

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

**ННК “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)

“ \_\_\_ ” \_\_\_\_\_ 2016 р.

**Дипломна робота**

першого (бакалаврського) \_\_\_\_\_ рівня вищої освіти  
(першого (бакалаврського), другого (магістерського))

зі спеціальності 7.05010102, 8.05010102 Інформаційні технології проектування  
7.05010103, 8.05010103 Системне проектування  
(код та назва спеціальності)

на тему: Аналіз фреймворків як засобів розробки Web-додатків

Виконав : студент 4 курсу, групи ДА-21  
(шифр групи)

\_\_\_\_\_ Проценко Михайло Михайлович \_\_\_\_\_  
(прізвище, ім'я, по батькові) (підпис)

Керівник \_\_\_\_\_ доцент к.т.н. Цурін Олег Пилипович \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант \_\_\_\_\_ \_\_\_\_\_  
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

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

Нормоконтроль \_\_\_\_\_ ст. викладач Бритов О.А. \_\_\_\_\_

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

Студент \_\_\_\_\_  
(підпис)

Київ – 2016 року



5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників, плакатів тощо)

1. UML діаграма послідовностей для архітектури MVC
2. Порівняльна характеристика сучасних WEB-фреймворків.
3. Інструкція по встановленню WEB-фреймворку.
4. Ілюстрація можливостей і роботи сайту, створеного за допомогою обраного WEB-фреймворку.

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економіка	Семенченко Н.В., професор		
Основна частина			

7. Дата видачі завдання 01.02.2016

#### Календарний план

№ з/п	Назва етапів виконання дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання	01.02.2016	
2	Аналіз технічного завдання та вивчення матеріалу з тематики.	15.02.2016	
3	Збір інформації	01.03.2016	
4	Розгляд і аналіз побудови сучасних WEB-фреймворків	16.03.2016	
5	Розгляд позитивних і негативних якостей WEB-фреймворків	20.03.2016	
6	Розробка сайту з використанням Yii 1.1.	01.04.2016	
7	Тестування стабільності роботи сайту	04.05.2016	
8	Підготовка графічного матеріалу, оформлення пояснювальної записки, підготовка до захисту.	20.05.2016	
9	Проходження нормоконтролю, отримання рецензії, подача роботи в ДЕК	14.06.2016	
10	Захист дипломної роботи	21.06.2016	

Студент

\_\_\_\_\_

(підпис)

М.М.Проценко

(ініціали, прізвище)

Керівник проекту (роботи)

\_\_\_\_\_

(підпис)

О.П.Цурін

(ініціали, прізвище)

## АНОТАЦІЯ

бакалаврської дипломної роботи Проценка Михайла Михайловича  
на тему: «Аналіз фреймворків як засобів розробки Web-додатків»

Дана дипломна робота присвячена вивченню можливостей сучасних фреймворків у створенні web-сайтів. В рамках дипломної роботи був проведений аналіз популярних фреймворків, як засобів створення web-додатків. Розглянуто переваги і недоліки використання фреймворків, а також виділено критерії для вибору оптимального фреймворку для створення web-додатку.

Було здійснено вибір фреймворку для реалізації сайту “Питання – відповіді КПІ - Телеком”. Фінальним етапом роботи над дипломною роботою стало створення сайту “Питання – відповіді КПІ - Телеком”, а також створено завдання лабораторної роботи №2б з курсу «Web-проекування».

Загальний обсяг роботи: 145 сторінок, 39 рисунків, 10 таблиць, 28 посилань та 2 додатки на 42 сторінки.

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

# АННОТАЦИЯ

бакалаврской дипломной работы Проценко Михаила Михайловича  
на тему: «Анализ фреймворков как средств разработки Web-приложений»

Данная дипломная работа посвящена изучению возможностей современных фреймворков в создании web-сайтов. В рамках дипломной работы был проведен анализ популярных фреймворков, как средств создания web-приложений. Рассмотрены преимущества и недостатки использования фреймворков, а также выделены критерии для выбора оптимального фреймворка для создания web-приложения.

Было осуществлено выбор фреймворка для реализации сайта "Вопрос - ответ КПИ - Телеком". Финальным этапом работы над дипломной работой стало создание сайта "Вопрос - ответ КПИ - Телеком", а также создан задачи лабораторной работы №2б по курсу «Web-проектирования».

Общий объем работы: 145 страниц, 39 рисунков, 10 таблиц, 28 ссылок и 2 дополнения на 42 страницы.

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

# ANNOTATION

to the bachelor thesis of by Protsenko Mikhail

On a theme “Analysis frameworks as tools for developing Web-based applications”

This course is devoted to studying the possibilities of modern frameworks to create web-sites. As part of the thesis was an analysis of popular frameworks as tools for creating web-applications. The advantages and disadvantages of using frameworks and highlighted the criteria for choosing the optimal framework for creating web-application.

It was a choice made framework for the implementation of the website "Questions - Answers KPI - Telecom". The final stage of master work was the creation of website "Questions - Answers KPI - Telecom" and created a task laboratory work №2b of the course «Web-design."

The total volume of 145 pages, 39 figures, 10 tables, 28 references and 2 applications on page 42.

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

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ.....	10
ВСТУП.....	11
1. ОГЛЯД І АНАЛІЗ СУЧАСНИХ ФРЕЙМВОРКІВ.....	13
1.1. Що повинен мати фреймворк, щоб розглядатися як WEB-технологія?.....	13
1.2. Архітектура Модель–Вид–Контролер.....	15
1.3. Аналіз сучасних Back - End фреймворків та критерії вибору фреймворку для Back – End.....	18
1.3.1. Огляд Back-end фреймворків.....	18
1.3.2. Загальні переваги використання Back-end фреймворку.....	27
1.3.3. Недоліки застосування Back-end фреймворку.....	28
1.3.4. Шляхи та області використання.....	28
1.3.5. Критерії раціональності використання Back-end фреймворку.....	29
1.4. Аналіз сучасних Front-end фреймворків для створення адаптивного дизайну.....	32
1.4.1. Огляд Front – End фреймворків.....	32
1.4.2. Переваги і недоліки застосування Front-end фреймворку.....	36
1.5. Висновки до розділу 1.....	37
2. ВИМОГИ ДО САЙТУ «ПИТАННЯ-ВІДПОВІДІ КПП-ТЕЛЕКОМ». ВИБІР ФРЕЙМВОРКУ ДЛЯ РЕАЛІЗАЦІЇ. ОГЛЯД РОБОТИ З ОБРАНИМ ФРЕЙМВОРКОМ.....	38
2.1. Вимоги до сайту «Питання-відповіді КПП- Телеком».....	38

2.2.	Вибір Back – End фреймворку для реалізації сайту	
	“ Питання – відповіді КПІ – Телеком ” .....	40
2.3.	Yii як програмний каркас для створення web–додатків.....	43
2.3.1.	Структура Yii.....	43
2.3.2.	Ключові компоненти додатку Yii.....	44
2.3.3.	Реалізація MVC в Yii .....	46
2.3.4.	Типовий запит Yii 1.1.....	47
2.3.5.	Життєвий цикл додатку Yii.....	49
2.3.6.	Процес розробки додатку з використанням Yii.....	50
2.4.	Висновки до розділу 2.....	51
3.	РОЗРОБКА САЙТУ ЗА ДОПОМОГОЮ YII ТА	
	СТВОРЕННЯ ЗАВДАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 2Б	
	З КУРСУ “WEB-ПРОЕКТУВАННЯ”.....	53
3.1.	Вибір програмного забезпечення.....	53
3.2.	Розробка дизайну сайту.....	53
3.3.	Розробка бази даних сайту .....	57
3.3.1.	ER-діаграма.....	57
3.3.2.	Створення БД за допомогою PhpMyAdmin.....	58
3.4.	Встановлення Yii на локальний сервер.....	60
3.5.	Налаштування Yii.....	62
3.5.1.	Налаштування з’єднання з базою даних .....	62
3.5.2.	Активація кодогенератора Gii.....	62
3.5.3.	Встановлення доповнення Yiiastrap.....	63
3.5.4.	Налаштування системи доступу на основі ролей.....	63
3.6.	Реалізація архітектури MVC сайту.....	64
3.6.1.	Генерація моделей сайту.....	65
3.6.2.	Генерація контролерів та представлень сайту.....	66
3.7.	Зміна стандартного інтерфейсу web-сайту.....	69



3.7.1. Зміна дизайну header та footer сайту.....	71
3.7.2. Зміна дизайну контентної частини сайту.....	72
3.8. Розміщення сайту на хостинг.....	75
3.9. Висновки до розділу 3.....	76
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.....	81
4.2. Обґрунтування системи параметрів ПП.....	83
4.2.1. Опис параметрів.....	83
4.2.2. Кількісна оцінка параметрів.....	84
4.2.3. Аналіз експертного оцінювання параметрів.....	87
4.3. Аналіз рівня якості варіантів реалізації функцій.....	91
4.4. Економічний аналіз варіантів розробки ПП.....	92
4.5. Вибір кращого варіанта ПП техніко-економічного рівня.....	97
4.6. Висновки до розділу 4.....	97
ВИСНОВКИ.....	99
ПЕРЕЛІК ПОСИЛАНЬ.....	101
Додаток А.....	104
Додаток Б.....	131

**Перелік скорочень, умовних позначень, термінів**

ПЗ	програмне забезпечення
API	Application Programming Interface
URL	Uniform Resource Locator
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
MVC	Model-View-Controller

## ВСТУП

Швидкість, з якою поширюється інформація сьогодні, є вражаючою. Інтернет став каталізатором до поширення інформації. В Інтернеті знаходяться мільйони веб-сайтів різного спрямування. Створення та підтримка веб-сайту є важкою працею.

Розробка веб-сайтів є перспективним напрямком в розвитку інформаційних технологій. Глобальна мережа Інтернет охоплює з кожним роком все більше і більше користувачів. Так, в 2015 році було проведено дослідження Міжнародним союзом електрозв'язку, яким було встановлено, що розмір інтернет- аудиторії на кінець 2015 року складе 3.2 млрд чоловік , що становить 43 % населення світу[1]. Для порівняння в 2012 році , кількість користувачів інтернету становила 2.4 млрд чоловік [2]. Також згідно з статистикою зростає кількість користувачів , які використовують смартфони для доступу в мережу Інтернет , так згідно з проведеним дослідженням компанією We Are Social's , в порівняння з 2014 роком кількість користувачів мобільного інтернету зросла на 115 млн чоловік[3]. Зрозуміло , що зі збільшенням інтернет-аудиторії збільшується кількість корпоративних і інформаційних ресурсів .

