популярність робить свою справу, тому ситуації, коли ви чогось не зможете знайти для bootstrap зводяться до мінімуму [8].

Bootstrap однаково відноситься до любителів Sass і Less. Відразу після виходу релізу CSS на LESS з'являється порт на Sass.

Розробники Bootstrap прекрасно розуміють, що їхній проект розрісся і навряд чи всі його компоненти та можливості будуть затребувані в межах одного додатка. Щоб не тягнути за собою купу непотрібних файлів, на офіційному сайті надається можливість зібрати свій дистрибутив Bootstrap, включивши в нього тільки необхідні елементи. Виділимо переваги та недоліки цього фреймворку:

Переваги:

- хороша документація;
- адаптивна філософія побудови сайту;
- кроссбраузерність;
- розрахований на швидке створення сайту.

Недоліки:

- через його велику популярність багато сайтів мають однаковий вигляд, тому дуже складно створити унікальний дизайн;
- велика кількість класів;
- розмір фреймворку дуже великий, що сповільнює завантження сторінки.

Zurb Foundation – це один з найпопулярніших фреймворків , для створення адаптивного дизайну. За своєю популярністю цей фреймворк не

поступається Bootstrap. Кількість зірок на GitHub зайвий раз підтверджує довіру розробників до цього фреймворку. Foundation був створений в компанії Zurb (звідси і назва), що спеціалізується на створенні дизайнів веб-додатків. Створювався проект, в першу чергу, для власних потреб, тому з коробки націлений вирішувати реальні завдання.

Foundation пропонує колекцію готових компонентів (віджетів), 12-ти колонкову сітку та інші звичні речі. Бібліотека UI компонент виглядає солідно і її цілком вистачить для роботи над реальними прототипами. Співтовариство навколо Foundation сформувалося давно, тому проблем з пошуком додаткових плагінів/інформації просто не існує.

Починаючи з четвертої версії, Foundation виводить на передній план принцип «спочатку мобільні». У випадку з Foundation це передбачає повну адаптивність і відмову від застосування бібліотеки jQuery на користь zepto.js.

З цікавих особливостей варто відзначити схожість з Bootstrap, тому при наявності досвіду перейти на новий фреймворк буде в рази простіше. Виділимо переваги та недоліки цього фреймворку:

Переваги:

- Вбудована валідація форм;
- Підтримка RTL;
- Легко розширити функціонал, тоу що існує велика кількість доповнень;
- Хороша підтримка мобільними пристроями. Недоліки:
- Високий поріг входження.

Semantic UI - фреймворк для побудови призначеного для користувача інтерфейсу веб-додатку з широкими можливостями по налаштуванню і “повне право вільно вирішувати дизайн”, як обіцяють самі розробники. Зараз набір включає 50+ елементів UI і 3000+ змінних CSS, але колекція швидко поповнюється. Фреймворк повністю адаптивний, всі компоненти

визначаються за допомогою `ems` або `gems`.

Один з основних наголосів фреймворка - семантичність класів. Компоненти, пропонувані Semantic UI включають кнопки, анімовані кнопки, роздільники, значки, зображення, мітки, спливаючі вікна, сегменти, кастомізовані `checkbox`'и і `radio`, `text loader`'и і багато іншого. Також неможливо не відзначити наявність цікавих прийомів. Наприклад, ви можете використовувати мітки, щоб відобразити «стрічку» в кутку елемента. [7]

Інші особливості фреймворка включають можливість додати 'Disabled' стан картинкам і піктограмам, в яких можуть бути відформатовані різні кольори і розміри. Ряд популярних іконок включений завдяки сервісу Fontello.

Можливості:

- Проста у використанні;
- Різноманітність компонентів;
- Теми;
- Семантичний;
- Дуже активний розвиток;
- Має більше можливостей, ніж бутстрап;
- компоненти для користувацького інтерфейсу.

Недоліки:

- Не підтримує препроцесори SASS, LESS;

Кожен з вищерозглянутих фреймворків має свої переваги та недоліки. Тому вибір фреймворка потрібно робити з урахуванням проекту, який буде створюватися. Це є головним критерієм вибору, адже жоден з фреймворків не є універсальним рішенням для будь-якого проекту.

Далі приведена таблиця порівняння основного функціоналу вищерозглянутих фреймворків.

Таблиця 1.2 – Порівняльна характеристика популярних Front-end фреймворків для створення адаптивного дизайну [9].

	Bootstrap	Zurb Foundations	Semantic UI
Препроцесори	LESS, SASS	SCSS	-
JS-бібліотеки	jQuery	jQuery	jQuery
Підтримка плагінів	-	FastClick, Modernizr	-
Адаптивна сітка	+	+	+
Навбар	+	+	+
Слайдери	+	+	-
Випадаючі списки	+	+	+
Canvas меню	-	+	-

1.2.1 Переваги і недоліки застосування Front-end фреймворків

З розглянутого вище можна виділити такі переваги й недоліки використання CSS-фреймворків у своєму проекті:

Переваги:

- Блочна верстка;
- Ретельно підібрані стилі покращують читабельність контенту;
- Кроссбраузерність;

- Пришвидшення розробки;
- Одноманітність коду дозволяє знизити кількість конфліктів при командній розробці;
- Допомогає недосвідченому розробнику правильно створювати HTML-макет;
- Дає можливість використовувати генератори коду.

Недоліки:

- Залежність від сторонньої бібліотеки;
- Деякі фреймворки використовують префікси, що ускладнює розуміння коду;
- Великий розмір фреймворків.

Як можна побачити CSS, фреймворки мають невелику кількість недоліків

у порівнянні з їх перевагами.

1.3 Висновки до розділу 1

Системи створення web-додатків або фреймворки активно використовуються розробниками при створенні web-додатків із різним функціоналом і рівнем складності. Оглянувши основні характеристики і можливості сучасних фреймворків ми можемо вибрати оптимальний варіант для конкретних web-додатків з урахуванням поставлених завдань. Вибір та використання фреймворків відіграє важливу роль при проектуванні, реалізації та супроводі як простих web-додатків, так і складних програмних комплексів.

Вибираючи фреймворк, необхідно звертати увагу на те, наскільки він легкий в освоєнні і розумінні. Це має дуже важливе значення для малодосвідченого програміста. Також необхідно переконатися, що база даних і web-сервер сумісні з архітектурою обраного фреймворку. Поширеною помилкою ще є неправильна установка фреймворку. Тому під час установки необхідно слідувати інструкції, щоб уникнути помилок.

2 ВИМОГИ ДО САЙТУ “БАЗИ МЕТОДИЧНИХ ПОСІБНИКІВ”. ВИБІР СИСТЕМИ УПРАВЛІННЯ БАЗАМИ ДАНИХ. ВИБІР ФРЕЙМВОРКУ ДЛЯ РЕАЛІЗАЦІЇ. ОГЛЯД РОБОТИ З ОБРАНИМ ФРЕЙМВОРКОМ.

В даному розділі буде описано вимоги до сайту «Бази методичних посібників» , який буде реалізований з використанням фреймворку. Також буде здійснено вибір фреймворку для реалізації тестового проекту.

2.1 Вимоги до сайту “Бази методичних посібників”.

Проектом для огляду можливостей фреймворків є створення сайту “Бази методичних посібників ”. Основна задача сайту – зберігання даних у відповідному форматі та генерація бази методичних посібників у потрібному вигляді. Для реалізації цієї функції треба розробити архітектуру БД, яка надасть змогу зберігати інформацію у зручному виді. Також необхідно розробити інтерфейс користувача.

Потрібно звернути увагу на відповідність даних державним стандартам, оскільки список публікацій формується згідно з нормами бібліографічного опису, які установлені в ДСТУ ГОСТ 7.1:2006.

Для уникнення помилок система має бути закритою. Адміністратор повинен самостійно додавати та видаляти авторів, що зможуть додавати свої посібники. Також треба розробити можливість редагування або видалення вже існуючих посібників. Для економії місця на диску та зменшення навантаження слід передбачити можливість додавання публікації до списку певного автора при зазначенні його співавтором.

Вимоги до сайту :

1) Вимоги до реалізації сайту:

1. Мова програмування PHP;
2. Використання бази даних MySQL для зберігання інформації;
3. Сервер Apache;
4. Сайт має мати високу бистродію;
5. Архітектура MVC.

2) Сайт повинен містити такі розділи:

1. Головна (основний розділ сайту, в якому будуть знаходитися методичні посібники)
2. FAQ
3. Про сайт
4. Вхід

3) На сайті повинна бути реалізована авторизація.

4) Сайт повинен мати такі ролі користувача:

1. Адміністратор (zareєстрований користувач, якому присвоєна роль адміністратора)
 - a. Реєстрація користувачів
 - b. Додавання методичних посібників
 - c. Видалення методичних посібників
 - d. Видалення користувачів
 - e. Перегляд методичних посібників
 - f. Пошук методичних посібників
2. Викладач(zareєстрований користувач, якому присвоєна роль викладач):
 - a. Додавання методичних посібників
 - b. Перегляд методичних посібників
 - c. Пошук методичних посібників

3. незареєстрований користувач, якому присвоєна роль викладач)
 - a. Перегляд методичних посібників
 - b. Пошук методичних посібників
- 5) Вимоги до дизайну сайту:
 1. Сайт повинен бути адаптивним та кроссбраузерним
 2. Не можна буде додати на сайт посібники з однаковими назвами
 3. Підтримка роботи на мобільних пристроях.
- 6) Вимоги до оформлення методичного посібника згідно з нормами бібліографічного опису, які установлені в ДСТУ ГОСТ 7.1:2006.

2.2 Аналіз структури формування опису методички

Згідно з Указом Держкомітету України з питань технічного регулювання і споживчої політики від 10.11.2006 р. № 322 з 1 липня 2007 року в Україні діє міждержавний стандарт ДСТУ 7.1:2006 “Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис, бібліографічний опис. Загальні вимоги та правила складання” [10].

Опис методичного посібника і формується у відповідності зі стандартами України, а саме з ДСТУ ГОСТ 7.1:2006, в якому описані формати для кожного виду публікацій (методичний посібник, книга, журнал, тощо). На пошук потрібного формату та на те, щоб привести до відповідного виду кожний бібліографічний опис витрачається достатньо багато часу, який міг бути використаний на інші важливі роботи.

Наш веб-сайт дасть змогу спростити цю задачу лише правильним заповненням форми додавання методичного посібника.

2.3 Вибір системи управління базами даних для реалізації сайту “База методичних посібників”

На сьогоднішній день існує досить багато систем управління базами даних. Було розглянуто деякі з них.

Oracle Database або Oracle RDBMS. Об'єктно-реляційна система управління базами даних компанії Oracle Corporation. Ця СУБД забезпечує ефективне, надійне і безпечне управління даними таких критично важливих для бізнесу додатків, як онлайніві середовища, виконує масштабну обробку транзакцій (OLTP), сховища даних з високою інтенсивністю потоку запитів, а також ресурсомісткі інтернет-додатки. Редакція Oracle Database Enterprise Edition надає інструментальні засоби і функції, що забезпечують відповідність вимогам сучасних корпоративних додатків в області доступності та масштабованості. Ця редакція містить всі компоненти Oracle Database, а також допускає розширення за допомогою придбання додаткових модулів та програм.

Система Oracle Database дозволяє звертатися до даних з будь-якого додатку, розробленого із застосуванням технологій Microsoft. NET, Visual Studio та веб-додатків. Основною умовою є лише наявність справних бібліотек, що дають змогу підключатися до серверу бази даних Oracle.

Oracle Database є комерційною СУБД, але є її безкоштовна версія, яку можна без проблем скачати прямо з офіційного сайту компанії.

Microsoft SQL Server. Система Microsoft SQL Server відштовхується від концепції платформи даних Майкрософт: вона спрощує управління будь-якими даними в будь-якому місці і в будь-який момент часу. Вона дозволяє зберігати в базах даних інформацію, отриману з структурованих, напівструктурованих і неструктурованих джерел, таких як зображення та музика. У SQL Server 2008 є великий набір інтегрованих служб, які

розширюють можливості використання даних: можна складати запити, виконувати пошук, проводити синхронізацію, робити звіти, аналізувати дані. Всі дані зберігаються на основних серверах, що входять до складу центру обробки даних. До них здійснюється доступ з настільних комп'ютерів і мобільних пристроїв. Таким чином, можна повністю контролювати дані незалежно від того, де вони збережені.

Система SQL Server дозволяє звертатися до даних з будь-якого додатку, розробленого із застосуванням технологій Microsoft .NET та Visual Studio, а також в межах сервісно-орієнтованої архітектури і бізнес-процесів – через Microsoft BizTalk Server. SQL Server дозволяє створити надійну, продуктивну, інтелектуальну платформу, що відповідає всім вимогам по роботі з даними.

SQL Server є комерційною СУБД.

PostgreSQL. Безкоштовна об'єктно-реляційна система управління базами даних.

Система PostgreSQL заснована на ядрі, створеному безліччю розробників. У подібних випадках розумно зосередитися на оснащенні системи новими можливостями, але не займатися оптимальним їх втіленням, оскільки у випадку виникнення необхідності завжди можна буде повернутися до оптимізації відповідних ділянок коду [11].

MySQL. Безкоштовна система управління базами даних. MySQL є власністю компанії Oracle Corporation, що отримала її разом з поглиненою Sun Microsystems, що здійснює розробку і підтримку програми. Розповсюджується під GNU General Public License або під власною комерційною ліцензією. Крім цього, розробники створюють функціональність за замовленням ліцензійних користувачів.

MySQL є рішенням для малих і середніх додатків. Входить до складу

серверів WAMP, LAMP і в портативні збірки серверів Denver, XAMPP. Зазвичай MySQL використовується як сервер, до якого звертаються локальні або віддалені клієнти, проте в дистрибутив входить бібліотека внутрішнього сервера, що дозволяє включати MySQL в автономні програми.

Оскільки інтерфейс системи буде веб-додатком, потрібно обрати СУБД, з якою можуть співпрацювати мови програмування для написання веб-додатків. Вибір було зупинено на MySQL.

Висока продуктивність MySQL на вузькому колі задач – прямий наслідок її функціональної простоти.

Отже, для наших цілей буде обрано MySQL.

2.3.1 Вибір архітектури бази даних

Базою даних є представлена в об'єктивній формі сукупність методичних посібників, систематизованих таким чином, щоб ці посібники могли бути знайдені та оброблені за допомогою електронної обчислювальної машини.