В останні роки спостерігається бурхливий розвиток Інтернет-технологій. Можна спостерігати значні зміни в мережі Інтернет. Сайти, які раніше були платформою для розташування статичного контенту, тепер стали багато функціональними, інтерактивними системами для надання різної інформації.

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.

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. Без використання фреймворків, стає набагато складніше створювати web-додатки, супроводжувати і модернізувати їх. Між тим, використання фреймворків робить процес створення програми набагато більш легким і функціональним. Фреймворки можна розділити на дві категорії фреймворки для дві категорії це фреймворки для написання Back-End(серверний функціонал сайту) та Front-End(клієнтський функціонал сайту) .

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.

Тема вибору фреймворку погано розкрита в мережі Інтернет та друкованих джерелах . Тому доцільно виділити критерії , на які потрібно спиратися при виборі фреймворку, щоб не допустити непоправної помилки. Для порівняння обраних фреймворків слід обрати такі критерії порівняння, що будуть найкраще демонструвати переваги та недоліки кожного. Беручи до уваги те, що порівнюватися будуть фреймворки з окремих груп (серверні та клієнтські фреймворки), необхідно обрати різні критерії порівняння для різних груп. Також потрібно оцінити які фреймворки є найбільш простими та дозволяють створити веб-сайт швидше за все, а які надають сайту найбільше можливостей й гнучкості.

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

# 1. ОГЛЯД І АНАЛІЗ СУЧАСНИХ ФРЕЙМВОРКІВ

## 1.1. Що повинен мати фреймворк, щоб розглядатися як WEB-технологія?

Фреймворк (в інформаційних системах) - це структура програмної системи, що полегшує розробку і об'єднання різних компонентів великого програмного проекту. На відміну від бібліотек, які об'єднують набір підпрограм близької функціональності, фреймворк містить 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.

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.

Також необхідно з'ясувати, що ж саме являють собою WEB фреймворки. WEB фреймворки - це каркас, призначений для створення динамічних веб-сайтів, мережеских 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 поводить себе 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.

## **1.2. Архітектура Модель–Вид–Контролер**

Для як найшвидшого розуміння суті архітектури MVC, необхідно розглянути фундаментальну концепцію web-додатків. Архітектура Модель-Вид-

Контролер (англ. *Model-View-Controller*) дуже важлива в контексті фреймворків, оскільки вона є типовою для більшості web-додатків.

Model View Controller (MVC, «модель-представлення-контролер», «модель-вид-контролер») – схема використання декількох шаблонів проектування, за допомогою яких модель, інтерфейс і взаємодія з користувачем розділені на три окремі компоненти таким чином, щоб модифікація одного з компонентів мала мінімальний вплив на інші. Дана схема проектування часто використовується для побудови архітектурного каркаса, коли переходять від теорії до реалізації в конкретній предметній області.

Архітектурний шаблон Модель-Вид-Контролер (MVC) поділяє програму на три частини. У тріаді до обов'язків компоненту Модель (Model) входить зберігання даних і забезпечення інтерфейсу до них; вигляд (View) відповідальний за відображення цих даних користувачеві; контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача і повідомляє про зміни компонента Модель. Така внутрішня структура в цілому поділяє систему на самостійні частини і розподіляє відповідальність між різними компонентами.

MVC поділяє цю частину системи на три самостійні: введення даних, компонент обробки даних і виведення інформації. Модель, як вже було відмічено, інкапсулює ядро даних і основний функціонал з їх обробки. Також компонент Модель не залежить від процесу введення або виведення даних. Компонент виводу Вигляд може мати декілька взаємопов'язаних областей, наприклад, різні таблиці і поля форм, в яких відображається інформація. У функції Контролера входить моніторинг за подіями, що виникають в результаті дій користувача (зміна положення курсора миші, натиснення кнопки або введення даних в текстове поле). Зареєстровані події транслюються в різні запити, що спрямовуються компонентам Моделі або об'єктам, відповідальним за відображення даних.



Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через Контролер внесе зміни до Моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися.



Рисунок 1.1 – *Архітектура Model–View–Controller*

Також існує архітектура Модель-Вид-Представник (Model-View-Presenter) (MVP) та архітектура N–Tier. Відмінності між MVC і MVP полягають в тому, як дані надходять з моделі у вид. Так, у MVC вони надходять з моделі у вид, а в MVP проходять через Вид. Очевидною перевагою при використанні концепції MVC є чіткий поділ логіки подання (інтерфейсу користувача) і логіки програми.

В даний час набирає популярності застосування об'єктно-реляційного відображення (Object Relational Mapping – ORM) для реалізації моделі web-додатку. ORM у широкому сенсі – це правило або метод, що зв'язує об'єкти мови програмування з відносинами реляційної бази даних і трансформує операції над об'єктами у відповідні операції над відносинами. У цьому сенсі в будь-якому проекті, розробка якого ведеться за допомогою об'єктно-орієнтованої мови, а дані зберігаються в реляційній СУБД, ця трансформація якимось чином вже здійснюється.

## 1.3. Аналіз сучасних Back - End фреймворків та критерії вибору фреймворку для Back – End

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

### 1.3.1. Огляд Back-end фреймворків

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)

**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

**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.

**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.

**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 вищерозглянутих 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 вищерозглянутих фреймворків

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

<b>Фреймворк</b>							
<b>Критерій</b>	<b>Zend</b>	<b>CakePHP</b>	<b>Symfony2</b>	<b>Yii</b>	<b>ASP.NET</b>	<b>ROR</b>	<b>Struts</b>
<b>Мова</b>	PHP	PHP	PHP	PHP	ASP.NET	Ruby	Java
<b>Шаблон проектування MVC</b>	+	+	+	+	+	+	+
<b>Застосування ORM</b>	+	+	+	+	ORM– незалежна	+	+
<b>Інтеграція з БД</b>	+	+	+	+	***	+	***
<b>Механізми інтернаціоналізації і локалізації</b>	+	+	+	+	***	+	+
<b>Використання шаблонів при створенні інтерфейсів користувача</b>	+	+	***	+	+	+	+
<b>Створення і перевірка форм</b>	+	+	+	+	+	+	+
<b>Управління доступом на основі ролей</b>	+	+	***	+	+	+	ні
<b>Використання AJAX</b>	+	+	***	+	+	+	+
<b>ЧПУ (Friendly URL)</b>	+	+	+	+	+	+	ні
<b>Модульне тестування (unit–testing)</b>	+	+	***	+	+	+	+
<b>Система кешування</b>	+	+	+	+	+	+	ні

### 1.3.2. Загальні 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:

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

### **1.3.3. Недоліки застосування Back-end фреймворку**

Недоліки застосування 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.

### **1.3.4. 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.

### **1.3.5. Критерії раціональності використання Back-end Фреймворку**

Фреймворки використовуються в різних областях залежно від їх основних характеристик. Виділимо основні критерії раціональності використання фреймворку в таблиці 1.3.

Таблиця 1.2 – Критерії раціональності використання Back-end фреймворку

Назва	Характеристика
<b>Підтримка баз даних</b>	Питання підтримки баз даних в PHP фреймворк дуже важливе. Наприклад, CodeIgniter підтримує MySQL, Oracle і SQLite, а фреймворк Kohana не підтримує Oracle і SQLite. Частина фреймворків мають вбудований ORM – шар, частина – ні. Залежно від використовуваної бази даних для розробки проекту доводиться вибирати той чи інший PHP фреймворк.
<b>Підтримка спільноти</b>	Для комфортного вирішення проблем повинен мати хороше співтовариство, не тільки з точки зору розміру, але і в якості і в готовності допомогти. Навіть якщо це маленька спільнота, але є зворотний зв'язок від спільноти, це можна вважати плюсом. Так само плюсом є наявність російськомовного співтовариства.
<b>Документація</b>	Частина фреймворк мають слаборозвинуту, застарілу документацію. Частина не мають російської документації. Тому перед вибором фреймворка необхідно переконатися в тому що документація актуальна, вчасно оновлюється і доповнюється, і що інструкція із застосування проста в розумінні.
<b>Продуктивність</b>	Ключовим фактором при виборі так же може бути продуктивність фреймворка, наприклад, частина фреймворків підтримує кешування на достатньому рівні, частина – ні.
<b>Безпека</b>	Не всі фреймворки стійкі до різного роду атак, тому перед вибором фреймворка необхідно ретельно проаналізувати активність розвитку, розмір спільноти, а так само наявність вбудованих засобів для захисту від атак.
<b>Поріг освоєння</b>	Не всі фреймворки прості в освоєнні, це дуже важливо враховувати при виборі, так як на освоєння одного фреймворка може не вистачити й року, а на освоєння іншого – вистачить всього тижня.

Таблиця 1.2 – Критерії раціональності використання Back-end фреймворку(Продовження)

Назва	Характеристика
<b>Швидкість розробки</b>	Так само слід врахувати той факт, що на одному фреймворці проект розробляється швидше, на іншому – ні. Приміром, розробки із застосуванням фреймворка zend триває більше ніж із застосуванням уїі .
<b>Архітектура</b>	Фреймворк також повинен використовувати MVC архітектуру. Більшість хороших PHP фреймворків мають бібліотеки, плагіни, модулі та розширення. Це дуже добре для того щоб реалізувати велике коло функціоналу та вдосконалити і прискорити процес розробки.
<b>Швидкість розвитку</b>	Цей пункт так само дуже важливий, так як деякі фреймворки оновлюються раз на пару років (codeigniter), а деякі раз на пару місяців. Це дозволяє уникнути використання старого, недопрацьованого коду при розробці.
<b>Зворотна сумісність</b>	Не всі фреймворки зворотньо сумісні, тобто, при оновленні фреймворка в проекті може виникнути необхідність у повній переробці проекту. Частина фреймворків умовно зворотньо сумісна, наприклад, при оновленні молодшої частини версії (minor) все сумісно, а при старшій – ні. Так само великим плюсом є керівництво по переходу на нову версію фреймворка.
<b>Використання шаблонів при створенні інтерфейсів користувача</b>	Наявність даної функції значно спрощує створення інтерфейсу сайту , а іноді коли потрібний простий сайт то може звести написання дизайну до мінімуму.
<b>Створення і перевірка форм</b>	Наявність даного функціоналу в фреймворку полегшує створення полів для вводу тексту, логіну чи пароллю до мінімуму. А також додає потрібну майже в усіх проектах функцію валідації форм.