Найбільш часто використовуються наступні характерні ознаки:

1. БД зберігається і обробляється в обчислювальній системі. Таким чином, будь-які не комп'ютерні сховища інформації (архіви, бібліотеки, картотеки і т.і.) базами даних не є.
2. Дані в БД логічно структуровані (систематизовані) з метою забезпечення можливості їх ефективного пошуку та обробки в обчислювальній системі. Структурованість має на увазі явне виділення складових частин (елементів), зв'язків між ними, а також типізацію елементів і зв'язків, при якій з типом елемента (зв'язку) співвідноситься певна семантика і допустимі операції.
3. БД включає метадані, що описують логічну структуру у формальному

вигляді.

Схема включає в себе описи змісту, структури і обмежень цілісності, які використовуються для створення і підтримки бази даних. База даних включає в себе набір постійних даних, визначених за допомогою схеми. Система управління даними використовує визначення даних у схемі для забезпечення доступу і управління доступом до даних в базі даних.

Існує величезна кількість різновидів баз даних, що відрізняються за різними критеріями.

Було розглянуто основну класифікацію БД по моделям даних для здійснення вибору оптимальної архітектури.

Ієрархічна модель бази даних складається з об'єктів з покажчиками від батьківських об'єктів до нащадків, з'єднуючи разом пов'язану інформацію. Ієрархічні бази даних можуть бути представлені як дерево, що складається з об'єктів різних рівнів. Верхній рівень займає один об'єкт, другий – об'єкти другого рівня і т. д. Між об'єктами існують зв'язки, кожен об'єкт може включати в себе кілька об'єктів більш низького рівня. Такі об'єкти перебувають у відношенні предка (об'єкт, більш близький до кореня) до нащадка (об'єкт більш низького рівня), при цьому можливе, коли об'єкт-предок не має нащадків або має їх кілька, тоді як в об'єкта-нащадка обов'язково тільки один предок. Об'єкти, що мають загального предка, називаються близнюками.

Приклад ієрархічної бази даних зображений на рис. 2.1.



Рисунок 2.1 - Приклад ієрархічної архітектури БД [12].

Модель мережевої бази даних. До основних понять мережевої моделі бази даних належать: рівень, елемент (вузол), зв'язок. Вузол – це сукупність атрибутів даних, що описують деякий об'єкт. У мережній структурі кожний елемент може бути пов'язаний з будь-яким іншим елементом. Мережеві бази даних подібні ієрархічним, за винятком того, що в них є покажчики в обох напрямках, які з'єднують споріднену інформацію. Незважаючи на те, що ця модель вирішує деякі проблеми, пов'язані з ієрархічною моделлю, виконання простих запитів залишається досить складним процесом. Також, оскільки логіка процедури вибірки даних залежить від фізичної організації цих даних, то ця модель не є повністю незалежною від програми. Іншими словами, якщо необхідно змінити структуру даних, то потрібно змінити і додаток. На рис 2.2 приведений приклад мережевої архітектури.

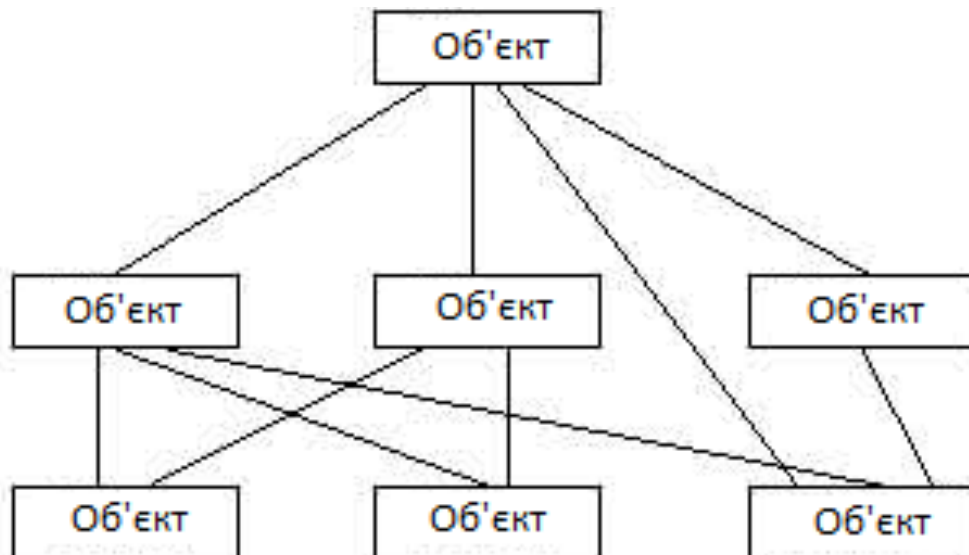


Рисунок 2.2 – Приклад мережевої архітектури БД [12].

Реляційна база даних – база даних, заснована на реляційній моделі даних. Термін «реляційний» означає, що теорія заснована на математичному понятті відношення (relation). В якості неформального синоніма терміну «відношення» часто зустрічається слово таблиця. Необхідно пам'ятати, що «таблиця» є поняття нестроге і неформальне і часто означає не «відношення» як абстрактне поняття, а візуальне уявлення відношення на папері або екрані.

Реляційна модель приховує деталі фізичного зберігання даних. Уся робота ведеться на логічному рівні. Так легше виявляти відносини, що існують між елементами даних.

Таблиця складається з рядків (записів) і стовпців (полів). У стовпці зберігаються відповідні значення кожного рядка; "пропусків" або коротких стовпців бути не може. Запис є окремою сутністю, а поля являють собою атрибути записів.

Для проектування БД розроблюваної системи обрана реляційна архітектура БД. Саме за її допомогою забезпечено такий рівень зв'язку, який дозволить швидко та зручно отримувати доступ до даних та формувати частини звітів, а також вести потрібні підрахунки. Дана архітектура передбачає можливість зручного нарощування об'ємів інформації, що є

важливим, адже БД буде нарощуватися з часом. Крім того, у реляційній моделі досягається інформаційна й структурна незалежність. Записи не зв'язані між собою настільки, щоб зміна однієї з них торкнулася інших, а зміна структури бази даних не обов'язково приводить до перекомпіляції працюючих з нею додатків.

Отже, розширення при необхідності структури БД не повинно викликати труднощів.

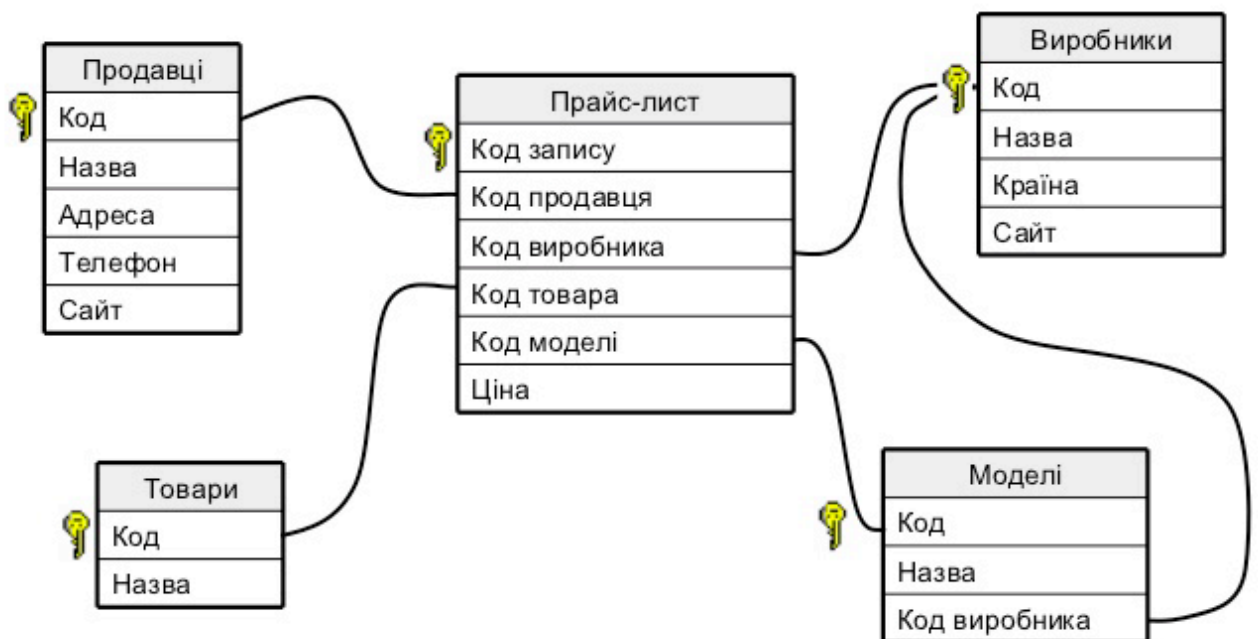


Рисунок 2.3 – Приклад реляційної БД [12].

2.4 Вибір Back – End фреймворку для реалізації сайту “База методичних посібників”

З описаного вище функціоналу сайту, а саме:

- Складність проекту(середня)
- Механізм авторизації на основі сесій
- Розподіл ролей
- Валідація форм
- Генерація капчі
- Використання бази даних MySQL для зберігання інформації можна побачити, що використання фреймворку для Back – End є доцільним.

Вибір фреймворку буде здійснено серед PHP фреймворків згідно з критеріями , які було розглянуто в минулому розділі. Проаналізувавши інформацію з різних джерел, вибір фреймворка можна звести до таблиці 2.1

Таблиця 2.1 – Вибір Back – End фреймворку

Назва	Zend	CakePHP	Symfony2	Yii
Підтримка Базы MySQL	+	+	+	+
Підтримка спільноти	Мала російськомовна спільнота	Не дуже активна	Відсутня російськомовна	Документація , форум, блоги
Документація	Мало російськомовної	Погана	Майже відсутня російськомовна	Офіційний мануал російською
Продуктивність	+	+++	++	++++
Безпека	Немає вбудованого захисту від SQL атак	Немає вбудованого захисту від SQL атак	Немає вбудованого захисту від SQL атак	+
Поріг освоєння	PHP5, ООП, шаблони проектування.	PHP, ООП, вміння розбиратися в вихідному коді фреймворка.	PHP5, ООП, ORM, консоль.	PHP5, ООП.
Швидкість розробки	Низька	+	+	+

Таблиця 2.1 – Вибір Back – End фреймворку (продовження)

Назва	Zend	CakePHP	Symfony2	Yii
Швидкість розвитку	+	+	+	+
Зворотна сумісність	Проекти на версії 1.x сумісні між собою	Проекти сумісні на версія від 1.x до 3.0 з мінімальною кількістю допрацювань	Проекти на версії 2.x повністю сумісні	Проекти на версії 1.x сумісні між собою.
Використання шаблонів при створенні інтерфейсів користувача	+	+	+	+
Створення і перевірка форм	+	+	+	+

Згідно з таблицею, по більшості параметрів, які є важливими для проекту, що створюється, лідером став фреймворк Yii. Основними критеріями для нас є безпека, поріг освоєння, а також, що найголовніше для проекту, висока продуктивність додатків, створених на Yii. Тому для створення проекту буде використано фреймворк Yii.

2.5 Yii як програмний каркас для створення web-додатків

Yii - це фреймворк для веб-програмування загального призначення, який може бути використаний для розробки практично будь-яких веб-додатків. Завдяки наявності засобів кешування, Yii особливо підходить для розробки додатків з великим потоком трафіку, таких як портали, форуми, системи управління контентом (CMS), системи електронної комерції та ін [14].

2.5.1 Структура Yii

Yii використовує шаблон проектування Модель-Представлення-Контролер (MVC, Model-View-Controller), який широко застосовується в веб-програмуванні.

MVC призначений для поділу бізнес-логіки і користувача інтерфейсу, щоб розробники могли легко змінювати окремі частини програми, не зачіпаючи інші. В архітектурі MVC модель надає дані і правила бізнес-логіки, уявлення відповідає за користувальницький інтерфейс (наприклад, текст, поля введення), а контролер забезпечує взаємодію між моделлю і представленням. Крім цього, Yii використовує фронт-контролер, який ще називають додатком (application), що інкапсулює контекст обробки запиту.

Додаток збирає інформацію про запит і передає її для подальшої обробки відповідного контролера.

Наступна діаграма відображає структуру програми Yii:

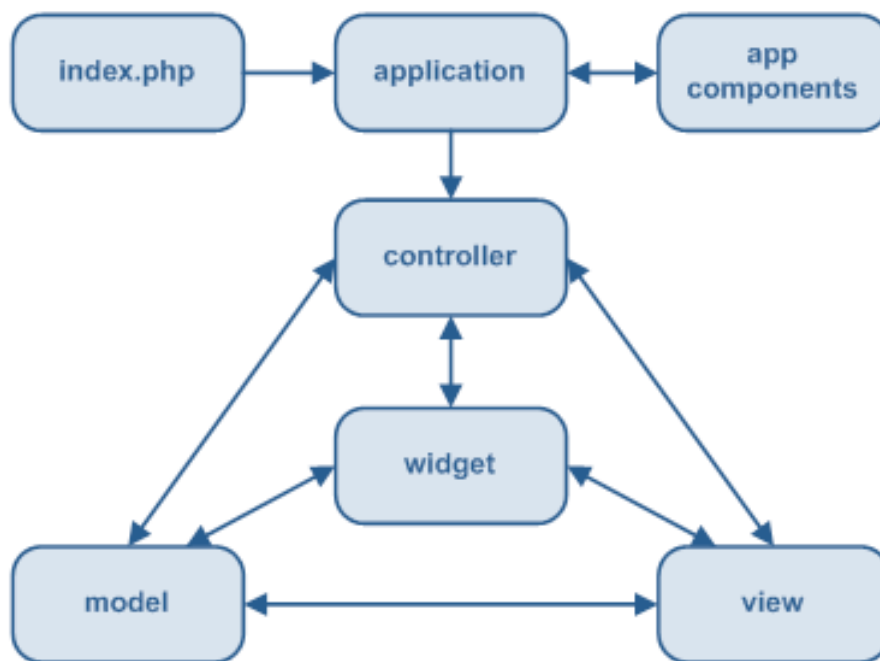


Рисунок 2.4 - Статична структура додатку Yii [15].

2.5.2 Ключові компоненти додатку Yii

Yii визначає набір компонентів ядра, які надають можливості, необхідні для більшості веб-додатків. Наприклад, компонент request використовується для збору інформації про запит користувача і надає різну інформацію, таку як URL і cookies. Ставлячи властивості компонентів, можна змінювати стандартну поведінку Yii практично як завгодно [14].