## 1.4. Аналіз сучасних Front-end фреймворків для створення адаптивного дизайну

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.

### 1.4.1. Огляд Front – End фреймворків

**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.



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 завантаження 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 `rems`.

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.

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 1.3 – Порівняльна характеристика популярних Front-end фреймворків для створення адаптивного дизайну.

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

#### 1.4.2. Переваги і недоліки застосування Front-end фреймворків

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

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.

## **1.5. Висновки до розділу 1**

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 забезпечитися 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 молодосвідченого 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.

## **2. ВИМОГИ ДО САЙТУ «ПИТАННЯ-ВІДПОВІДІ КПП-ТЕЛЕКОМ». ВИБІР ФРЕЙМВОРКУ ДЛЯ РЕАЛІЗАЦІЇ. ОГЛЯД РОБОТИ З ОБРАНИМ ФРЕЙМВОРКОМ.**

В даному розділі буде описано вимоги до сайту «Питання-відповіді КПП-Телеком» , який буде реалізований з використанням фреймворку. Також буде здійснено вибір фреймворку для реалізації тестового проекту за критеріями, які були розглянуті вище. Після вибору буде розкрито особливості обраного фреймворку та принципи створення web – додатків за його допомогою. Всі вищезгадані аспекти будуть описані в окремих підрозділах.

### **2.1. Вимоги до сайту «Питання-відповіді КПП-Телеком»**

Проектом для огляду можливостей фреймворків є створення сайту “ Питання – відповіді КПП – Телеком ”. Основною задачею сайту є комунікація користувачів провайдера КПП – Телеком з адміністрацією провайдера. Користувачі можуть ставити свої питання стосовно роботи провайдера , цінової політики , а також будь-які інші запитання, які можуть виникнути у користувачів.

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) Питання – відповіді(основний розділ сайту, в якому користувач ставить своє питання, а адміністратор дає відповідь) ;
  - 2) FAQ ( часті запитання) ;

- 3) Інформація ( інформація про структуру , цінову політику , апаратне та програмне забезпечення КПП - Телеком, а також інформація для зв'язку з КПП – Телеком ) ;
- 4) Про КПП – телеком ( загальна інформація про КПП – Телеком).
3. Сайт повинен підтримувати написання питання українською, російською та англійською мовами.
4. На сайті повинна бути реалізована авторизація.
5. Сайт повинен мати такі ролі користування:
  - 1) Адміністратор (зарєєстрований користувач, якому присвоєна роль адміністратора). Має мати такі права доступу :
    - a) редагування та видалення питань
    - b) відповідь на питання
    - c) перенесення питання до розділу частих запитань
    - d) заповнення інформаційних розділів інформацією.
  - 2) Гість. Має мати такі права доступу :
    - a) перегляд сайту
    - b) можливість написання запитання в розділі «Питання – відповіді».
6. Вимоги до функціоналу сайту:
  - 1) Написання повідомлень користувачами
  - 2) Можливість адміністратора відповідати на повідомлення
  - 3) Захист від спаму(для реалізації буде використана капча)
  - 4) Можливість адміністратора перенести питання з розділу “Питання - відповіді” до розділу “FAQ”(перенесення з редагуванням і без редагування)
  - 5) Можливість адміністратора редагувати повідомлення користувача
  - 6) Редагування адміністратором контенту розділів “Інформація” та “FAQ”

- 7) Відповідь на питання посиланням на відповідь, що була раніше ( при цьому відповідь, на яку вказує посилання, повинна підсвічуватись);
- 8) Видалення адміністратором запитань користувачів
- 9) Валідація форм.

7. Вимоги до дизайну сайту:

- 1) Сайт повинен бути адаптивним та кроссбраузерним
- 2) Чіткий розподіл пари питання – відповідь
- 3) Підтримка роботи на мобільних пристроях.

8. Сайт потрібно розмістити на хостингу, за адресою `vstup.kpi.ua`.

## **2.2. Вибір Back – End фреймворку для реалізації сайту “ Питання – відповіді КПІ – Телеком ”**

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 такої таблиці:



Таблиця 2.1 – Вибір Back – End фреймворку

Назва	Zend	CakePHP	Symfony2	Yii
Підтримка бази даних 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	RHP5, ООП, шаблони проектування.	RHP, ООП, вміння розбиратися в вихідному коді фреймворка.	RHP5, ООП, ORM, консоль.	RHP5, ООП.
Швидкість розробки	Низька	+	+	+
Архітектура MVC	+	+	+	+

Таблиця 2.1 – Вибір Back – End фреймворку(Продовження)

<b>Назва</b>	<b>Zend</b>	<b>CakePHP</b>	<b>Symfony2</b>	<b>Yii</b>
<b>Швидкість розвитку</b>	+	+	+	+
<b>Зворотна сумісність</b>	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.
<b>Використання шаблонів при створенні інтерфейсів користувача</b>	+	+	-	+
<b>Створення і перевірка форм</b>	+	+	+	+

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.

## **2.3. 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.

### **2.3.1 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:

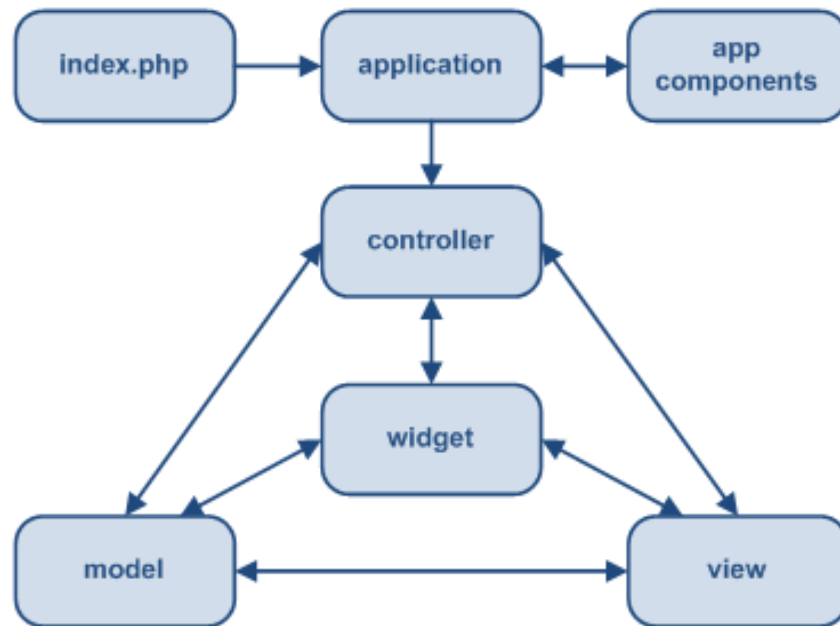


Рисунок 2.1 - Статична структура додатку Yii[12]

### 2.3.2. 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.

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

- `assetManager`: `CAssetManager` - управляє 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 зверненні до компоненту буде повернуто `null`;

- `clientScript`: `CClientScript` - управляє клієнтськими скриптами (JavaScript 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` - обробляє спіймані помилки і виключення PHP;
- `format`: `CFormatter` - форматує дані для їх подальшого відображення;
- `messages`: `CPhpMessageSource` - надає переклади повідомлень, використовуваних в Yii-додатку;
- `request`: `CHttpRequest` - містить інформацію про запит;
- `securityManager`: `CSecurityManager` - надає функції, пов'язані 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 парсингу і формування URL;
- `user`: `CWebUser` - надає ідентифікаційну інформацію поточного користувача;
- `themeManager`: `CThemeManager` - управляє темами оформлення.

### 2.3.3. 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`.

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 користувальницького 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`.

#### **2.3.4. Типовий запит Yii 1.1.**

Розглянувши структуру додатків у Yii 1.1., перейдемо до специфіки взаємодії кожного компонента із системою для виконання базового запиту.

Наступна діаграма описує типову послідовність процесу обробки користувальницького запиту додатком:

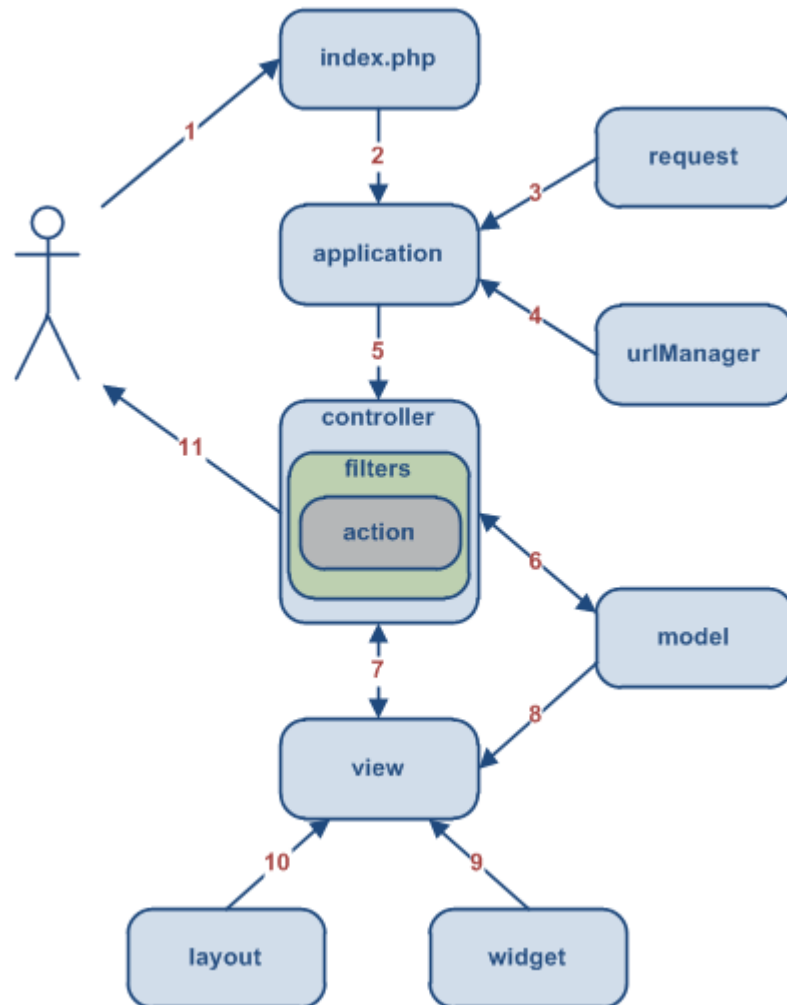


Рисунок 2.2 - Типова послідовність роботи додатку Yii[12]

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.

### 2.3.5. Життєвий цикл додатку Yii

Життєвий цикл додатку при обробці користувацького запиту виглядає наступним чином:

1. Попередня ініціалізація додатку через `CApplication :: preinit ()`.
2. Ініціалізація обробника помилок.
3. Реєстрація компонентів ядра.
4. Завантаження конфігурації програми.
5. Ініціалізація додатки `CApplication :: init ()`:

- 1) реєстрація поведень додатку;
- 2) завантаження статичних компонентів програми.

6. Виклик події `onBeginRequest`.

- 1) Опрацювання запиту;
- 2) збір інформації про запит;
- 3) створення контролера;
- 4) запуск контролера.

7. Виклик події `onEndRequest`.

### **2.3.6. 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.

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 використовуюваного 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.

## **2.4. Висновки до розділу 2**

Після аналізу вимог сайту «Питання-відповіді КПП-Телеком» , 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 тлі 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. Плюс до цього мною було створене російськомовну спільноту яке постійно поповнюється новими матеріалами.
2. Допомога від розробника фреймворка. Великий плюс підтримки Yii - це можливість в будь-який день поспілкуватися і задати питання розробнику Yii на форумі. Було відмічено що він практично кожен день переглядає і відповідає на питання на форумі, допомагає порадами і корисними "фішками". Ви завжди можете написати йому приватне

повідомлення та бути впевнені що отримаєте найправильніший відповідь на вашу питання. На жаль, розробник фреймворка погано знає російську мову, тому повідомлення просимо викладати йому англійською або китайською.

3. Повна підтримка ООП. Фреймворк повністю заточений під п'яту версію php що дозволяє підтримувати весь функціонал при об'єктно орієнтованому програмуванні

4. 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.

5. 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 ТА СТВОРЕННЯ ЗАВДАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 2Б З КУРСУ “WEB-ПРОЕКТУВАННЯ”**

В даному розділі буде розглянуто створення сайту “Питання-відповіді КПІ-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 при вході гостя:

Питання-відповіді КПІ-Телеком

Питання та відповіді

FAQ

Інформація ▾

Про КПІ-Телеком

### Задати питання

**Misha** 2016-05-30 12:47:48  
Що таке персональний кабінет?

**Administrator** 2016-05-30 12:48:08  
Персональний кабінет - Ваша персональна захищена web-сторінка, де можна отримувати відомості про замовлені послуги, керувати послугами, дивитись свій трафік, задавати нам запитання і т.д.

**Misha** 2016-05-30 12:46:45  
Чи можу я подивитись на вхідний і вихідний трафік?

**Administrator** 2016-05-30 16:35:08  
Так, це можна зробити. Для цього необхідно зайти у свій "Персональний кабінет", у ньому Ви побачите опцію "Статистика". Клікнувши на неї Ви зможете переглянути свій вхідний і вихідний трафік поденно за останні два місяці.

**Misha** 2016-05-26 00:28:09  
Які правила зміни пакетів послуги Інтернет?

**Administrator** 2016-05-30 16:46:27  
Щоб змінити свій тарифний план, необхідно зайти у свій "Персональний кабінет" у розділі "Керування пакетом" є опція "Запланувати зміну пакету", де необхідно ввести пакет, яким Ви бажаєте користуватись і свій пароль, після чого клікнути на кнопку "Відправити".

Copyright © 2016 Проценко , Цурін КПІ-Телеком  
All Rights Reserved.  
Powered by [Yii Framework](#).

Рисунок 3.1 - Головна сторінка сайту при вході гостя.

## Головна сторінка сайту при вході під іменем адміністратора:

Питання-відповіді КПІ-Телеком    Питання та відповіді    FAQ    Інформація ▾    Про КПІ-Телеком    Logout (admin)

### Задати питання

**Misha** 2016-05-30 12:47:48  
Що таке персональний кабінет?  
Редагувати Видалити

**Administrator** # 2016-05-30 12:48:08  
Персональний кабінет - Ваша персональна захищена web-сторінка, де можна отримувати відомості про замовлені послуги, керувати послугами, дивитись свій трафік, задавати нам запитання і т.д.  
Додати до FAQ Відредагувати та додати до FAQ Редагувати Видалити

**Misha** 2016-05-30 12:46:46  
Чи можу я подивитись на вхідний і вихідний трафік?  
Редагувати Видалити

**Administrator** # 2016-05-30 16:35:08  
Так, це можна зробити. Для цього необхідно зайти у свій "Персональний кабінет", у ньому Ви побачете опцію "Статистика". Клікнувши на неї Ви зможете переглянути свій вхідний і вихідний трафік поденно за останні два місяці.  
Додати до FAQ Відредагувати та додати до FAQ Редагувати Видалити

**Misha** 2016-05-26 00:28:09  
Які правила зміни пакетів послуги Інтернет?  
Редагувати Видалити

Copyright © 2016 Проценко , Цурін КПІ-Телеком  
All Rights Reserved.  
Powered by [Yii Framework](#).

Рисунок 3.2 -Головна сторінка сайту при вході під іменем адміністратора.

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

Мобільна версія сайту:

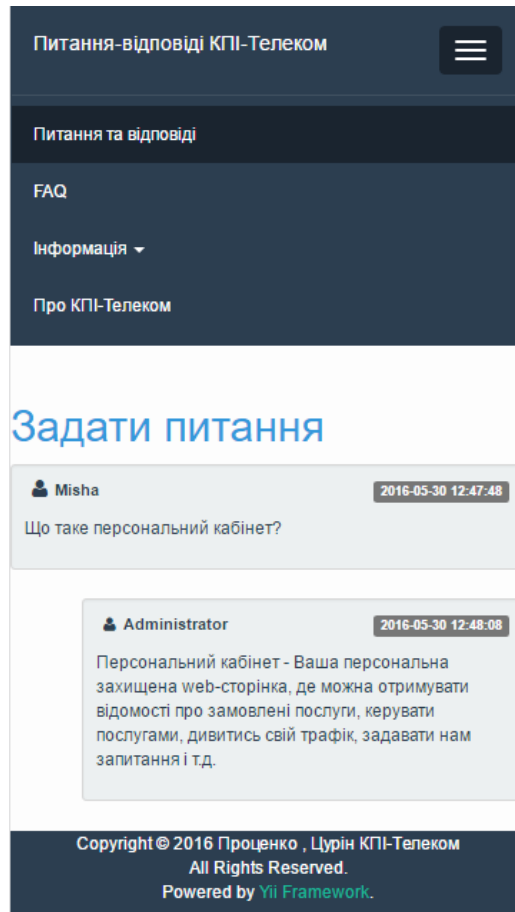


Рисунок 3.3 - Мобільна версія сайту.



### 3.3. Розробка бази даних сайту

Розробка структури БД передбачає повний опис всіх атрибутів сутностей (полів таблиць). Кожна таблиця - масив з однорідних елементів, які прийнято називати записами. Запис - неподільна одиниця інформації в БД, але можна сформулювати такий запит, щоб отримати якусь частину цієї інформації. Запис може містити в собі одну або кілька іменованих полів, що задаються при створенні таблиці. Кожне з полів має певний тип (наприклад, ціле число, текст та інші).

У всі таблиці можна додавати записи, видаляти їх, робити пошук за таблицями і виводити потрібну інформацію. Всі ці дії виконуються за допомогою SQL (Structured Query Language) запитів SELECT, DELETE, CREATE, INSERT.

#### 3.3.1. ER-діаграма

Виходячи з вимог до створення веб-сайту, була складена структура БД.

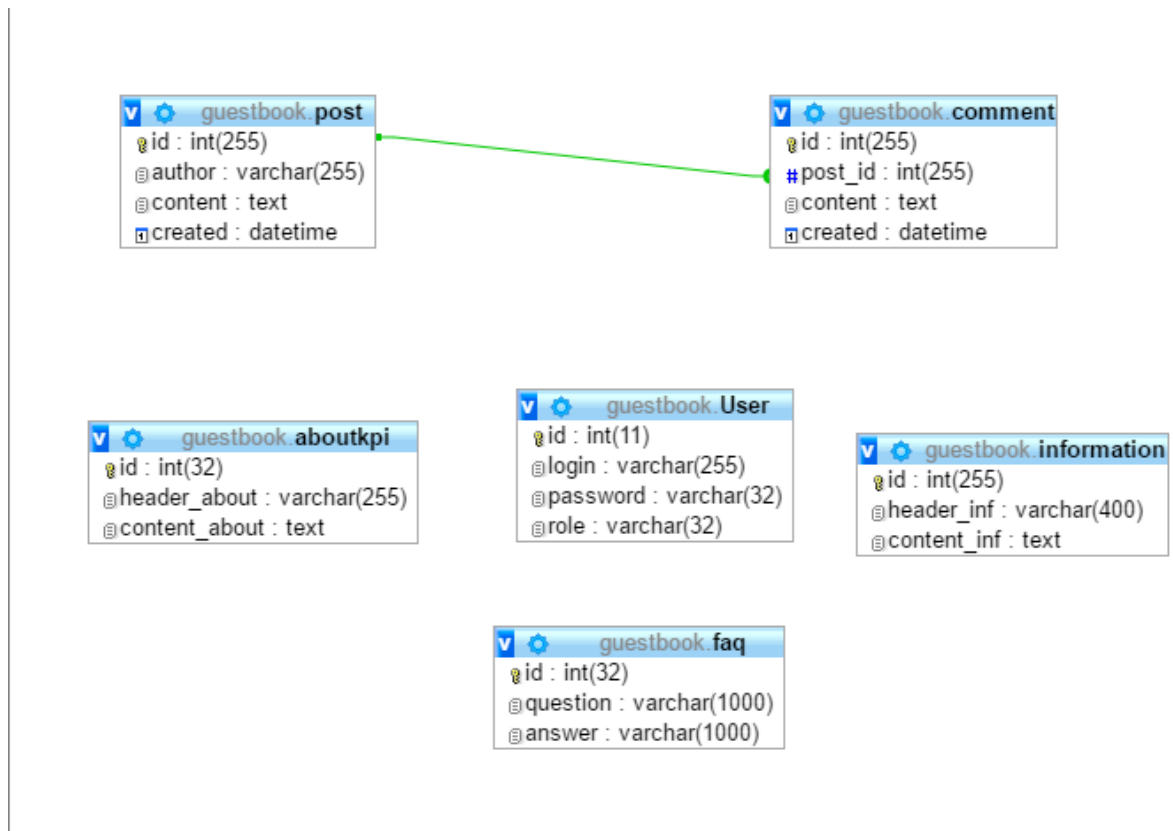


Рисунок 3.4 - ER-діаграма.