Далі перераховані ключові компоненти, які використовує клас `CWebApplication`:

- `assetManager`: `CAssetManager` — керує публікацією файлів ресурсів (asset files);
- `authManager`: `CAuthManager` — контролює доступ на основі ролей (RBAC);
- `cache`: `CCache` — надає можливості кешування даних; врахуйте, що ви повинні вказати використовуваний клас (наприклад, `CMemCache`, `CDbCache`), інакше при звертанні до компонента буде повернуто `null`;
- `clientScript`: `CClientScript` — керує клієнтськими скриптами (javascripts і CSS);
- `coreMessages`: `CPhpMessageSource` — надає переклади системних повідомлень Yii-Фреймворка;
- `db`: `CDbConnection` — обслуговує з'єднання з базою даних; зверніть увагу, що для використання компонента необхідно встановити властивість `connectionString`;
- `errorHandler`: `CErrorHandler` — оброблює не піймані помилки й виключення PHP;
- `format`: `CFormatter` - форматує дані для їх подальшого відображення;
- `messages`: `CPhpMessageSource` — надає переклади повідомлень, використаних в Yii-додатку;
- `request`: `CHttpRequest` — надає інформацію, що відноситься до запиту користувача;
- `securityManager`: `CSecurityManager` — забезпечує функціональність, пов'язану з безпекою (наприклад, хешування, шифрування);
- `session`: `CHttpSession` — забезпечує функціональність, пов'язану із сесіями;

- `statePersister: CStatePersister` — надає метод для збереження глобального стану;
- `urlManager: CUrlManager` — надає функції парсинга й формування URL;
- `user: CWebUser` — надає ідентифікаційну інформацію поточного користувача;
- `themeManager: CThemeManager` — управляє темами оформлення.

2.5.3 Реалізація MVC в Yii

Контролер (controller) — це екземпляр класу `CController` або успадкованого від нього класу. Він створюється об'єктом додатку тоді, коли користувач робить відповідний запит. Під час запуску контролер виконує відповідну дію, що зазвичай передбачає створення відповідних моделей і рендеринг необхідних представлень. У найпростішому випадку дія — це метод класу контролера, назва якого починається з `action` [13].

У контролера є дія за замовчуванням, яка виконується у випадку, коли користувач не вказує дію при запиті. За замовчуванням ця дія називається `index`. Змінити її можна шляхом встановлення значення `CController::defaultAction`.

Модель (model) — це екземпляр класу `CModel` або класу, успадкованого від нього. Модель використовується для зберігання даних або застосованих до них бізнес-правил.

Модель представляє собою окремий об'єкт даних. Це може бути запис таблиці бази даних або HTML-форма з полями для вводу даних. Кожне поле об'єкту даних представляється атрибутом моделі, кожний атрибут має мітку та може бути перевірений на коректність за допомогою набору правил.

Yii надає два типи моделей: модель форми та Active Record. Обидва типи є розширенням базового класу CModel.

Модель форми — це екземпляр класу CFormModel. Вона використовується для зберігання даних, введених користувачем. Як правило, ми отримуємо ці дані, обробляємо, а потім позбавляємося від них. Наприклад, на сторінці авторизації модель такого типу може бути використана для представлення інформації про ім'я користувача та пароль. Детальний опис роботи з формами подано у розділі Робота з формами.

Active Record (AR) — це шаблон проектування, який використовується для абстрагування доступу до бази даних у об'єктно-орієнтованій формі. Кожен об'єкт AR являє собою екземпляр класу CActiveRecord або класу, успадкованого від нього, та представляє окремий рядок у таблиці бази даних. Поля цього рядку відповідають властивостям AR-об'єкту. Детальніше з AR-моделлю можна ознайомитись у розділі Active Record.

Представлення — це PHP-скрипт, який складається переважно з елементів користувацького інтерфейсу. Він може охоплювати вирази PHP, проте рекомендується, щоб ці вирази не змінювали дані і залишалися відносно простими. Згідно з концепцією розділення логіки та представлення, більша частина коду логіки повинна бути розміщена у контролері або моделі, а не у скрипті представлення.

Представлення має ім'я, яке використовується з метою ідентифікації файлу скрипта представлення у процесі рендерингу. Ім'я представлення повинно співпадати із назвою файлу представлення. Наприклад, для представлення edit відповідний файл скрипта повинен називатися edit.php. Щоб відобразити представлення необхідно викликати метод CController::render(), вказавши ім'я представлення, при цьому метод спробує виявити відповідний файл у директорії protected/views/ControllerID.

Всередині скрипта представлення екземпляр контролера доступний через `$this`. Таким чином, ми можемо звернутися до властивостей контролера із коду представлення: `$this->propertyName`.

2.5.4 Типовий запит Yii

Розглянувши структуру додатків у Yii, перейдемо до специфіки взаємодії кожного компонента із системою для виконання базового запиту.

Наступна діаграма описує типову послідовність процесу обробки користувальницького запиту додатком:

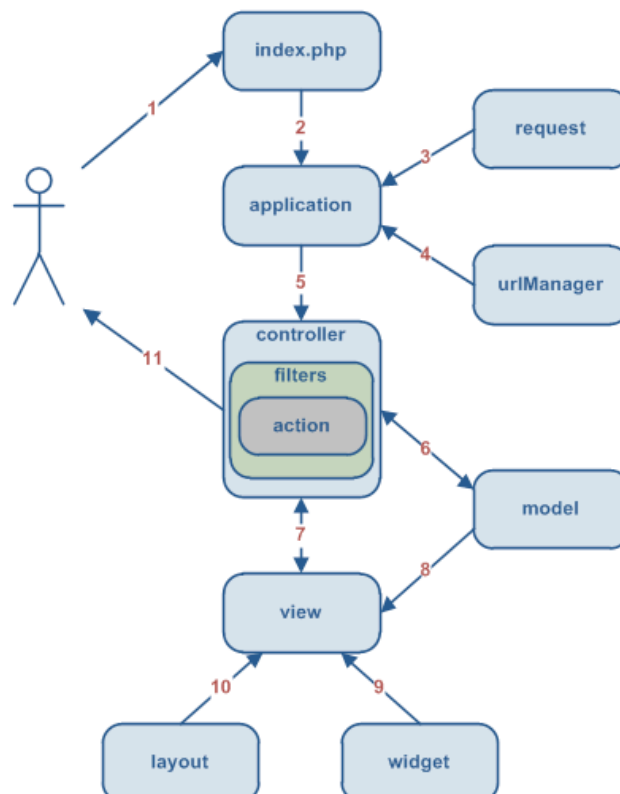


Рисунок 2.5 - Типова послідовність роботи додатку Yii [14]

Детальний опис послідовності процесу обробки користувальницького запиту додатком :

1. Користувач здійснює запит за допомогою URL `http://www.example.com/index.php?r=post/show&id=1`, і веб-сервер обробляє його, запускаючи скрипт ініціалізації `index.php`.
2. Скрипт ініціалізації створює екземпляр додатку і запускає його на виконання.
3. Додаток отримує детальну інформацію про запит користувача від компонента додатка `request`.
4. Додаток визначає запитані контролер і дію за допомогою компонента `urlManager`. В даному прикладі контролером буде `post`, що відноситься до класу `PostController`, а дією - `show`, суть якого визначається контролером.
5. Додаток створює екземпляр запитуваного контролера для подальшої обробки запиту користувача. Контролер визначає відповідність дії `show` методу `actionShow` в класі контролера. Далі створюються і застосовуються фільтри (наприклад, `access control`, `benchmarking`), пов'язані з даним процесом, і, якщо фільтри дозволяють, дія виконується.
6. Дія зчитує з бази даних модель `Post` з ID рівним 1.
7. Дія підключає уявлення `show`, передаючи в нього модель `Post`.
8. Подання отримує і відображає атрибути моделі `Post`.
9. Подання підключає деякі віджети.
10. Сформоване уявлення вставляється в макет сторінки.
11. Дія завершує формування уявлення і виводить результат користувачеві.

2.5.5 Життєвий цикл додатку Yii

Життєвий цикл додатку при обробці користувальницького запиту:

1. Попередня ініціалізація додатку через `CApplication::preinit()`.
2. Ініціалізація обробника помилок.
3. Реєстрація компонентів ядра.
4. Завантаження конфігурації додатку.
5. Ініціалізація додатку `CApplication::init()`:
 - a. реєстрація поведінок додатку;
 - b. завантаження статичних компонентів додатку;
6. Виклик події `onBeginRequest`.
7. Обробка запиту:
 - a. збір інформації про запит;
 - b. створення контролера;
 - c. запуск контролера;
8. Виклик події `onEndRequest`.

2.5.6 Процес розробки додатку з використанням Yii

Розглянувши фундаментальні концепції Yii, доцільно описати загальний процес створення веб-додатків з використанням фреймворку. Процес ґрунтується на тому, що аналіз вимог уже проведений, як і аналіз структури додатку [15].

Основні етапи при розробці додатку виглядають так:

- 1) Створення структури директорій. Утиліта `yii` може бути використана для того, щоб прискорити цей процес.

- 2) Конфігурація додатку шляхом модифікації файлу конфігурації програми. Цей етап також може зажадати написання деяких компонентів програми (наприклад, компонента управління користувачами).
- 3) Створення класу моделі для кожного використовуваного типу даних. Для автоматичної генерації всіх необхідних моделей Active Record можна скористатися інструментом Gii, описаним в розділах «створення першого додатку» та «автоматична генерація коду».
- 4) Створення класу контролера для кожного типу користувальницького запиту. Класифікація призначених для користувача запитів залежить від поточних вимог. У загальному випадку, якщо клас моделі використовується користувачем, повинен існувати відповідний клас контролера. Утиліта Gii також може автоматизувати цей процес.
- 5) Створення дій і уявлень. Саме тут і відбувається основна робота.
- 6) Конфігурація необхідних фільтрів для дій в класах контролерів.
- 7) Створення тем оформлення при необхідності.
- 8) Переклад повідомлень в разі, коли потрібна локалізація додатку.
- 9) Виявлення даних і уявлень, які можуть бути закешовані, і застосування відповідних технік кешування.
- 10) Налаштування продуктивності і розгортання.

2.6 Висновки до розділу 2

Після аналізу вимог сайту «Бази методичних посібників» ,було обрано базу даних MySQL та фреймворк Yii було обрано для створення проекту. Вибір був здійснений згідно з критеріями які були оглянуті в розділі 1. Yii має багато можливостей які виділяють цей фреймворк на фоні інших і роблять розробку з його допомогою дуже швидкою і приємною.

Ось невеликий список який варто виділити:

1. Відмінна підтримка. Для російськомовних користувачів на офіційному форумі фреймворка створений власний розділ де ви завжди можете отримати відповіді на свої питання.
2. Повна підтримка ООП. Фреймворк повністю заточений під п'яту версію php, що дозволяє підтримувати весь функціонал при об'єктно орієнтованому програмуванні.
3. Генератор коду. Yii надає відмінний вбудований генератор вихідного коду. Вказавши лише основні параметри – Yii генерує для вас загальну структуру програми яка буде містити всі необхідні моделі / контролери / відображення для старту.
4. Захист. Всі стандартні класи Yii заточені під високий рівень безпеки що при вмілому зверненні дозволяє повністю убезпечити свій сайт від Sql-Inj, XSS, CSRF і інших атак.

3 РОЗРОБКА САЙТУ ЗА ДОПОМОГОЮ YII

В даному розділі буде розглянуто створення сайту “Питання-відповіді КПП- Телеком” за допомогою використання WEB-фреймворку Yii . Буде детально описано етапи встановлення фреймворку на локальний сервер, створення бази даних для сайту , а також розробка сайту з використанням архітектури MVC. Всі вищезгадані аспекти будуть розглянуті в окремих підрозділах.

3.1. Вибір програмного забезпечення

Для оптимізації розробки проекту та зручності були використані наступні

програми:

1. Пакет “ Open Server ”(серверна платформа)
2. Графічний інтерфейс “ PhpMyAdmin ” для управління СУБД MySQL
3. Редактор програмного коду SublimeText

Кожен з цих додатків має низку унікальних, незамінних властивостей, які полегшують розробку і тестування web-додатку.

3.2 Розробка дизайну сайту

Хороший дизайн може надати власникові сайту багато переваг перед конкурентами, а також матеріальну вигоду. Залежно від того, які цілі ставить перед собою компанія при створенні сайту, дизайн може підкреслити статус компанії - власника, створити додаткову рекламу компанії або послуг, сформувати впізнаваний бренд і т.д.

Дизайн сайту було розроблено з використанням програми Photoshop 2015.

Головна сторінка сайту:

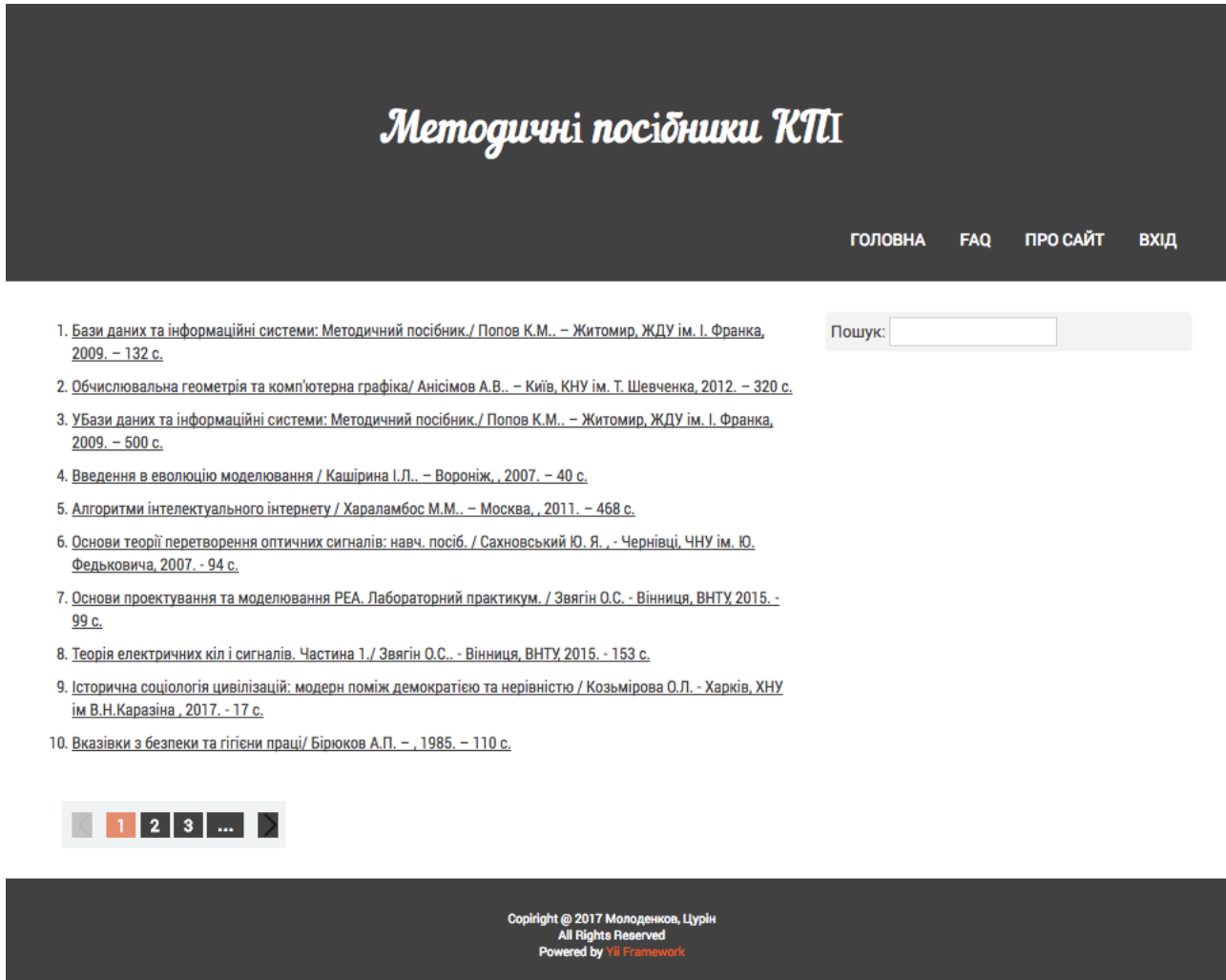


Рисунок 3.1 - Головна сторінка сайту.

Мобільна версія сайту:

В наш час великої популярності набули смартфони, тому, крім звичайної версії сайту, буде створено також мобільну версію (з використанням доповнення Yiistrap).

Методичні посібники КТІ



ГОЛОВНА

FAQ

ПРО САЙТ

ВХІД

4. [Введення в еволюцію моделювання / Кашірина І.Л.. – Воронеж, , 2007. – 40 с.](#)

5. [Алгоритми інтелектуального інтернету / Хараламбос М.М.. – Москва, , 2011. – 468 с.](#)

6. [Основи теорії перетворення оптичних сигналів: навч. посіб. / Сахновський Ю. Я. , - Чернівці, ЧНУ ім. Ю. Федьковича, 2007. - 94 с.](#)

7. [Основи проектування та моделювання РЕА. Лабораторний практикум. / Звягін О.С. - Вінниця, ВНТУ, 2015. - 99 с.](#)

Рисунок 3.2 - Мобільна версія сайту.

3.3 Розробка таблиць бази даних

Ключовим елементом БД являється таблиця. Необхідно пам'ятати, що в реляційній БД поняття «таблиця» є нестроге і неформальне, яке зазвичай означає не «відношення», як абстрактне поняття, а візуальне представлення

По функціонально-логічним ознакам таблиці поділяються на 3 види:

- таблиці сутності – основні таблиці, в яких міститься динамічно змінювана інформація (співробітники, студенти, публікації, дисципліни і т.д.). На даному етапі розробки системи супроводження публікацій за внесення цих даних відповідає адміністратор;

- таблиці-словники – статична інформація, яка вноситься в БД один раз і може інколи обновлятися. В основному це переліки типів, видів, статусів;
- таблиці зв'язку – таблиці які забезпечують зв'язок типу «багато-до-багатьох».

3.3.1 Створення БД за допомогою PhpMyAdmin

PhpMyAdmin - веб-додаток з відкритим кодом, написаний на мові PHP і представляє собою веб-інтерфейс для адміністрування СУБД MySQL. PhpMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати команди SQL і переглядати вміст таблиць і баз даних. Додаток користується великою популярністю у веб-розробників, тому що дозволяє управляти СУБД MySQL без безпосереднього введення SQL команд, надаючи дружній інтерфейс. Саме він буде використовуватися для роботи з базою даних в сайті, що створюється.

Створення БД за допомогою PhpMyAdmin включає такі етапи:

1. Вибираємо в головному вікні пункт “Создать БД”:

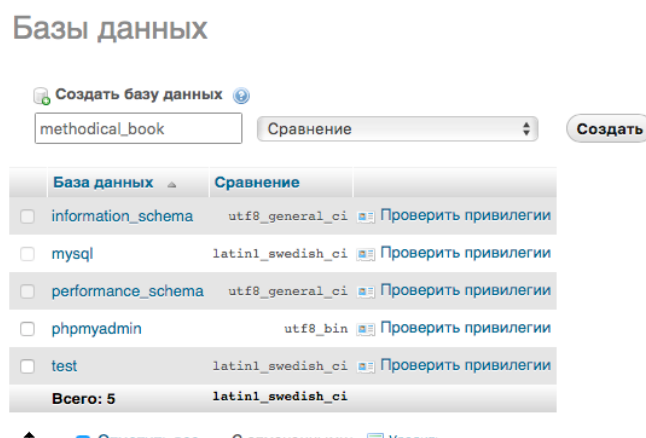


Рисунок 3.3 - Створення БД в PhpMyAdmin.

Тиснемо кнопку “Создать” і БД створена.

2. Після успішного створення БД для сайту створюємо необхідні таблиці:

Имя таблицы: log_pass Add 1 column(s) Вперёд

Имя	Тип	Длина/Значения	По умолчанию	Сравнение	Атрибуты	Null	Индекс
id	INT	11	Нет			<input type="checkbox"/>	PRIMARY
login	VARCHAR	15	Нет			<input type="checkbox"/>	---
password	VARCHAR	20	Нет			<input type="checkbox"/>	---
name	VARCHAR	45	Нет			<input checked="" type="checkbox"/>	---

Рисунок 3.4 - Створення таблиці log_pass.

Аналогічно створюємо решту таблиць.

3.3.2 Розробка таблиці сутності «Автори»

Таблиця сутності «Автори» містить усіх авторів, які можуть користуватися системою, а саме додавати та редагувати свої публікації.

Всі записи до таблиці додаються безпосередньо адміністратором.

На рис. 3.5 зображено таблицю "log_pass" («Автори»), що містить усі поля таблиці.

	#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/>	1	id 	int(11)			Нет	Нет
<input type="checkbox"/>	2	login	varchar(15)	latin1_swedish_ci		Нет	Нет
<input type="checkbox"/>	3	password	varchar(20)	latin1_swedish_ci		Нет	Нет
<input type="checkbox"/>	4	name	varchar(45)	latin1_swedish_ci		Да	NULL

Рисунок 3.5 – Таблица сутності «log_pass» – Автори

Інформація по окремих стовпцях таблиці:

- id – оригінальний номер запису в таблиці, який однозначно ідентифікує певного автора. Всі таблиці в БД мають такий стовпець, несучи однакове ідейне навантаження. У всіх випадках полю призначена властивість первинного ключа. Також цей стовпець має властивість автоінкременту;
- login – особистий ідентифікатор користувача, який буде їм надалі використовуватися для входу в систему;
- password – особистий пароль користувача, який буде служити захистом від небажаного додавання або редагування публікації сторонніми користувачами;
- name – ПІБ користувача, необхідне для генерації списку авторів публікацій;

Вибір типу даних для кожного поля (це стосується усіх таблиць БД) проводився з врахуванням максимально можливої кількості записів при використуванні підсистеми як «мультипідроздільної». Наприклад, для первинного ключа встановлений тип «INT».

Всі таблиці бази даних закодовані в UTF-8. В якості строкових типів даних використовується тип «varchar».

3.3.3 Розробка таблиці сутності «Публікації»

Таблиця «Публікації» містить в собі всі методичні посібники користувачів системи. Нові дані додаються авторами, які були зареєстровані та зайшли під своїм логіном та паролем.

На рис. 3.6 зображено таблицю "publications" («Публікації»), що містить усі поля таблиці.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/>	1 id	int(11)			Нет	Нет
<input type="checkbox"/>	2 type	varchar(45)	latin1_swedish_ci		Да	NULL
<input type="checkbox"/>	3 primary_name	longtext	latin1_swedish_ci		Да	NULL
<input type="checkbox"/>	4 all_authors	longtext	latin1_swedish_ci		Да	NULL
<input type="checkbox"/>	5 secondary_name	longtext	latin1_swedish_ci		Да	NULL
<input type="checkbox"/>	6 year_city_country	longtext	latin1_swedish_ci		Да	NULL
<input type="checkbox"/>	7 year	int(11)			Да	NULL
<input type="checkbox"/>	8 publisher	longtext	latin1_swedish_ci		Да	NULL
<input type="checkbox"/>	9 city	varchar(45)	latin1_swedish_ci		Да	NULL
<input type="checkbox"/>	10 pages	varchar(45)	latin1_swedish_ci		Да	NULL
<input type="checkbox"/>	11 serie_number	int(11)			Да	NULL
<input type="checkbox"/>	12 text	longtext	latin1_swedish_ci		Да	NULL
<input type="checkbox"/>	13 theses	longtext	latin1_swedish_ci		Да	NULL
<input type="checkbox"/>	14 doc	longtext	latin1_swedish_ci		Да	NULL

Рисунок 3.6 – Таблица сутності «publications» – Публікації

Інформація по окремих стовпцях таблиці:

- **id** – оригінальний номер запису в таблиці, який однозначно ідентифікує певну публікацію;
- **type** – поле таблиці, що вказує на тип публікації (книга, журнал, конференція);

- primary_name – основне ім'я публікації;
- all_authors – список всіх авторів через кому, першим у списку стоїть ПІБ головного автора;
- secondary_name – загальне ім'я публікації;
- year_city_country – спеціальне поле для публікацій типу конференція, містить дані про рік, місто, державу в якій проводилася конференція;
- year – рік видання публікації;
- publisher – видавництво, яке надрукувало публікацію;
- city – місто, де публікація була видана;
- pages – кількість сторінок або конкретна сторінка публікації;
- serie_number – номер серії;
- text – згенерований програмно текст з даними про публікацію у відповідному форматі згідно з ДСТУ ГОСТ 7.1:2006;
- theses – поле таблиці, у якому зберігається шлях до файлу з тезами, який був загрузений на сервер користувачем;
- doc – поле таблиці, у якому зберігається шлях до файлу з повним текстом публікації, який був загрузений на сервер користувачем

3.3.4 Розробка таблиці зв'язку «Автор_Публікація»

Таблиця «Автор_Публікація» однозначно визначає, хто є автором якої публікації. Дану таблицю користувач БД заповнює у процесі додавання нової публікації. Таблиця виступає не як таблиця сутності, а як таблиця зв'язку між іншими таблицями. На рис. 3.7 зображено таблицю "auth_publ" («Автор_Публікація»), що містить усі поля таблиці.


#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/> 1	id 	int(11)			Нет	Нет
<input type="checkbox"/> 2	author_id	int(11)			Да	NULL
<input type="checkbox"/> 3	publication_id	int(11)			Да	NULL

Рисунок 3.7 – Таблица зв'язку «auth_publ» – Автор_Публікація

Інформація по окремих стовпцях таблиці:

- **id** – оригінальний номер запису в таблиці, який однозначно ідентифікує певну публікацію;
- **author_id** – поле, що вказує на автора публікації. Це значення міститься у відповідній таблиці сутності;
- **publication_id** – поле, що вказує на публікацію. Це значення міститься у відповідній таблиці сутності.

3.4 Встановлення Yii на локальний сервер

Спочатку потрібно скачати Yii з сайту <http://www.yiiframework.com> та розпакувати його в папку сайту на локальному сервері (C:\WebSever\OpenServer\domains\methodical). Етапи встановлення Yii на локальний сервер :

1. Запускаємо Командний рядок
2. Вводимо команду: `cd C:\`
3. Вводимо команду: `WebSever\OpenServer\modules\php\PHP-5.4\php.exe C:\WebSever\OpenServer\domains\methodical\framework\yiiic webapp C:\WebSever\OpenServer\domains\methodical\`

4. Далі відобразяться, повідомлення про те чи бажаємо створити додаток.

Вводимо : yes

Отримано такий результат:



```
Командная строка
Microsoft Windows [Version 10.0.10240]
(c) Корпорация Майкрософт (Microsoft Corporation), 2015 г. Все права защищены.

C:\Users\MSHQA>cd C:\
C:\>cd C:\WebSever\OpenServer\modules\php\PHP-5.4\php.exe C:\WebSever\OpenServer\domains\GuestBook\framework\yiiic webapp C:\WebSever\OpenServer\domains\GuestBook\
create a web application under 'C:\WebSever\OpenServer\domains\GuestBook' ? (yes/no)
no)
no) yes
mkdir C:\WebSever\OpenServer\domains\GuestBook\assets
mkdir C:\WebSever\OpenServer\domains\GuestBook\css
generate css/bg.gif
generate css/form.css
generate css/ie.css
generate css/main.css
generate css/print.css
generate css/screen.css
mkdir C:\WebSever\OpenServer\domains\GuestBook\images
generate index-test.php
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected
generate protected/.htaccess
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\commands
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\commands\shell
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\components
generate protected\components\Controller.php
generate protected\components\UserIdentity.php
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\config
generate protected\config/console.php
generate protected\config/database.php
generate protected\config/main.php
generate protected\config/test.php
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\controllers
generate protected\controllers\SiteController.php
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\data
generate protected\data/schema.mysql.sql
generate protected\data/schema.sqlite.sql
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\extensions
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\messages
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\migrations
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\models
generate protected\models\ContactForm.php
generate protected\models\LoginForm.php
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\runtime
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\tests
generate protected\tests\bootstrap.php
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\tests\fixtures
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\tests\functional
generate protected\tests\functional\siteTest.php
generate protected\tests\phpunit.xml
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\tests\report
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\tests\unit
generate protected\tests\WebTestCase.php
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\vendor
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\views
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\views\layouts
generate protected\views\layouts\column1.php
generate protected\views\layouts\column2.php
generate protected\views\layouts\main.php
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\views\site
generate protected\views\site\contact.php
generate protected\views\site\error.php
generate protected\views\site\index.php
generate protected\views\site\login.php
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\views\site\pages
generate protected\views\site\pages\about.php
generate protected\yiiic
generate protected\yiiic.bat
generate protected\yiiic.php
mkdir C:\WebSever\OpenServer\domains\GuestBook\themes
mkdir C:\WebSever\OpenServer\domains\GuestBook\themes\classic
mkdir C:\WebSever\OpenServer\domains\GuestBook\themes\classic\views
generate themes\classic\views\.htaccess
mkdir C:\WebSever\OpenServer\domains\GuestBook\themes\classic\views\layout
mkdir C:\WebSever\OpenServer\domains\GuestBook\themes\classic\views\site
mkdir C:\WebSever\OpenServer\domains\GuestBook\themes\classic\views\system

Your application has been created successfully under C:\WebSever\OpenServer\domains\GuestBook.
```

Рисунок 3.8 - Успішне встановлення Yii

5. Для перевірки того , чи дійсно створений додаток, потрібно ввести в браузері назву папки, в яку розпакований фреймворк : localhost/methodical Отриманий результат:

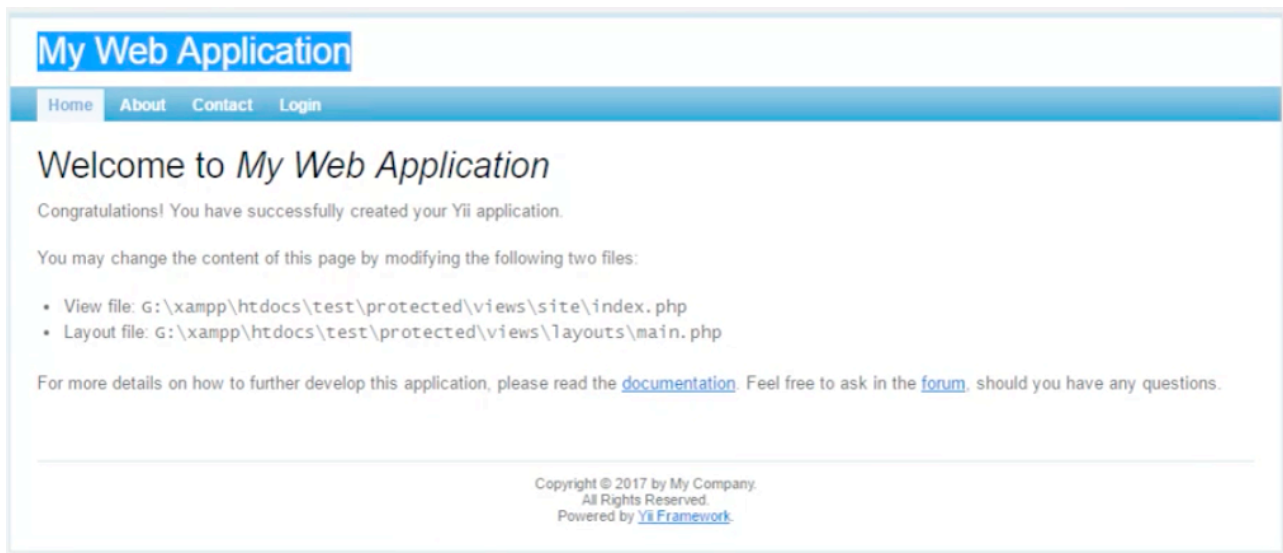


Рисунок 3.9 - Стандартно-згенерований додаток Yii .

3.5 Налаштування Yii

3.5.1 Налаштування з'єднання з базою даних

Для створення свого проекту потрібно налаштувати Yii для роботи з базою даних, яка була створена раніше. Для цього потрібно відредагувати файл `database.php` (, розміщений за адресою: `C:\WebSever\OpenServer\domains\methodical\protected\config\database.php`).

Для цього потрібно додати наступний код:

```
<?php
return array(
    'connectionString'=>'mysql:host=localhost;dbname=mehodical,
    'emulatePrepare' => true,
    'username' => 'root',
    'password' => "",
    'charset' => 'utf8', );
```

?>

Тепер проект використовує створену раніше базу даних під назвою methodical.

3.5.2 Активація кодогенератора Gii

Фреймворк доступний разом із вбудованим генератором коду Gii. Gii дозволяє генерувати моделі на основі таблиць в базі даних, а так само CRUD-контролери для основних дій з управління записами, такими як додавання запису (Create), перегляд запису (Read), редагування запису (Update) і видалення (Delete). Для активації Gii в файл конфігурації програми було додано опис підключення модуля gii:

```
'Gii' => array (  
  
    'Class' => 'system.gii.GiiModule',  
  
    'Password' => 'generate',  
  
    // Дозволені ip-адреси  
  
    'IpFilters' => array ( '127.0.0.1', ':: 1'),  
  
),
```

Для того щоб зайти в Gii необхідно перейти за наступним посиланням в браузері: <http://methodical/index.php?r=gii/>. Після авторизації буде доступний простий і зручний інтерфейс для генерації коду.

3.5.3 Встановлення доповнення Yiiastrap

Також для створення адаптивного дизайну потрібно завантажити та встановити доповнення Yiistrap2. При встановленні є такі етапи:

1. Створити папку vendor в корньовому каталозі сайту.
3. Розархівувати завантажений архів з доповненням в папку vendor.
4. Додати в файл main.php(шлях до C:\WebSever\OpenServer\domains\GuestBook\protected\config\main.php) наступний код:

```
'import'=>array(
    'bootstrap.helpers.*',
    'bootstrap.behaviors.*',
    'bootstrap.components.*',
    'bootstrap.form.*',
    'bootstrap.widgets.*',
),
'components'=>array(
    'bootstrap' => array(
        'class' => 'bootstrap.components.TbApi',
    ),
```

3.5.4 Налаштування системи доступу на основі ролей

В додатку буде використовуватися система доступу на основі ролей. Для її використання потрібно створити такі файли:

1. PhpAuthManager.php (шлях до файлу : C:\WebSever\OpenServer\domains\methodical\protected\components\Php

- AuthManager.php);
2. UserIdentity.php (шлях до файлу :
C:\WebSever\OpenServer\domains\methodical\protected\components\UserIdentity.php);
 5. WebUser.php (шлях до файлу :
C:\WebSever\OpenServer\domains\methodical\protected\components\WebUser.php);
 6. auth.php (шлях до файлу :
C:\WebSever\OpenServer\domains\methodical\protected\config\auth.php) .

Також потрібно додати в файл main.php наступний код:

```
'components'=>array(
    'user' => array(
        'class' => 'WebUser',
    ),
    'authManager' => array(
        'class' => 'PhpAuthManager',
        'defaultRoles' => array('guest'),
    ),
),
```

На цьому налаштування Yii завершено.

3.6 Реалізація архітектури MVC сайту

Сайт буде мати декілька роздів:

- Головна;
- FAQ (часті запитання) ;
- Про сайт (інформація про структуру) ;
- Вхід (сторінка створена для входу на сайт адміністратора та автора

(доступна за адресою: адреса_проекту/ index.php?r=site/login).

Оскільки основним розділом сайту є розділ “Головна”, на якій розміщені методичні посібники на прикладі його буде показано реалізацію архітектури MVC. Решту розділів розглядати не доцільно, адже вони будуть створюватися аналогічно.

3.6.1 Генерація моделей сайту

Для створення моделей, необхідних для сайту, скористаємось кодогенератором Gii. Щоб зайти в Gii, необхідно перейти за наступним посиланням в браузері: `http://methodical/index.php?r=gii` і пройти авторизацію. Згенеруємо модель для таблиці Post. Для цього в Gii потрібно обрати Model Generator та у поле Table Name ввести назву таблиці, в даному випадку Post. Після цього отримано такий результат:

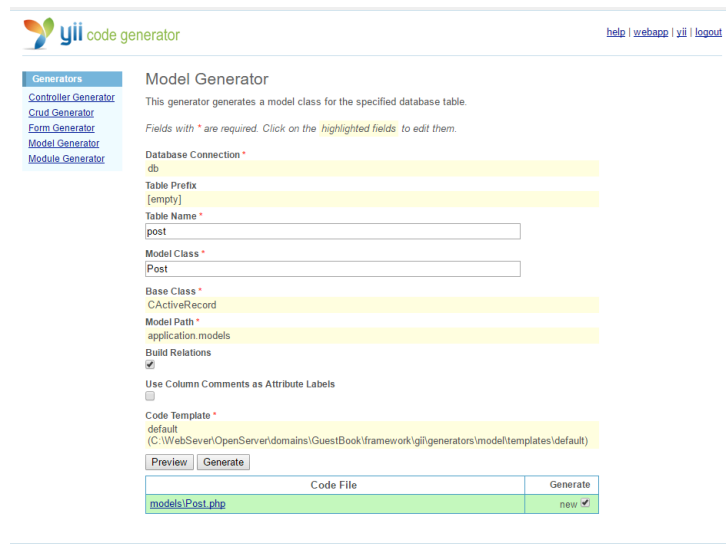



Рисунок 3.10 - Генерація моделі Post.

Далі потрібно натиснути кнопку “Generate”. І у випадку успішної

генерації моделі буде виведено повідомлення:



The code has been generated successfully.

Рисунок 3.11 - Успішно згенерована модель.

Після цих дій у папці `models` буде згенеровано файл `Post.php`, код якого наведено в додатку. Аналогічно згенеруємо моделі для решти таблиць.

Також у папці розміщений стандартно згенерований файл моделі `LoginForm.php`, який використовується для валідації форми авторизації.

Згенеровані за допомогою Gii моделі містять правила валідації полів таблиці, функції для пошуку по таблиці та звязки між таблицями.

3.6.2 Генерація контролерів та представлень сайту

Для генерації контролерів з набором функцій створення, читання, редагування та видалення (CRUD) використовується `CRUD Generator` в Gii. Так він автоматично генерує представлення для кожної з операцій.

Для генерації `CRUD` контролера для моделі `Post` потрібно в Gii вибрати `CRUD Generator` та ввести назву моделі, для якої буде згенеровано контролер та відповідні представлення:

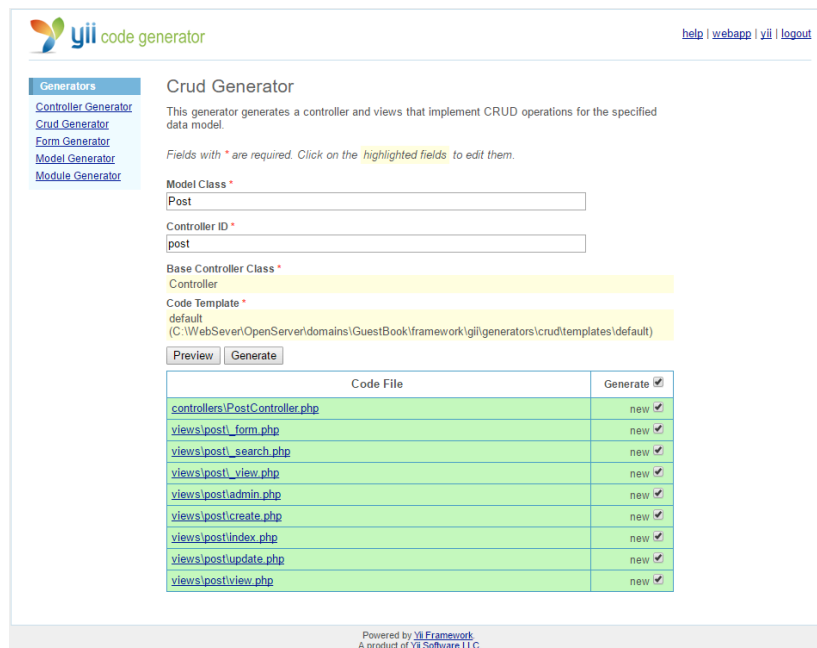


Рисунок 3.12 - Генерація CRUD контролера для моделі Post.

Далі потрібно натиснути кнопку “Generate”. І у випадку успішної генерації контролера буде виведено повідомлення:

The controller has been generated successfully. You may [try it now](#).

Рисунок 3.13 - Успішно згенерований контролер.

Аналогічно згенеруємо контролери і представлення для решти моделей.

Після виконання цих етапів було успішно згенеровано каркас для сайту “Бази методичних посібників”. Звісно, він ще потребує допрацювання, але основа для створення сайту вже закладена.

На даному етапі сайт потребує масштабного допрацювання, а також те,

що він не відповідає макету, який був спроектований раніше. Далі буде проведено редагування контролерів та предствлень для відповідності дизайну сайту створеному раніше макету. Для створення адаптивної сітки сайту та основної частини оформлення буде використано доповнення Yiistrap(Bootstrap для Yii).

3.6.3 Зміна дизайну header та footer сайту

Header та footer сайту зберігаються в окремому файлі в папці `views/layouts/main.php` , вони стандартно використовуються на всіх сторінках сайту. Для зміни їх дизайну було використано стандартні класи Yiistrap та CSS . Після допрацювання header та footer сайту , набули бажаного вигляду:

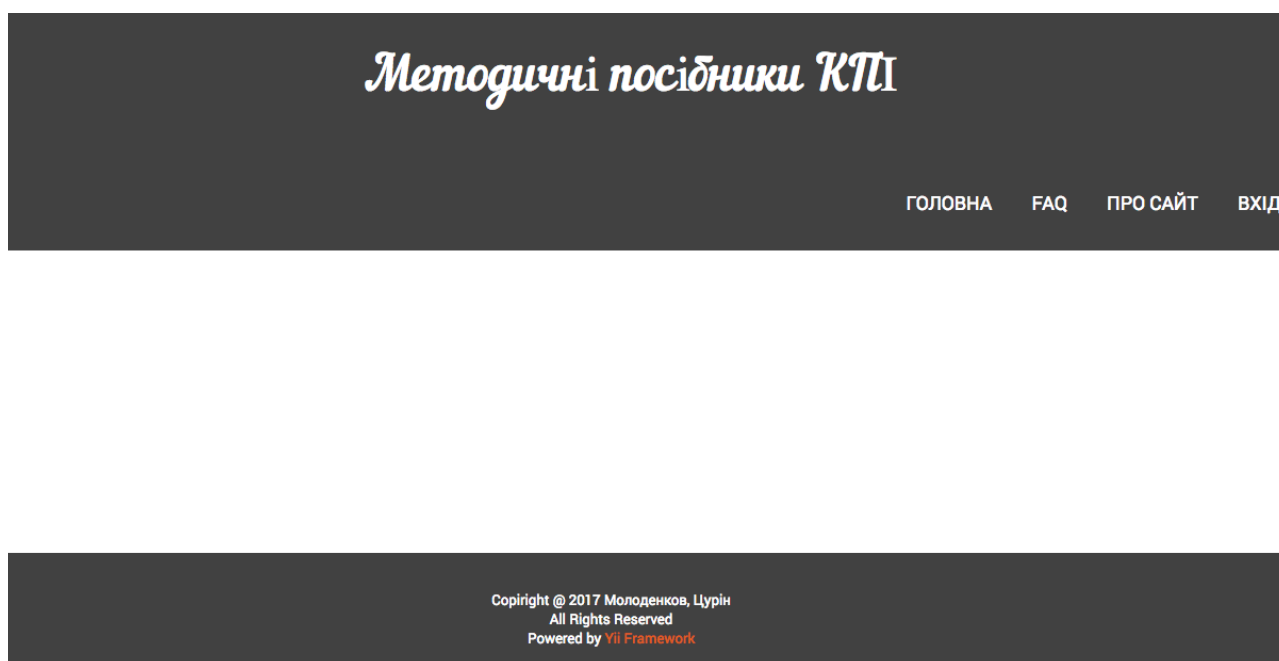


Рисунок 3.14 - Header та footer сайту після допрацювання.