В даній ER-діаграмі ми бачимо 6 таблиць. Розглянемо їх детальніше:

**Post** - таблиця в якій зберігаються питання користувачів сайту.

**Comment** - містить відповіді адміністрації сайту на задані питання. Тут пункт `post_id` пов'язаний з пунктом `id` в таблиці `Post` (при видаленні питання, автоматично видалиться і відповідь на нього).

**User** - містить дані адміністраторів сайту( логін, пароль та роль користування сайтом).

**Faq** - містить часті питання та відповіді на них.

**Information** - дані для наповнення розділу “Інформація” на сайті.

**Aboutkpi** - дані для наповнення розділу “Про КПІ – Телеком” на сайті.

### 3.3.2. 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”**:

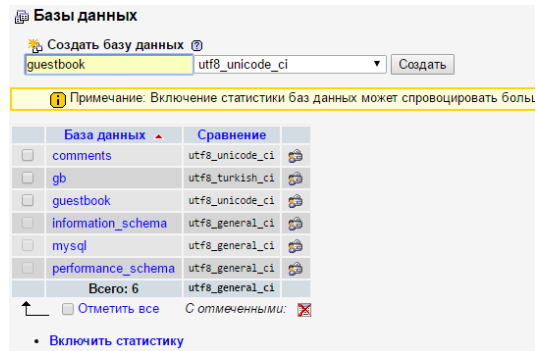


Рисунок 3.5 - Створення БД в PhpMyAdmin.

Тиснемо кнопку **“Создать”** і БД створена.

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

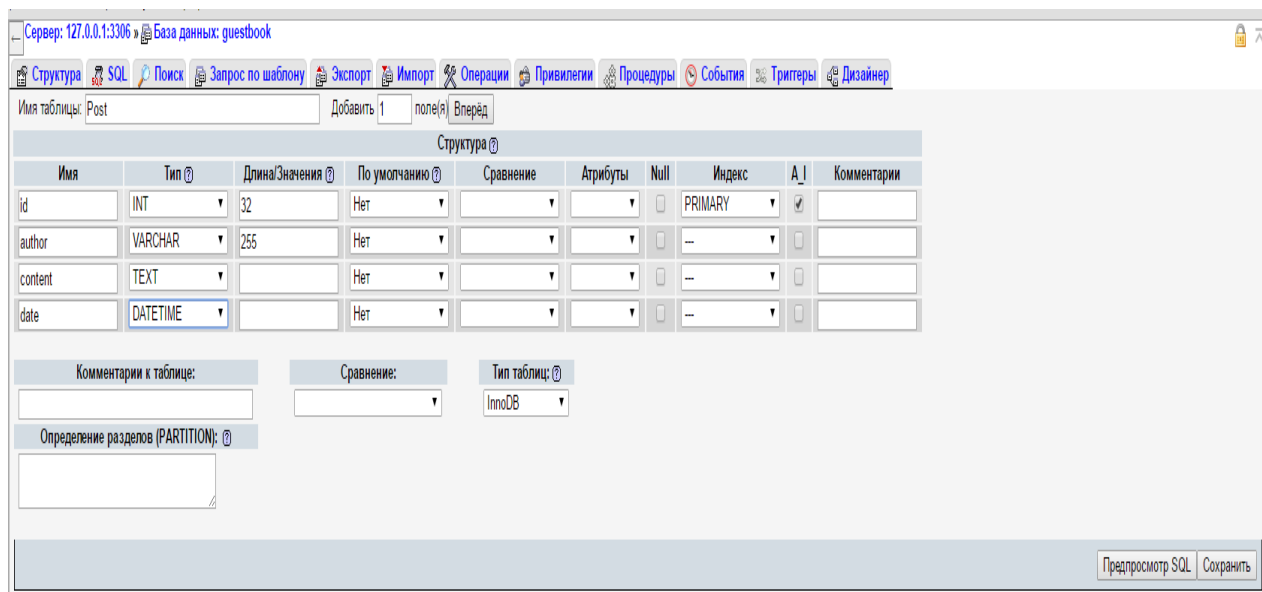


Рисунок 3.6 - Створення таблиці Post.

Аналогічно створюємо решту таблиць. В результаті отримуємо:

Таблица ▲	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
<input type="checkbox"/> aboutkpi	★ [иконки]	1	InnoDB	utf8_unicode_ci	32 КИБ	-
<input type="checkbox"/> comment	★ [иконки]	3	InnoDB	utf8_unicode_ci	16 КИБ	-
<input type="checkbox"/> faq	★ [иконки]	3	InnoDB	utf8_unicode_ci	16 КИБ	-
<input type="checkbox"/> information	★ [иконки]	5	InnoDB	utf8_unicode_ci	16 КИБ	-
<input type="checkbox"/> post	★ [иконки]	3	InnoDB	utf8_unicode_ci	16 КИБ	-
<input type="checkbox"/> User	★ [иконки]	1	InnoDB	utf8_unicode_ci	16 КИБ	-
<b>6 таблиц</b>	<b>Всего</b>	<b>16</b>	<b>InnoDB</b>	<b>utf8_unicode_ci</b>	<b>112 КИБ</b>	<b>0 Байт</b>

Отметить все    С отмеченными:

Рисунок 3.7 - Список таблиць бази даних проекту.

На цьому створення бази даних для проекту завершено.

### 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\GuestBook ). 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\GuestBook\framework\yiiic webapp C:\WebSever\OpenServer\domains\GuestBook\`
4. Далі відобразиться, повідомлення про те чи бажаємо створити додаток.  
Вводимо : `yes`

Отримано такий результат:

```

Командний рядок
Microsoft Windows [Version 10.0.10240]
(c) Корпорація Майкрософт (Microsoft Corporation), 2015 р. Все права захищені.
C:\Users\MISHA>cd C:\
C:\WebSever\OpenServer>php PHP-5_4.php.exe C:\WebSever\OpenServer\doma
ns\GuestBook\framework\yii C:\WebSever\OpenServer\domains\GuestBook
Create a Web application under 'C:\WebSever\OpenServer\domains\GuestBook'? (yes
no) In: 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
generate index.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/testdrive.db
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\unit
generate protected\tests\WebTestCase.php
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\vendor
mkdir C:\WebSever\OpenServer\domains\GuestBook\protected\views\layouts
generate protected\views\layouts\column1.php
generate protected\views\layouts\column2.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\yii
generate protected\yii.bat
generate protected\yii.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
s
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\doma
ins\GuestBook.

```

Рисунок 3.8 - Успішне встановлення Yii 1.1.

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

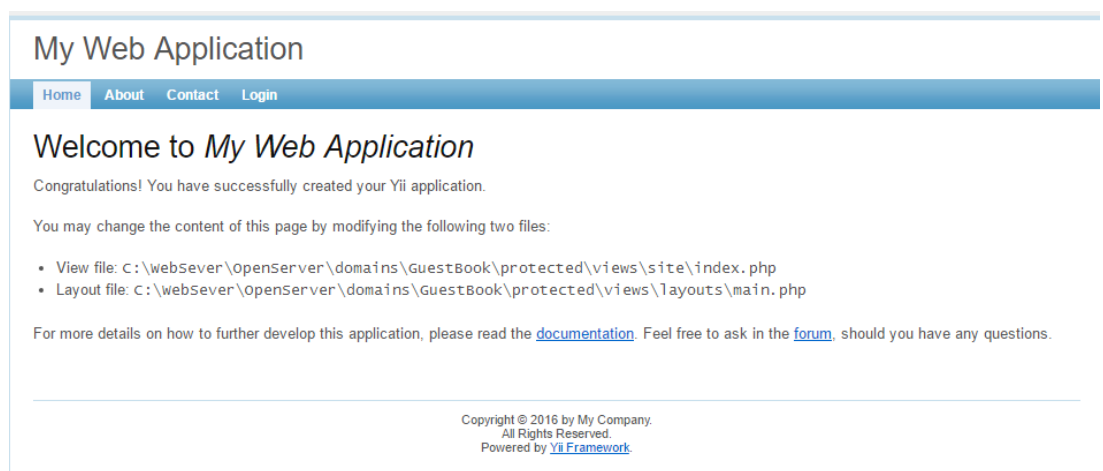


Рисунок 3.9 - Стандартно-згенерований додаток Yii 1.1.

## 3.5. Налаштування Yii

### 3.5.1. Налаштування з'єднання з базою даних

Для створення свого проекту потрібно налаштувати Yii 1.1 для роботи з базою даних, яка була створена раніше. Для цього потрібно відредагувати файл `database.php`, розміщений за адресою: `C:\WebSever\OpenServer\domains\GuestBook\protected\config\database.php`.

Для цього потрібно додати наступний код:

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

Тепер проект використовує створену раніше базу даних під назвою `guestbook`.

### 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://guestbook/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 Yiiistrap. При встановленні є такі етапи:

1. Створити папку vendor в корньовому каталозі сайту.
2. Розархівувати завантажений архів з доповненням в папку vendor.
3. Додати в файл main.php( шлях до файлу: 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\guestnew\protected\components\PhpAuthManager.php);
2. UserIdentity.php (шлях до файлу :  
C:\WebSever\OpenServer\domains\guestnew\protected\components\UserIdentity.php);
3. WebUser.php (шлях до файлу :  
C:\WebSever\OpenServer\domains\guestnew\protected\components\WebUser.php);
4. auth.php (шлях до файлу :  
C:\WebSever\OpenServer\domains\guestnew\protected\config\auth.php)

Код цих файлів описано в додатку.

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 1.1. завершено.

### 3.6. Реалізація архітектури MVC сайту

Сайт буде мати декілька роздів:

- Питання – відповіді ;
- FAQ ( часті запитання) ;



- Інформація ( інформація про структуру , цінову політику , апаратне та програмне забезпечення КПІ - Телеком, а також інформація для зв'язку з КПІ – Телеком ) ;
- Про КПІ – телеком ( загальна інформація про КПІ – Телеком);
- Вхід( сторінка прихована на сайті , створена для входу на сайт адміністратора(доступна за адресою: адреса\_проекту/index.php?r=site/login).

Оскільки основним розділом сайту є розділ “Питання – відповіді ”, 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://guestbook/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:

The screenshot displays the 'Model Generator' interface of the Yii Code Generator. It includes a sidebar with navigation links for various generators. The main area contains the following configuration options:

- Database Connection:** db
- Table Prefix:** [empty]
- Table Name:** post
- Model Class:** Post
- Base Class:** CActiveRecord
- Model Path:** application.models
- Build Relations:**
- Use Column Comments as Attribute Labels:**
- Code Template:** default (C:\WebSever\OpenServer\domains\GuestBook\framework\gii\generators\model\templates\default)

At the bottom, there is a table with the following content:

Code File	Generate
models\Post.php	Generate new ✓

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. Після генерації всіх моделей папка models повинна містити такі файли:

Имя	Дата изменения	Тип	Размер
Aboutpri.php	05.06.2016 18:16	Файл "PHP"	3 КБ
Comment.php	05.06.2016 18:16	Файл "PHP"	3 КБ
Information.php	05.06.2016 18:16	Файл "PHP"	3 КБ
LoginForm.php	05.06.2016 16:30	Файл "PHP"	2 КБ
Post.php	05.06.2016 18:10	Файл "PHP"	3 КБ
User.php	05.06.2016 18:16	Файл "PHP"	3 КБ

Рисунок 3.12 - Структура папки models після генерації всіх моделей.

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:

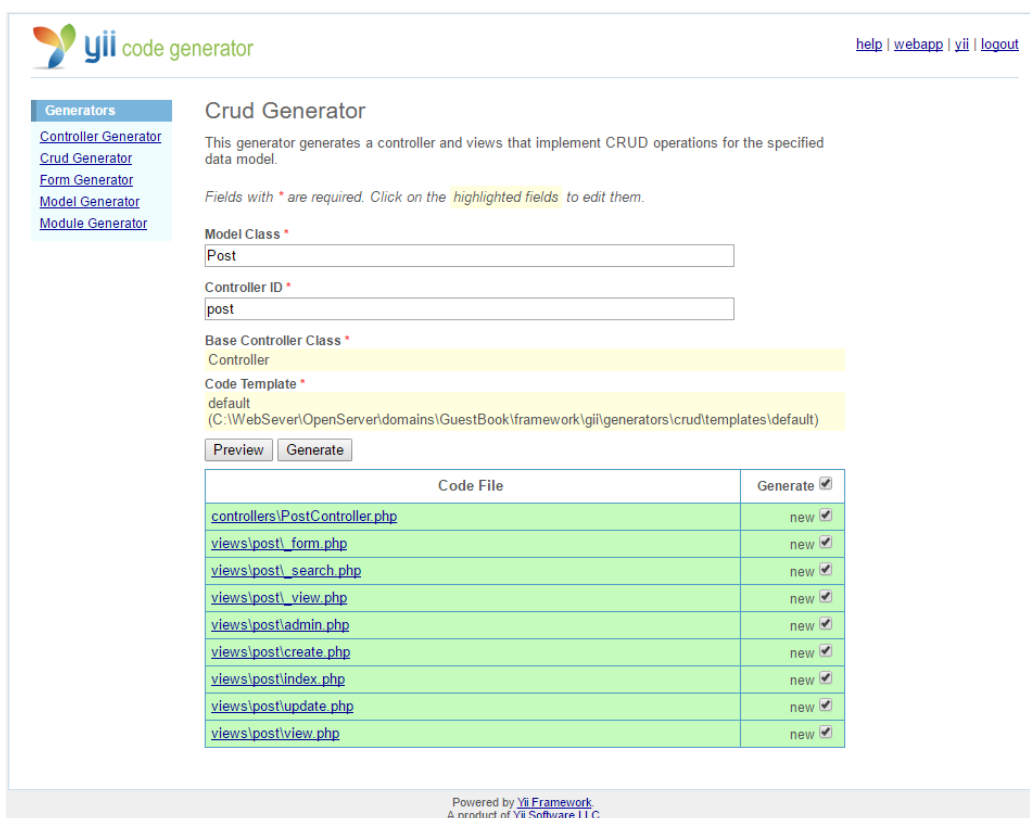


Рисунок 3.13 - Генерація CRUD контролера для моделі Post.

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

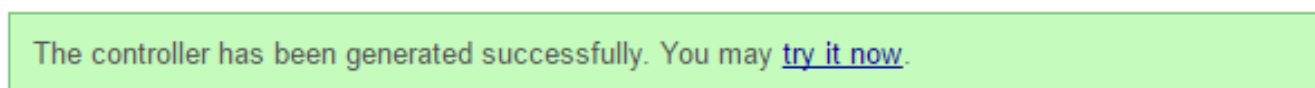


Рисунок 3.14 - Успішно згенерований контролер.

Перейшовши по посиланню “try it now”, можна побачити згенероване представлення:



Рисунок 3.15 - Згенероване представлення для моделі Post.

Аналогічно згенеруємо контролери і представлення для решти моделей. В результаті ми отримуємо таку структуру папок:

Имя	Дата изменения	Тип	Размер
AboutkpiController.php	27.05.2016 0:33	Файл "PHP"	3 КБ
CommentController.php	26.05.2016 23:26	Файл "PHP"	4 КБ
FaqController.php	30.05.2016 17:37	Файл "PHP"	6 КБ
InformationController.php	27.05.2016 0:01	Файл "PHP"	3 КБ
PostController.php	30.05.2016 13:12	Файл "PHP"	5 КБ
SiteController.php	05.06.2016 18:49	Файл "PHP"	2 КБ

Рисунок 3.16 - Структура папки controllers.

Имя	Дата изменения	Тип
aboutkpi	27.05.2016 0:18	Папка с файлами
comment	25.05.2016 23:13	Папка с файлами
faq	30.05.2016 17:28	Папка с файлами
information	26.05.2016 23:52	Папка с файлами
layouts	25.05.2016 15:15	Папка с файлами
post	26.05.2016 22:04	Папка с файлами
site	05.06.2016 18:48	Папка с файлами

Рисунок 3.17 - Структура папки views.

Як зображено на рисунку, всі представлення розміщені в окремих папках, тому розглянемо одну із них:

Имя	Дата изменения	Тип	Размер
_form.php	05.06.2016 19:24	Файл "PHP"	2 КБ
_search.php	05.06.2016 19:24	Файл "PHP"	1 КБ
_view.php	05.06.2016 19:24	Файл "PHP"	1 КБ
admin.php	05.06.2016 19:24	Файл "PHP"	2 КБ
create.php	05.06.2016 19:24	Файл "PHP"	1 КБ
index.php	05.06.2016 19:24	Файл "PHP"	1 КБ
update.php	05.06.2016 19:24	Файл "PHP"	1 КБ
view.php	05.06.2016 19:24	Файл "PHP"	1 КБ

Рисунок 3.18 - Структура папки представлення Post.

Аналогічну структуру має і решта папок з представленнями.

Після виконання цих етапів було успішно згенеровано каркас для сайту “Питання – відповіді КПП - Телеком”. Звісно, він ще потребує допрацювання, але основа для створення сайту вже закладена.

### 3.7. Зміна стандартного інтерфейсу web-сайту

На даному етапі сайт має такий дизайн:

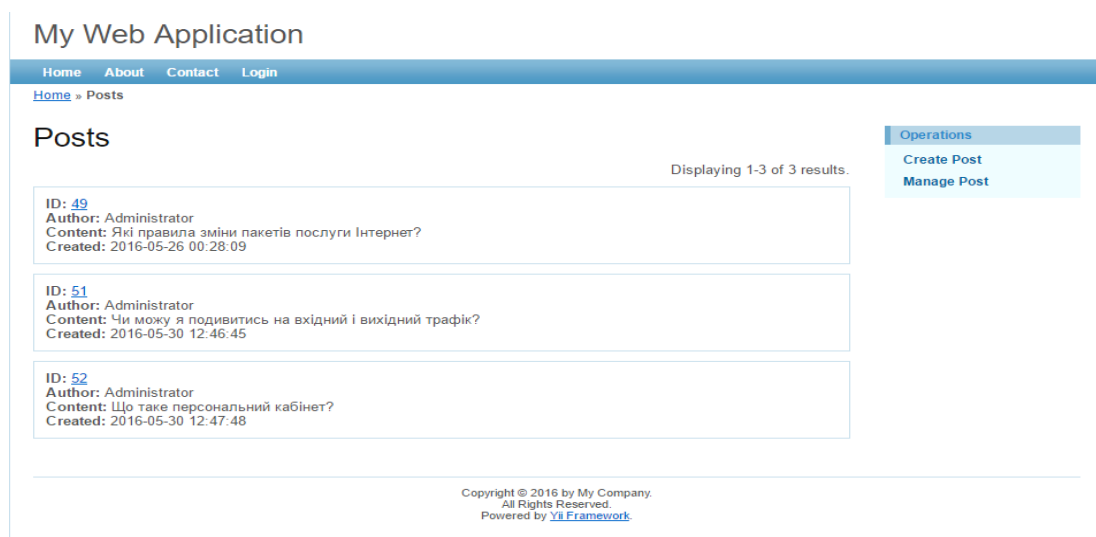


Рисунок 3.19- Сторінка з питаннями.

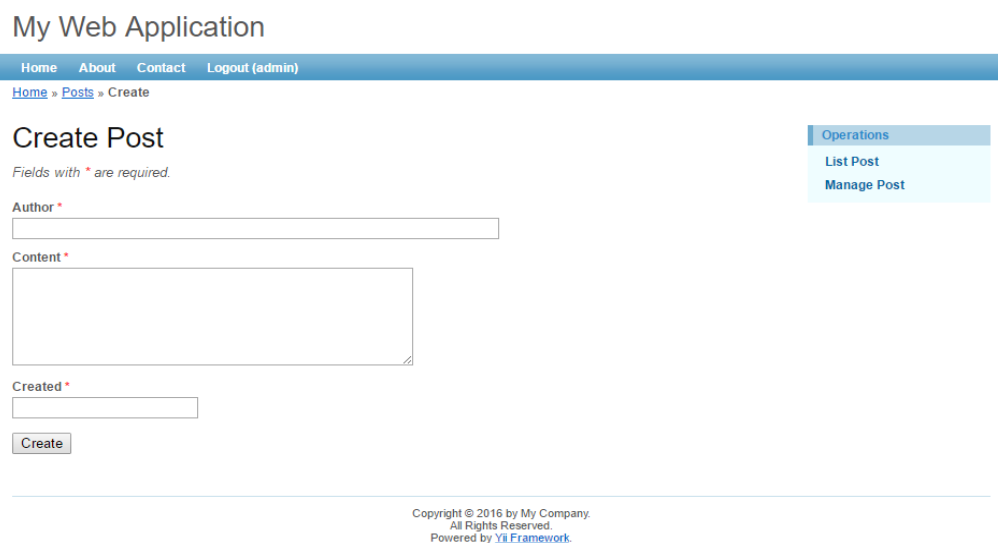


Рисунок 3.20 - Сторінка створення питання.