3.6.4 Зміна дизайну контентної частини сайту

Допрацювання дизайну частини сайту, що містить контент, розглянемо на прикладі сторінки “Головна”. На першому етапі залишимо в папках для представлень ті представлення, які потрібні для сайту, що створюється. Папка з представленнями для сторінки “Головна” після редагування:

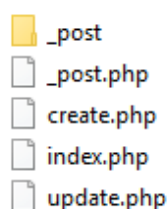


Рисунок 3.15 - Папка з представленнями для моделі Post після допрацювання.

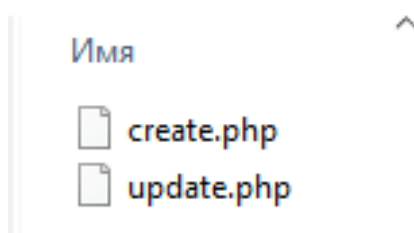


Рисунок 3.16 - Папка з представленнями для моделі Comment після допрацювання.

Як видно із зображень, в папці з представленнями для моделі Comment після допрацювання залишились тільки представлення для сторінок створення та редагування методичних посібників. Так зроблено тому, що посібники будуть відображатися в представленні `_views.php` для моделі Post .

Далі потрібне редагування коду всіх представлень та контролерів . Для оформлення використовувались стандартні класи Yiistrap та CSS. Стандартно згенерований код представлень був майже повністю переписаний.

Відредагований код наведено у додатку.

В результаті отримано такий дизайн сайту:

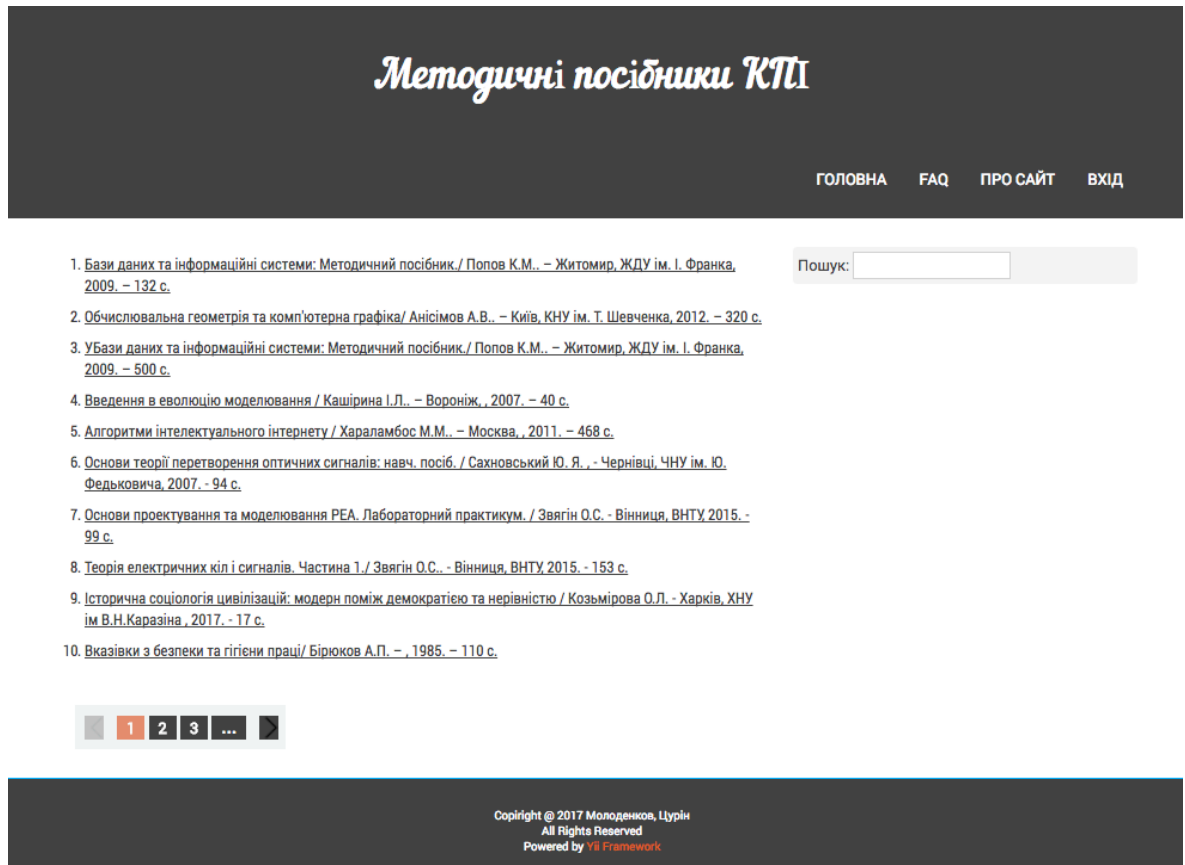


Рисунок 3.17 - Сторінка "Головна".

3.7 Висновки до розділу 3

Як завершальний етап у роботі з фреймворками, було створено web-сайт за допомогою Yii, для якого характерні усі основні особливості сайту, побудованого за допомогою обраного Frameworkу. Він має тривірневу структуру Model-View-Controller , а всі дані , що містяться на сайті, зберігаються в базі даних.

Для удосконалення інтерфейсу було використано доповнення YiiStrap, яке є аналогом фреймворку Bootstrap для Front-End розробки.

Після роботи над розробкою сайту можна сказати , що WEB-фреймворк обрано дуже вдало, адже він допомагає створювати багатofункціональні сайти за короткий період.

4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

У даному розділі проводиться оцінка основних характеристик програмного продукту, призначеного для створення сайту «Бази методичних посібників». Сайт буде створюватися за допомогою WEB-фреймворку .

Фінальний сайт призначений для додавання методичних посібників до захищеної бази. Користувачі можуть додавати, переглядати та знаходити методичні посібники. Сайт повинен однаково добре відображатися у браузерях, таких як Google Chrome, Mozilla Firefox, Opera та інші.

Нижче наведено аналіз різних варіантів реалізації модуля з метою вибору оптимальної, з огляду при цьому як на економічні фактори, так і на характеристики продукту, що впливають на продуктивність роботи і на його сумісність з апаратним забезпеченням. Для цього було використано апарат функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) – це технологія, яка дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. Як прямі, так і побічні витрати розподіляються по продуктах та послугах у залежності від потрібних на кожному етапі виробництва обсягів ресурсів. Виконані на цих етапах дії у контексті методу ФВА називаються функціями.

Мета ФВА полягає у забезпеченні правильного розподілу ресурсів, виділених на виробництво продукції або надання послуг, на прямі та непрямі витрати. У даному випадку – аналізу функцій програмного продукту й виявлення усіх витрат на реалізацію цих функцій.

Фактично цей метод працює за таким алгоритмом:

- визначається послідовність функцій, необхідних для виробництва продукту. Спочатку – всі можливі, потім вони розподіляються по двом групам: ті, що впливають на вартість продукту і ті, що не впливають. На цьому ж етапі оптимізується сама послідовність скороченням кроків, що не впливають на цінність і відповідно витрат.
- для кожної функції визначаються повні річні витрати й кількість робочих часів.
- для кожної функції на основі оцінок попереднього пункту визначається кількісна характеристика джерел витрат.
- після того, як для кожної функції будуть визначені їх джерела витрат, проводиться кінцевий розрахунок витрат на виробництво продукту.

4.1 Постановка задачі техніко-економічного аналізу

У роботі застосовується метод ФВА для проведення техніко-економічний аналізу розробки системи аналізу нелінійних нестационарних процесів. Оскільки основні проектні рішення стосуються всієї системи, кожна окрема підсистема має їм задовольняти.

Відповідно цьому варто обирати і систему показників якості програмного продукту.

Технічні вимоги до продукту наступні:

- програмний продукт повинен функціонувати на персональних комп'ютерах із стандартним набором компонентів;
- забезпечувати цілодобову можливість користувачам сайту переглядати методичні посібники
- забезпечувати зручність і простоту взаємодії з користувачем або з адміністратором програмного забезпечення;
- передбачати мінімальні витрати на впровадження програмного продукту.

4.1.1 Обґрунтування функцій програмного продукту

Головна функція F_0 – обрання програмного продукту для встановлення на фізичний сервер КПІ-Телеком. Виходячи з конкретної мети, можна виділити наступні основні функції ПП:

F_1 – вибір WEB-фреймворку;

F_2 – вибір типу підключення до бази даних;

F_3 – вибір серверу на який буде встановлюватися WEB-фреймворк.

Кожна з основних функцій може мати декілька варіантів реалізації.

Функція F_1 :

а) WEB-фреймворк Yii;

б) WEB-фреймворку Symfony;

Функція F_2 :

а) підключення MySQLi.

б) підключення MySQL;

Функція F_3 :

а) сервер PHP Apache ;

б) сервер PHP Denwer.

4.1.2 Варіанти реалізації основних функцій

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 4.1). На основі цієї карти побудовано позитивно-негативну матрицю варіантів основних функцій (таблиця 4.1).

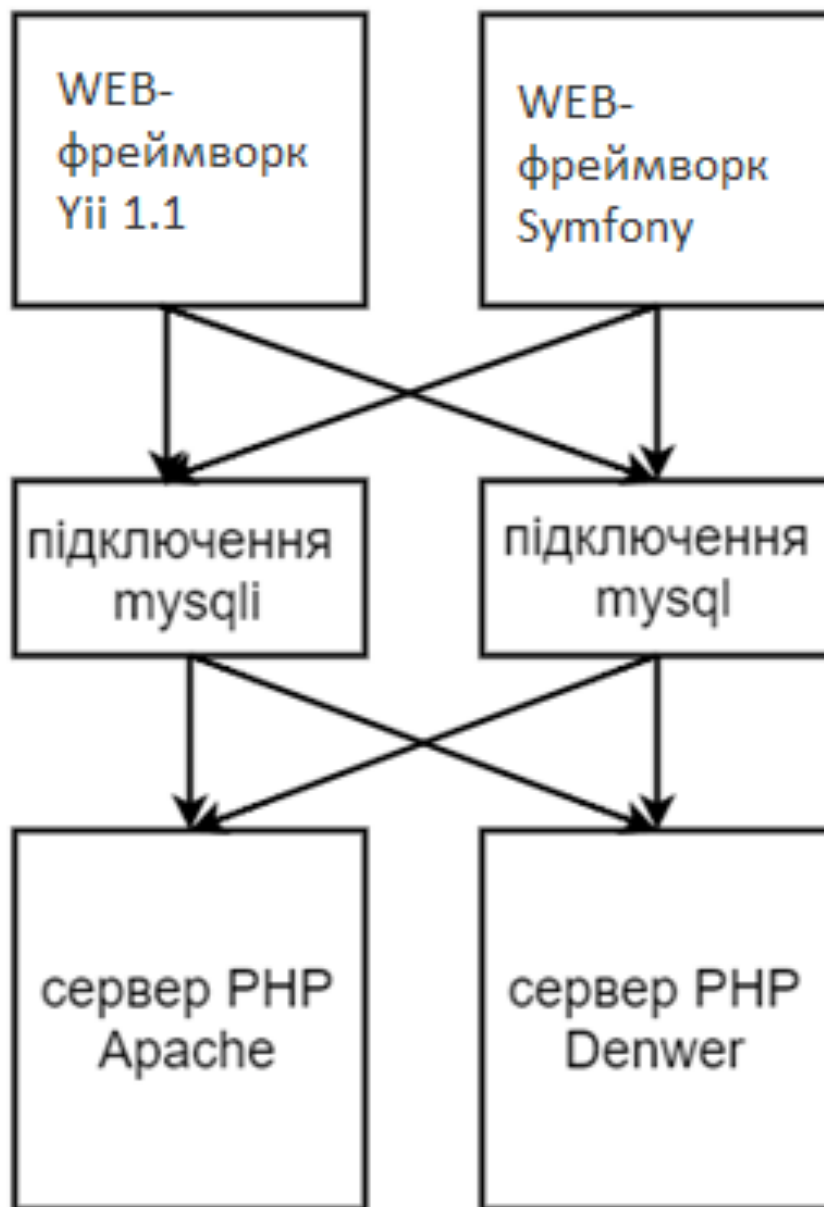


Рисунок 4.1 – Морфологічна карта

Морфологічна карта відображує всі можливі комбінації варіантів реалізації функцій, які складають повну множину варіантів ПП.

Таблиця 4.1 – Позитивно-негативна матриця

Осно вні функції	Варіан ти реалізації	Переваги	Недоліки
F1	А	Введення і валідація форм ,аутифікація і авторизація, використання AJAX і інтеграція з jQuery, генерація базового PHP-коду для CRUD- операцій (скаффолдинг), підтримка тем оформлення для їх легкої зміни, можливість підключення сторонніх бібліотек, міграції бази даних.	Не дуже потужний роутинг і наявність великого кількості віджетів, на настройку яких витрачається багато часу.
	Б	Підтримує безліч баз даних (MySQL, PostgreSQL, SQLite, або будь- яка інша PDO-сумісна СУБД) , вбудовані класи для роботи з email, гнучка система шаблонів у поданні, підтримка французького спонсора Sensio	Складний в освоєнні, підходить тільки для великих проектів, відсутність російської документації.
F2	А	Більша оптимізація ніж у аналога, більша швидкість доступу.	Неможливе з'єднання з старими версіями баз даних.
	Б	Підтримка старих версій MySQL баз даних та PHP.	Менша швидкість.
F3	А	Можливість використання різних версій PHP.	PHP та MySQL необхідно встановлювати додатково.
	Б	Повна комплектація з PHP та MySQL.	Неможливість використання інших версій PHP та MySQL

На основі аналізу позитивно-негативної матриці робимо висновок, що при обранні програмного продукту деякі варіанти реалізації функцій варто відкинути, тому що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція F1:

Обидва WEB-фреймворки доволі розвинені та мають багато розширень та доповнень, тому обидва варіанти А і Б гідні розгляду.

Функція F2:

У роботі нам важлива підтримка старих версій PHP, тому обираємо варіант Б.

Функція F3:

Так як обрані фреймворки використовують різні версії PHP для своєї роботи, необхідно обрати сервер Apache для більш тонкого налаштування серверу. Отже варіант А.

Таким чином, будемо розглядати такі варіанти реалізації ПП:

F1a – F2б – F3a

F1б – F2б – F3a

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

4.2 Обґрунтування системи параметрів ПП

4.2.1 Опис параметрів

На підставі даних про основні функції, що повинен реалізувати програмний продукт, вимог до нього, визначаються основні параметри виробу, що будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

- $X1$ – час відповіді на запит клієнта;
- $X2$ – кількість модулів на сайті;
- $X3$ – об'єм даних, що передається через Інтернет;
- $X4$ – об'єм оперативної пам'яті, що використовується.