## My Web Application

[Home](#)
[About](#)
[Contact](#)
[Logout \(admin\)](#)

[Home](#) » [Posts](#) » [49](#) » Update

### Update Post 49

Fields with \* are required.

Author \*

Administrator

Content \*

Які правила зміни пакетів послуги Інтернет?

Created \*

2016-05-26 00:28:09

Save

Operations

List Post

Create Post

View Post

Manage Post

Copyright © 2016 by My Company.  
All Rights Reserved.  
Powered by [Yii Framework](#).

### Рисунок 3.21 - Сторінка редагування питання.

## My Web Application

[Home](#)
[About](#)
[Contact](#)
[Logout \(admin\)](#)

[Home](#) » [Informations](#)

### Informations

Displaying 1-5 of 5 results.

Operations

Create Information

Manage Information

ID: 1

Header Inf: Структура мережі

Content Inf: Душа моя озарена неземной радостью, как эти чудесные весенние утра, которыми я наслаждаюсь от всего сердца. Я совсем один и блаженствую в здешнем краю, словно созданном для таких, как я. Я так счастлив, мой друг, так упоен ощущением покоя, что искусство мое страдает от этого. Ни одного штриха не мог бы я сделать, а никогда не был таким большим художником, как в эти минуты. Когда от милой моей долины поднимается пар и полдневное солнце стоит над непроницаемой чащей темного леса и лишь редкий луч проскальзывает в его святая святых, а я лежу в высокой траве у быстрого ручья и, прильнув к земле, вижу тысячи всевозможных былинки и чувствую, как близок моему сердцу крошечный мирок, что снует между стебельками, наблюдаю эти неисчислимые, непостижимые разновидности червяков и мошек и чувствую близДуша моя озарена неземной радостью, как эти чудесные весенние утра, которыми я наслаждаюсь от всего сердца. Я совсем один и жником, как в эти минуты. Когда от милой моей долины поднимается пар и полдневное солнце стоит над непроницаемой чащей темного леса и лишь редкий луч проскальзывает в его святая святых, а я лежу в высокой траве у быстрого ручья и, прильнув к земле, вижу тысячи всевозможных былинки и чувствую, как близок моему сердцу крошечный мирок, что снует между стебельками, наблюдаю эти неисчислимые, непостижимые разнообразно<br/>сти червяков и мошек и чувствую близДуша моя озарена неземной радостью, как эти чудесные весенние утра, которыми я наслаждаюсь от всего сердца. Я совсем один и жником, как в эти минуты. Когда от милой моей долины поднимается пар и полдневное солнце стоит над непроницаемой чащей темного леса и лишь редкий луч проскальзывает в его святая святых, а я лежу в высокой траве у быстрого ручья и, прильнув к земле, вижу тысячи всевозможных былинки и чувствую, как близок моему сердцу крошечный мирок, что снует между стебельками, наблюдаю эти неисчислимые, непостижимые разновидности червяков и мошек и чувствую близДуша моя озарена неземной радостью, как эти чудесные весенние утра, которыми я наслаждаюсь от всего сердца. Я совсем один и

ID: 2

Header Inf: Апаратне забезпечення

Content Inf: Душа моя озарена неземной радостью, как эти чудесные весенние утра, которыми я наслаждаюсь от всего сердца. Я совсем один и блаженствую в здешнем краю, словно созданном для таких, как я. Я так счастлив, мой друг, так упоен ощущением покоя, что искусство мое страдает от этого. Ни одного штриха не мог бы я сделать, а никогда не был таким большим художником, как в эти минуты. Когда от милой моей долины поднимается пар и полдневное солнце стоит над непроницаемой чащей темного леса и лишь редкий луч проскальзывает в его святая святых, а я лежу в высокой траве у быстрого ручья и, прильнув к земле, вижу тысячи всевозможных былинки и чувствую, как близок моему сердцу крошечный мирок, что снует между стебельками, наблюдаю эти неисчислимые, непостижимые разновидности червяков и мошек и чувствую близ<br/>леса и лишь редкий луч проскальзывает в его святая святых, а я лежу в высокой траве у быстрого ручья и, прильнув к земле, вижу тысячи всевозможных былинки и чувствую, как близок мое<br/>му сердцу крошечный мирок, что снует между стебельками, наблюдаю эти неисчислимые, непостижимые разновидности червяков и мошек и чувствую близ леса и лишь редкий луч проскальзывает в его святая святых, а я лежу в высокой траве у быстрого ручья и, прильнув к земле, вижу тысячи всевозможных былинки и чувствую, как близок моему сердцу крошечный мирок, что снует между стебельками, наблюдаю эти неисчислимые, непостижимые разновидности червяков и мошек и чувствую близлеса и лишь редкий луч проскальзывает в его святая святых, а я лежу в высокой траве у быстрого ручья и, прильнув к земле, вижу тысячи всевозможных былинки и чувствую, как близок моему <br/>сердцу крошечный мирок, что снует между стебельками, наблюдаю эти неисчислимые, непостижимые разновидности червяков и мошек и чувствую близ

### Рисунок 3.22 - Розділ сайту “Інформація”.

Із цих зображень зрозуміло , що 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).

### 3.7.1. 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:

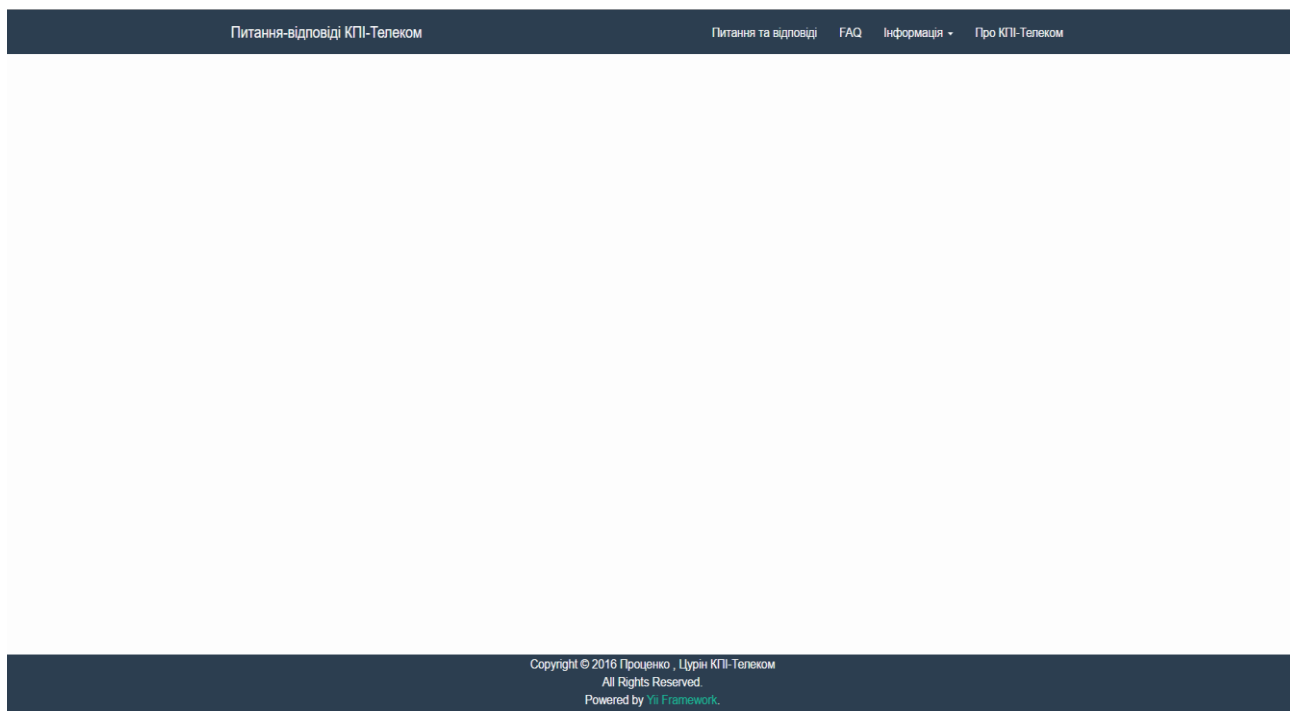


Рисунок 3.23 - Header та footer сайту після допрацювання.

### 3.7.2. Зміна дизайну контентної частини сайту

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






 _post	25.05.2016 15:36	Папка с файлами	
 _post.php	26.05.2016 11:45	Файл "PHP"	2 КБ
 create.php	30.05.2016 15:36	Файл "PHP"	6 КБ
 index.php	30.05.2016 17:33	Файл "PHP"	2 КБ
 update.php	30.05.2016 14:06	Файл "PHP"	5 КБ

Рисунок 3.24 - Папка з представленнями для моделі Post після допрацювання.



Имя	Дата изменения	Тип	Размер
 create.php	27.05.2016 1:59	Файл "PHP"	3 КБ
 update.php	27.05.2016 2:00	Файл "PHP"	3 КБ

Рисунок 3.25 - 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 відповіді. Так зроблено тому, що відповіді 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:

Питання-відповіді КПІ-Телеком

Питання та відповіді FAQ Інформація Про КПІ-Телеком

### Задати питання

**Administrator** 2016-05-30 12:47:48  
Що таке персональний кабінет?

**Administrator** 2016-05-30 12:48:08  
Персональний кабінет - Ваша персональна захищена веб-сторінка, де можна отримувати відомості про замовлені послуги, керувати послугами, дивитись свій трафік, задавати нам запитання і т.д.

**Administrator** 2016-05-30 12:46:45  
Чи можу я подивитись на вхідний і вихідний трафік?

**Administrator** 2016-05-30 16:35:08  
Так, це можна зробити. Для цього необхідно зайти у свій "Персональний кабінет", у ньому Ви побачите опцію "Статистика". Клікнувши на неї Ви зможете переглянути свій вхідний і вихідний трафік щоденно за останні два місяці.

**Administrator** 2016-05-26 00:28:09  
Які правила зміни пакетів послуги Інтернет?

**Administrator** 2016-05-30 16:46:27

Copyright © 2016 Проценко , Цурін КПІ-Телеком  
All Rights Reserved.  
Powered by [Yii Framework](#).