$X1$: Відображає час, що необхідний серверу для обробки запиту клієнту та відправлення відповіді.

$X2$: Відображає кількість модулів, що будуть використані для створення сайту.

$X3$: Відображає розмір даних, що передаються клієнту через мережу Інтернет.

$X4$: Показує кількість оперативної пам'яті, що необхідні для стабільної роботи серверу.

4.2.2. Кількісна оцінка параметрів

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію ПП як показано у табл. 4.2.

Таблиця 4.2 – Основні параметри ПП

Назва Параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	серед ні	кращі
Час відповіді на запит клієнта	X1	с	10	4	2
Кількість модулів на сайті	X2	Кількість модулів	3	5	10
Об'єм даних, що передається через Інтернет	X3	Кб	2000	800	500
Об'єм оперативної пам'яті, що використовується	X4	Мб	150	100	70

За даними таблиці 4.2 будуються графічні характеристики параметрів –
рис. 4.2 – рис. 4.5

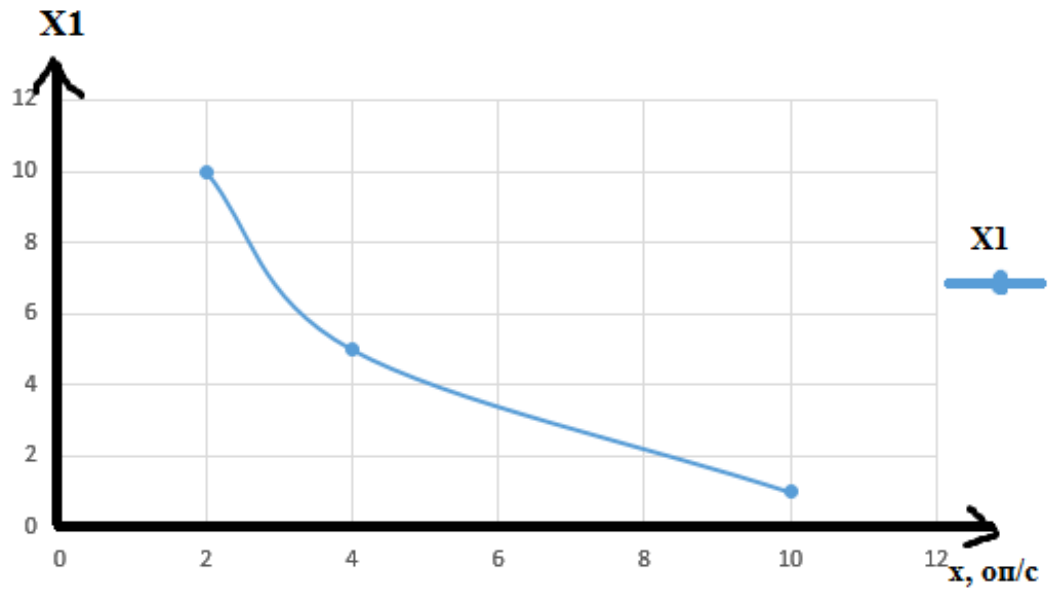


Рисунок 4.2 – $X1$, завантаженість підключення до Інтернету

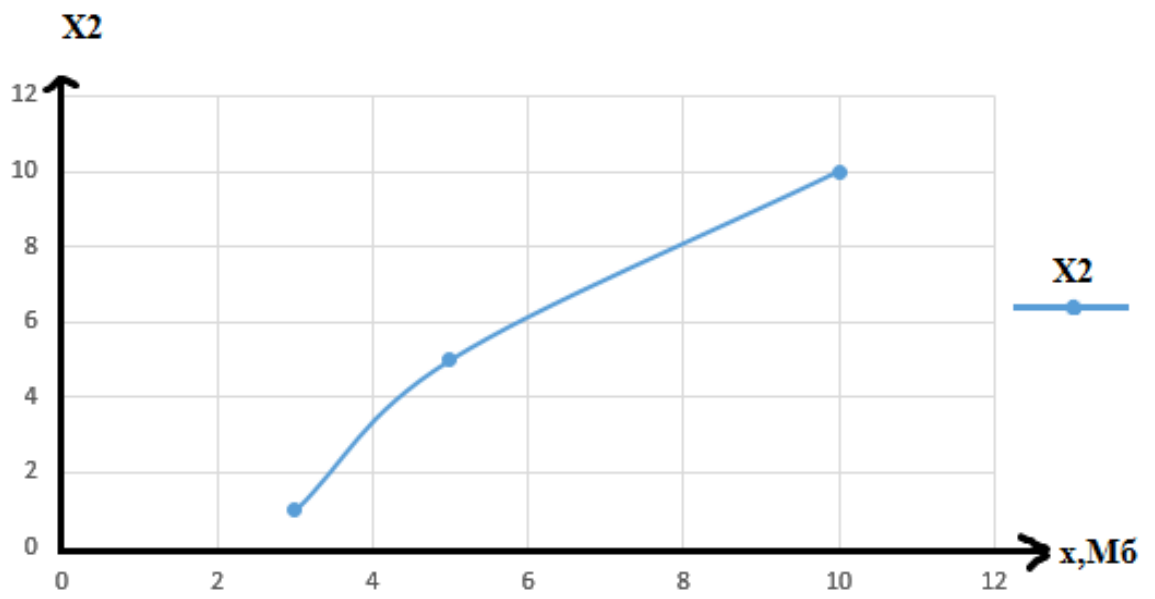


Рисунок 4.3 – $X2$, об'єм оперативної пам'яті, що використовується

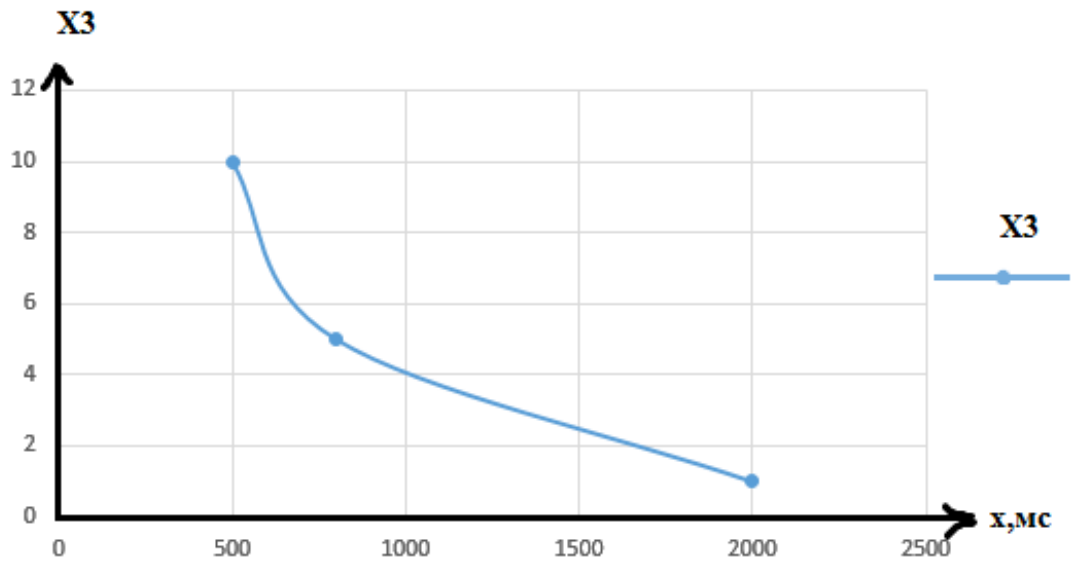


Рисунок 4.4 – X3, час необхідний на підключення до медіа серверу

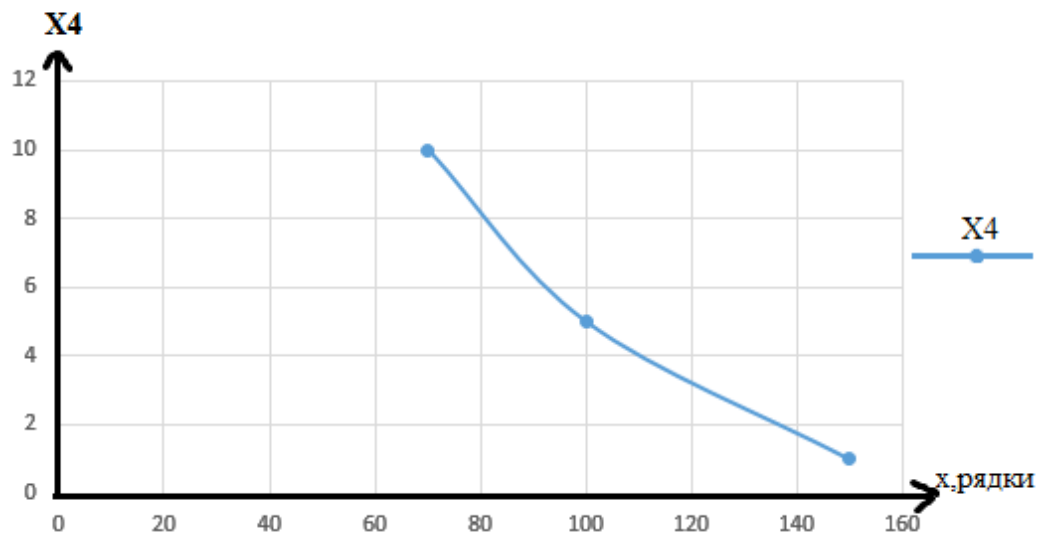


Рисунок 4.5 – X4, кількість функцій, доступних для віддаленого доступу

4.2.3 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – вибір програмного продукту, який має найбільш зручний та функціональний набір можливостей при низькій необхідності у ресурсах.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 4.3

Таблиця 4.3 – Результати ранжування параметрів

Позначення параметра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта								Сума рангів R_i	Відхилення Δ_i	Δ_i^2
Час відповіді на запит клієнта	X1	С									9	-8,5	72,25
Кількість модулів на сайті	X2	Кількість модулів									25	7,5	56,25
Об'єм даних, що передається через Інтернет	X3	Кб									13	-4,5	20,25
Об'єм оперативної пам'яті, що використовується	X4	Мб									23	5,5	30,25
Разом			0	0	0	0	0	0	0		70	0	179

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70,$$

де N – число експертів, n – кількість параметрів;

б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 17,5.$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T$$

Сума відхилень по всім параметрам повинна дорівнювати 0;

г) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^N \Delta_i^2 = 179.$$

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3-n)} = \frac{12 \cdot 179}{7^2(4^3-4)} = 0,73 > W_k = 0,67$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 4.4.

Таблиця 4.4 – Попарне порівняння параметрів

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	>	>	>	>	>	>	>	>	1,5
X1 і X3	>	<	>	<	>	>	>	>	1,5
X1 і X4	>	>	>	>	>	>	>	>	1,5
X2 і X3	<	<	<	<	<	<	<	<	0,5
X2 і X4	<	>	<	>	<	<	>	<	0,5
X3 і X4	>	>	>	>	>	<	>	>	1,5

Числове значення, що визначає ступінь переваги i -го параметра над j -тим, a_{ij} визначається по формулі:

$$a_{ij} = \begin{cases} 1.5 & \text{при } X_i > X_j \\ 1.0 & \text{при } X_i = X_j \\ 0.5 & \text{при } X_i < X_j \end{cases}$$

З отриманих числових оцінок переваги складемо матрицю $A = \| a_{ij} \|$.

Для кожного параметра зробимо розрахунок вагомості K_{vi} за наступними формулами:

$$K_{vi} = \frac{b_i}{\sum_{i=1}^n b_i}, \text{ де } b_i = \sum_{j=1}^N a_{ij}.$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{vi} = \frac{b'_i}{\sum_{i=1}^n b'_i}, \text{ де } b'_i = \sum_{j=1}^N a_{ij} b_j.$$

Як видно з таблиці 4.5, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 4.5 – Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j				Перша ітер.		Друга ітер.		Третя ітер	
	X1	X2	X3	X4	b_i	K_{Bi}	b_i^1	K_{Bi}^1	b_i^2	K_{Bi}^2
	1	1	1	1						
	,0	,5	,5	,5	5,5	0,3 44	21, 25	0,3 60	77, 875	0,3 61
X2	0	1	0	0	2,5	0,1 56	9,2 5	0,1 57	34, 125	0,1 58
X3	0	1	1	1	4,5	0,2 81	16, 25	0,2 75	59, 125	0,2 74
X4	0	1	0	1	3,5	0,2 19	12, 25	0,2 08	44, 875	0,2 07
Всього:					16	1	59	1	216	1

4.3 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів $X1$ (час відповіді на запит клієнта) та $X4$ (Об'єм оперативної пам'яті, що використовується) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра $X2$ (Кількість модулів на сайті) буде найкращим у випадку обрання у F2 варіанта Б і становитиме 5, для варіанту А це значення буде 8.

Абсолютне значення параметра $X3$ (Об'єм даних, що передається через Інтернет) буде найкраще при обрані варіанту А 600, а при обрані варіанту Б

воно буде середнім 1500.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (таблиця 5.6):

$$K_K(j) = \sum_{i=1}^n K_{ei,j} B_{i,j},$$

де n – кількість параметрів; K_{ei} – коефіцієнт вагомості i -го параметра; B_i – оцінка i -го параметра в балах.

Таблиця 4.6 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Осн овні функції	Варіа нт реалізаці ї функції	Па ра ме три	Абсолю тне значення параметра	Бальна оцінка параметр а	Коефіці єнт вагомості параметра	Коефіці єнт рівня якості
F1	А	X2	5	5	0,158	0,79
F2	Б	X3	600	7	0,274	1,918
F3	Б	X2	8	8	0,158	1,264
		X3	1500	2,5	0,274	0,685
	А	X1	3	6,5	0,361	2,35
		X4	100	5	0,207	1,035

За даними з таблиці 4.6 за формулою

$$K_K = K_{TY}[F_{1k}] + K_{TY}[F_{2k}] + \dots + K_{TY}[F_{zk}],$$

визначаємо рівень якості кожного з варіантів:

$$K_{Kl} = 0,79 + 1,918 + 2,35 + 1,035 = 6,093$$

$$K_{K2} = 1,264 + 0,685 + 2,35 + 1,035 = 5,334$$

Як видно з розрахунків, кращим є перший варіант, для якого коефіцієнт технічного рівня має найбільше значення.

4.4 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Обидва варіанти включають в себе два окремих завдання:

1. Розробка проекту програмного продукту;
2. Програмна реалізація рішення;

Обидва завдання за ступенем новизни відносяться до групи В. За складністю алгоритми, які використовуються в завданні 1 належать до групи 1; а в завданні 2 – до групи 3.

Для реалізації завдання 1 використовується нормативно-довідкова інформація, а завдання 2 використовує банк даних.

Проведемо розрахунок норм часу для виконання кожного з завдань. Загальна трудомісткість обчислюється як:

$$T_0 = T_r \cdot K_{\text{П}} \cdot K_{\text{СК}} \cdot K_{\text{М}} \cdot K_{\text{СТ}} \cdot K_{\text{СТ.М}}, \quad (4.11)$$

де T_r – трудомісткість розробки ПП; $K_{\text{П}}$ – поправочний коефіцієнт; $K_{\text{СК}}$ – коефіцієнт на складність вхідної інформації; $K_{\text{М}}$ – коефіцієнт рівня мови програмування; $K_{\text{СТ}}$ – коефіцієнт використання стандартних модулів і прикладних програм; $K_{\text{СТ.М}}$ – коефіцієнт стандартного математичного забезпечення

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру ступеня новизни В та групи складності алгоритму 1, трудомісткість дорівнює: $T_p = 43$ людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання: $K_{PI} = 0.81$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для обох завдань рівний 1: $K_{СК} = 1$. Оскільки при розробці кожного з завдань використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $K_{СТ} = 0.8$. Тоді, за формулою 4.11, загальна трудомісткість першого завдання дорівнює:

$$T_1 = 43 \cdot 0.81 \cdot 0.8 = 27,864 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для другого завдання.

Для другого завдання (використовується алгоритм третьої групи складності, степінь новизни В), тобто $T_p = 12$ людино-днів, $K_{PI} = 0.5$, $K_M = 1$, $K_{СК} = 1$, $K_{СТ} = 0.8$, $K_{СТ.M} = 1.6$ (однакові для обох варіантів реалізації):

$$T_2 = 12 \cdot 0.5 \cdot 0.8 \cdot 1.6 = 7.68 \text{ людино-днів.}$$

Складаємо трудомісткість відповідних завдань для отримання загальної трудомісткості реалізації ПП, яка буде однаковою для обох варіантів тому, що обидва фреймворки використовують мову програмування РНР і їх вибір впливає лише на зручність роботи розробника, а не на вартість кінцевого ПП:

$$T = (27.864 + 7.68) \cdot 8 = 284.352 \text{ людино-годин;}$$

В розробці бере участь один програміст з окладом 9000 грн. Визначимо зарплату за годину за формулою:

$$C_{\text{ч}} = \frac{M}{T_m \cdot t} \text{ грн.,} \quad (4.12)$$

де M – місячний оклад працівників; T_m – кількість робочих днів тиждень; t – кількість робочих годин в день.

$$C_q = \frac{9000}{21 \cdot 8} = 53,57 \text{ грн.}$$

Тоді, розрахуємо заробітну плату за формулою:

$$C_{ЗП} = C_q \cdot T_i \cdot K_d, \quad (4.13)$$

де C_q – величина погодинної оплати праці програміста; T_i – трудомісткість відповідного завдання; K_d – норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

$$C_{ЗП} = 53.57 \cdot 284.352 \cdot 1.2 = 18279.28 \text{ грн.}$$

Відрахування на єдиний соціальний внесок незалежно від групи професійного ризику становить 22%:

$$СВІД = C_{ЗП} \cdot 0.22 = 18279.28 \cdot 0.22 = 4021.44 \text{ грн.}$$

Тепер визначимо витрати на оплату однієї машино-години. (C_M)

Так як одна ЕОМ обслуговує одного програміста з окладом 9000 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$$C_{Г} = 12 \cdot M \cdot K_3 = 12 \cdot 9000 \cdot 0,2 = 21600 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{ЗП} = C_{Г} \cdot (1 + K_3) = 21600 \cdot (1 + 0.2) = 25920 \text{ грн.}$$

Відрахування на єдиний соціальний внесок:

$$СВІД = C_{ЗП} \cdot 0.22 = 25920 \cdot 0,22 = 5702.4 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 12000 грн.

$$C_A = K_{TM} \cdot K_A \cdot ЦПР = 1.15 \cdot 0.25 \cdot 12000 = 3450 \text{ грн.}, \quad (4.18)$$

де K_{TM} – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача; K_A – річна норма амортизації; $ЦПР$ – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = K_{TM} \cdot ЦПР \cdot K_P = 1.15 \cdot 12000 \cdot 0.05 = 690 \text{ грн.}, \quad (4.19)$$

де K_P – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$T_{EF} = (D_K - D_B - D_C - D_P) \cdot t_3 \cdot K_B, \quad (4.20)$$

де D_K – календарна кількість днів у році; D_B , D_C – відповідно кількість вихідних та святкових днів; D_P – кількість днів планових ремонтів устаткування; t – кількість робочих годин в день; K_B – коефіцієнт використання приладу у часі протягом зміни.

$$T_{EF} = (365 - 104 - 8 - 16) \cdot 8 \cdot 0.9 = 1706.4 \text{ годин}$$

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{EL} = T_{EF} \cdot N_C \cdot ЦЕН = 1706,4 \cdot 0,5 \cdot 1,93819 = 1653,66 \text{ грн.}$$

, де N_C – середньо-споживча потужність приладу; $ЦЕН$ – тариф за 1 кВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_H = ЦПР \cdot 0.67 = 12000 \cdot 0,67 = 8040 \text{ грн.} \quad (4.22)$$

Тоді річні експлуатаційні витрати будуть:

$$\text{СЕКС} = \text{СЗП} + \text{СВІД} + \text{СА} + \text{СР} + \text{СЕЛ} + \text{СН}$$

(4.23)

$$\text{СЕКС} = 25920 + 5702.4 + 3450 + 690 + 1653,66 + 8040 = 45451,06$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$\text{СМ-Г} = \text{СЕКС} / \text{ТЕФ} = 45451.06 / 1706.4 = 26.63 \text{ грн/час.}$$

(4.24)

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту, ведуться на ЕОМ, витрати на оплату машинного часу складає:

$$\text{СМ} = \text{СМ-Г} \cdot \text{Т}$$

(4.25)

$$\text{СМ} = 26.63 * 284,352 = 7572.29 \text{ грн.};$$

Накладні витрати складають 67% від заробітної плати:

$$\text{СН} = \text{СЗП} \cdot 0,67$$

(4.26)

$$\text{СН} = 18279.28 * 0,67 = 12247.12 \text{ грн.};$$

Отже, вартість розробки ПП за варіантами становить:

$$\text{СПП} = \text{СЗП} + \text{СВІД} + \text{СМ} + \text{СН}$$

(4.27)

$$\text{СПП} = 18279.28 + 4021.44 + 7572.29 + 12247.12 = 42120,13 \text{ грн.};$$

4.5 Вибір кращого варіанта ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{TEPj} = K_{Kj} / C_{\Phi j}$$

$$K_{TEP1} = 6.093 / 42120,13 = 0,1446 \cdot 10^{-3};$$

$$K_{TEP2} = 5.334 / 42120,13 = 0,1266 \cdot 10^{-3};$$

Оскільки обидва варіанти однаково трудомісткі, найбільш ефективним є перший варіант реалізації програми (через більший рівень якості) з коефіцієнтом техніко-економічного рівня $K_{TEP1} = 0,1446 \cdot 10^{-3}$.

4.6 Висновки до розділу 4

В даному розділі було проведено функціонально-вартісний аналіз програмного продукту, що буде створено. Процес аналізу складався з двох частин.

У першій проведено дослідження програмного продукту з технічної точки зору: були поставлені основні параметри, що повинні бути головними при обранні кращої реалізації. На основі отриманих значень параметрів, оцінок експертів було обчислено коефіцієнт технічного рівня, який і дав змогу визначити оптимальну з технічної точки зору альтернативу реалізації функцій ПП.

У другій частині виконувалися обрахунки вартості розробки ПП з урахуванням витрат на заробітні плати, електроенергії, накладні витрати. Трудомісткість обох варіантів реалізації вважалась однаковою, адже різниця

полягала лише у виборі фреймворку (обидва фреймворки використовують мову програмування PHP).

Після виконання функціонально-вартісного аналізу програмного продукту що розроблюється, можна зробити висновок, що перший варіант є найбільш оптимальним для реалізації (через більше значення коефіцієнту технічного рівня). Його показник техніко-економічного рівня якості $KTEP_1 = 0,1446 \cdot 10^{-3}$;

Цей варіант реалізації програмного продукту має такі параметри:

- WEB-фреймворк Yii;
- Підключення MySQL;
- Сервер Apache.

Даний варіант реалізації є найбільш продуктивним за рахунок використання більш оптимального для проекту фреймворку та сервера, а також надає користувачу зручний інтерфейс, непоганий функціонал і швидкодію.

ВИСНОВКИ

Системи створення web-додатків або фреймворки активно використовуються розробниками при створенні web-додатків із різним функціоналом і рівнем складності. Аналіз основних характеристик і можливостей сучасних фреймворків дозволяє вибрати оптимальний варіант для конкретних web-додатків з урахуванням поставлених завдань. Вибір та використання фреймворків відіграє важливу роль при проектуванні, реалізації та супроводі як простих web-додатків, так і складних програмних комплексів.

Будь-хто може помилитися при виборі фреймворку, однак можна унебезпечитися від такого роду помилок. Для цього необхідно переконатися, що обраний фреймворк має достатню функціональність і хорошу підтримку, тому що зазвичай невеликі фреймворки створюються особами, знання яких в певній мірі обмежені. Це може викликати різні помилки і питання, які в кінцевому рахунку заважають і уповільнюють процес розробки.

Вибираючи фреймворк, необхідно звертати увагу на те, наскільки він легкий в освоєнні і розумінні. Це має дуже важливе значення для малодосвідченого програміста. Також необхідно переконатися, що база даних і web-сервер сумісні з архітектурою обраного фреймворку.

У разі недотримання вищевказаних вимог можливі падіння в продуктивності розробки проекту, продуктивності коду і легкості супроводу. Іншою поширеною помилкою є неправильна установка фреймворку. Тому під час установки необхідно слідувати інструкції, щоб уникнути помилок.

Після аналізу вимог сайту «Бази методичних посібників», фреймворк Yii було обрано для створення проекту. Вибір був здійснений згідно з критеріями які були оглянуті в розділі 1. Yii має багато можливостей які

виділяють цей фреймворк на фоні інших і роблять розробку з його допомогою дуже швидкою і приємною.

ПЕРЕЛІК ПОСИЛАНЬ

- 1 Інтернет технології [Електронний ресурс] – Режим доступу:
<https://uk.wikipedia.org/wiki/Інтернет-технології> - дата доступу:
15.03.2017
- 2 Веб-фреймворки и с чем их едят [Електронний ресурс] – Режим
доступу: <http://iwsn.ru/blog/show/veb-freymvorki-i-s-chem-ih-edyat> - дата
доступу: 17.03.2017
- 3 Фреймворки в веб-разработке – режим доступу: https://web-creator.ru/articles/about_frameworks - дата доступу: 16.05.2017
- 4 Фреймворки в веб-разработке [Електронний ресурс] – Режим доступу
до ресурсу: : https://web-creator.ru/articles/about_frameworks - дата
доступу: 20.03.2017
- 5 Результаты тестирования шести ведущих фреймворков на
производительность [Електронний ресурс] – Режим доступу до
ресурсу: [http://www.alrond.com/ru/2007/jan/25/rezultaty-testirovaniya-6-
frameworks](http://www.alrond.com/ru/2007/jan/25/rezultaty-testirovaniya-6-frameworks) - дата доступу: 27.03.2017
- 6 Порівняльний аналіз найпоширеніших фреймворків для PHP у сфері
веб-розробок – режим доступу:
[http://dspace.kntu.kr.ua/jspui/bitstream/123456789/5594/1/ilovepdf_com-
75-78.pdf](http://dspace.kntu.kr.ua/jspui/bitstream/123456789/5594/1/ilovepdf_com-75-78.pdf) - дата доступу: 11.03.2017
- 7 Використання PHP фреймворків в розробці сайту [Електронний ресурс]
– Режим доступу до ресурсу: [http://ukrbukva.net/page,5,39718-Ispol-
zovanie-PHP- freimvorkov-v-razrabotke-saiyta.html](http://ukrbukva.net/page,5,39718-Ispolzovanie-PHP-freymvorkov-v-razrabotke-saiyta.html). - дата доступу:
26.03.2017
- 8 Адаптивные CSS-фреймворки, сетки, классы видимости [Електронний
ресурс] – Режим доступу: [http://klondike-studio.ru/blog/responsive-css-
framework](http://klondike-studio.ru/blog/responsive-css-framework) - дата доступу: 26.03.2017

- 9 Обзор CSS-фреймворков [Электронный ресурс] – Режим доступа до ресурсу: <http://iantonov.me/page/obzor-css-frejmworkov> - дата доступу: 27.03.2017
- 10 Загальні вимоги до змісту та оформлення навчальних посібників та навчально-методичної літератури : методичні рекомендації для викладачів на основі чинних нормативних документів / [уклад. Л. О. Котлова]; - Житомир, 2014. – 56 с.
- 11 Ю.Н.Артименко MySQL. Справочник по языку / Ю.Н.Артименко - Пер. с англ. — М. : Издательский дом "Вильямс", 2005. — 432 с.
- 12 Бази даних. Інформаційні системи – режим доступу: https://www.slideshare.net/escuela_80/ss-12309160 - дата доступу: 07.08.2012
- 13 Модель-Представление-Контроллер (MVC) [Электронный ресурс] – режим доступу: <http://www.yiiframework.com/doc/guide/1.1/ru/basics.mvc>
- 14 Ключові компоненти додатку [Электронный ресурс] – режим доступу: <https://yiiframework.com.ua/uk/doc/guide/basics.application/>- дата доступу: 18.04.2017
- 15 Життєвий цикл додатку [Электронный ресурс] – режим доступу: <https://yiiframework.com.ua/uk/doc/guide/basics.application/>- дата доступу: 19.04.2017
- 16 Что такое фреймворк? [Электронный ресурс] – Режим доступу: <http://www.dbhelp.ru/what-is-framework/page/>. - дата доступу: 28.03.2017
- 17 Десять причин избегать тяжеловесных фреймворков, а также лишних зависимостей в проекте [Электронный ресурс] – Режим доступу: <http://eax.me/avoid-frameworks/>. - дата доступу: 28.03.2017
- 18 Веллинг Л. Разработка веб-приложений с помощью PHP и MySQL / Л. Веллинг, Л. Томсон. – Москва: Вильямс, 2010. – 848 с.
- 19 Шлосснейгл Д. Профессиональное программирование на PHP / Джордж Шлосснейгл. – Москва: Вильямс, 2006. – 624 с.
- 20 Кузнецов М. В. PHP 5 на примерах /С. – Москва, 2007. – 382 с.