Рисунок 3.26 - Сторінка “Питання-відповіді”.

Питання-відповіді КПІ-Телеком

Питання та відповіді FAQ Інформація Про КПІ-Телеком

Вимоги до змісту повідомлення:

1. Максимально 1000 символів!
2. Прямі та непрямі образи кого-небудь, зокрема, політиків, чиновників, адміністраторів ресурсу, користувачів ресурсу, у тому числі за національною, етнічною, расовою або релігійною приналежністю, а також шовіністичні висловлювання.
3. Нецензурні вирази, висловлювання порнографічного, еротичного чи сексуального характеру.
4. Будь-яка образлива поведінка по відношенню до авторів статей і всіх учасників ресурсу.
5. Реклама, комерційні повідомлення, а також повідомлення, які не мають інформаційного навантаження і не стосуються тематики ресурсу.
6. Будь-які повідомлення, які заборонені законодавством України.

**Author \***

**Content \***

**qamesox** Введіть текст з картинки

[Додати запитання](#)

Copyright © 2016 Проценко , Цурін КПІ-Телеком  
All Rights Reserved.  
Powered by [Yii Framework](#).

Рисунок 3.27 - Сторінка створення питання.

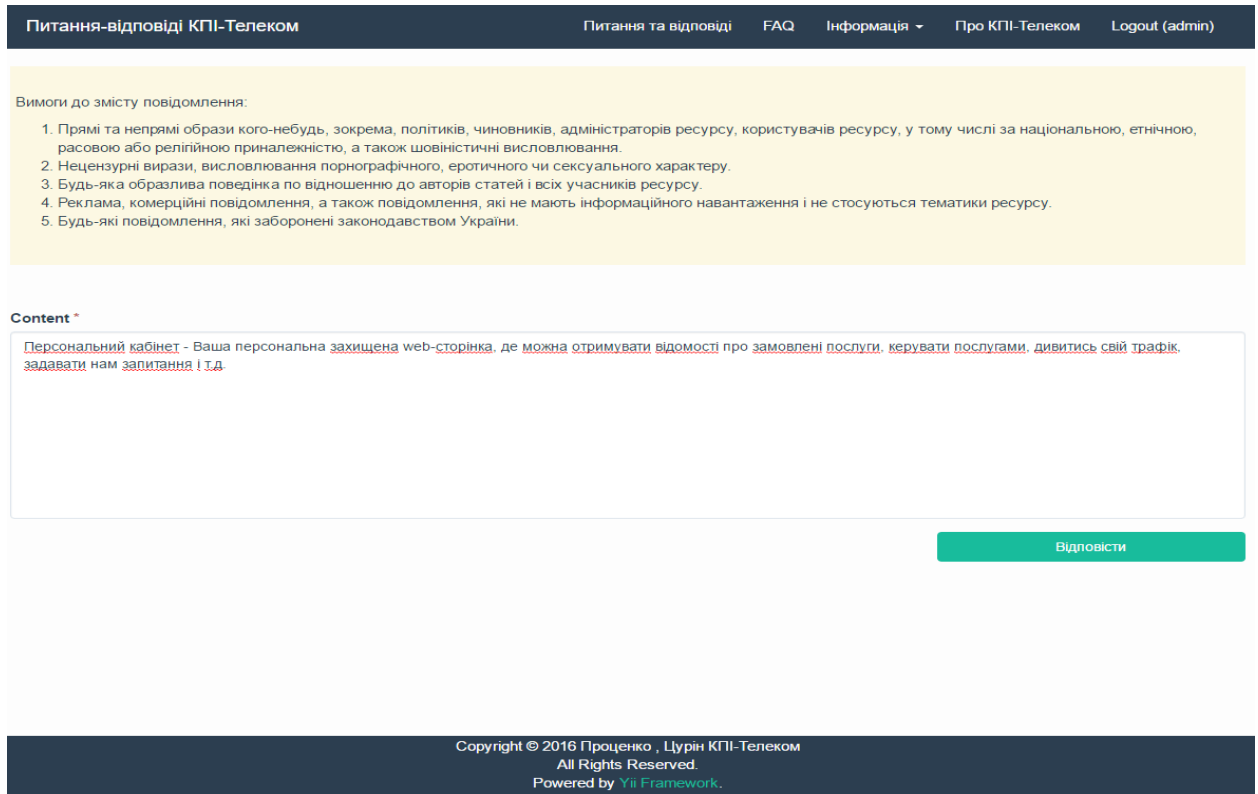


Рисунок 3.28 - Сторінка створення відповіді на питання.

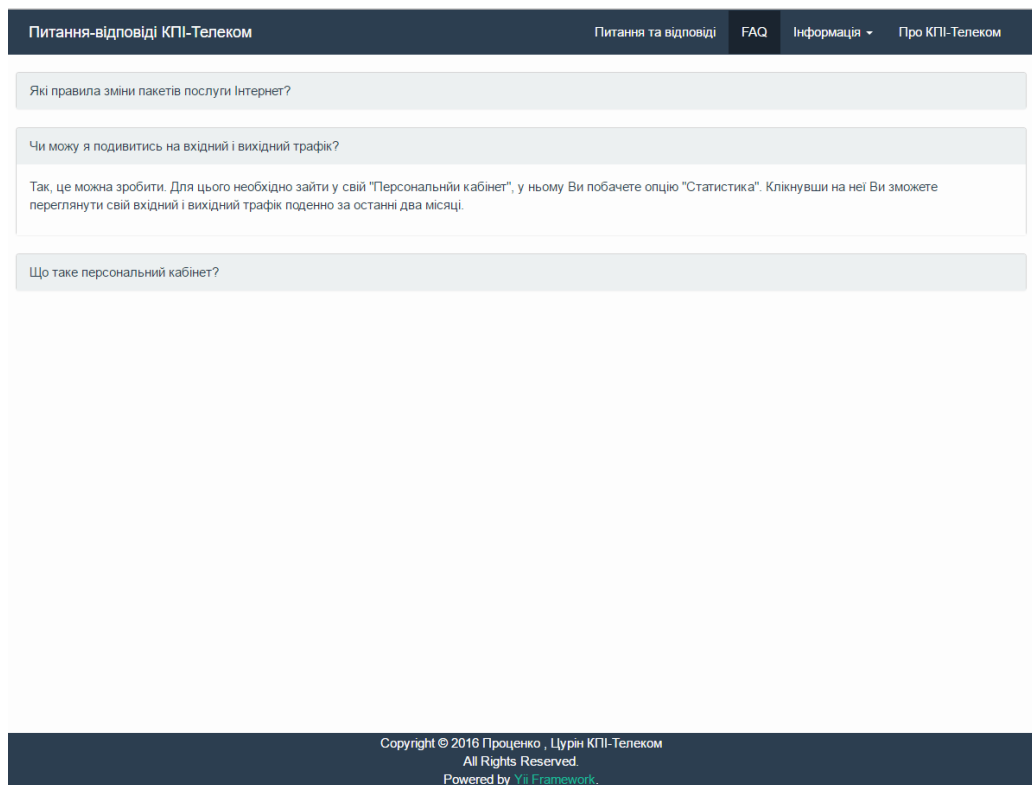


Рисунок 3.29 - Сторінка частих питань(FAQ).

Також була реалізована відповідь посиланням на іншу відповідь. За допомогою JavaScript було реалізовано підсвічування відповіді, на яку веде посилання:

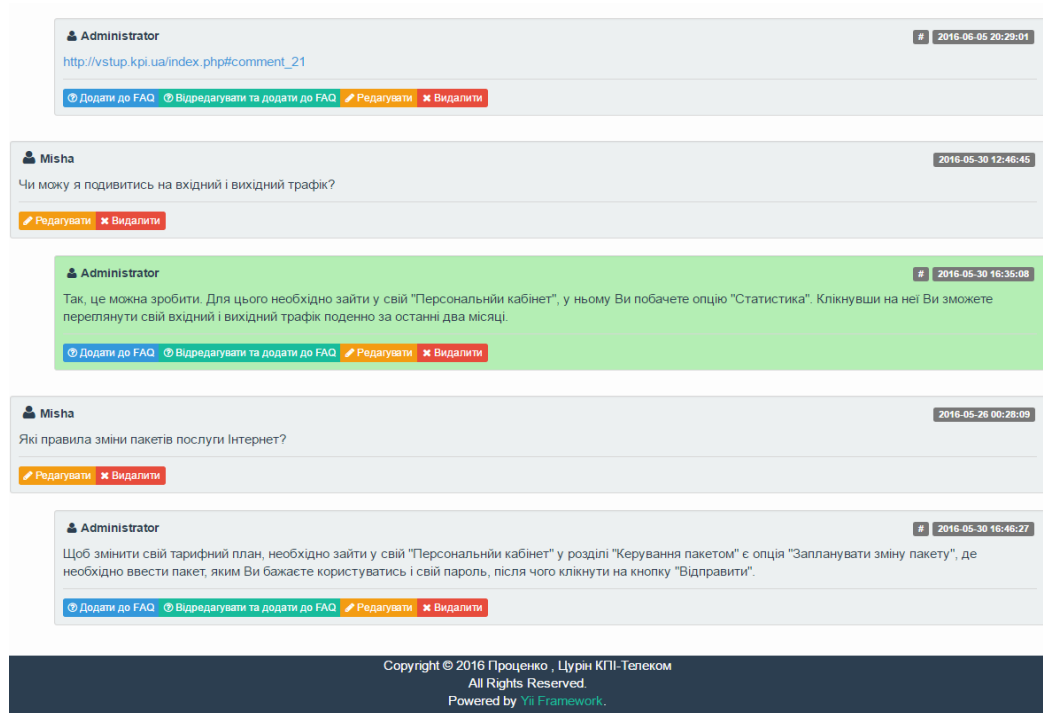


Рисунок 3.30 - Підсвічування відповіді.

### 3.8. Розміщення сайту на хостинг

Для того щоб створений сайт був доступний користувачам з мережі Інтернет, його буде розміщено на хостинг за адресою: `vstup.kpi.ua`. Перенесення сайту на хостинг полягає в копіюванні його файлів з папки локального серверу на сервер хостингу, а також перенесенні інформації з локальної БД до БД хостингу та редагуванні файлу налаштування підключення до БД `database.php`, а саме встановленні паролю та імені користувача. Після проведення цих дій сайт доступний в мережі Інтернет за адресою: `vstup.kpi.ua`.

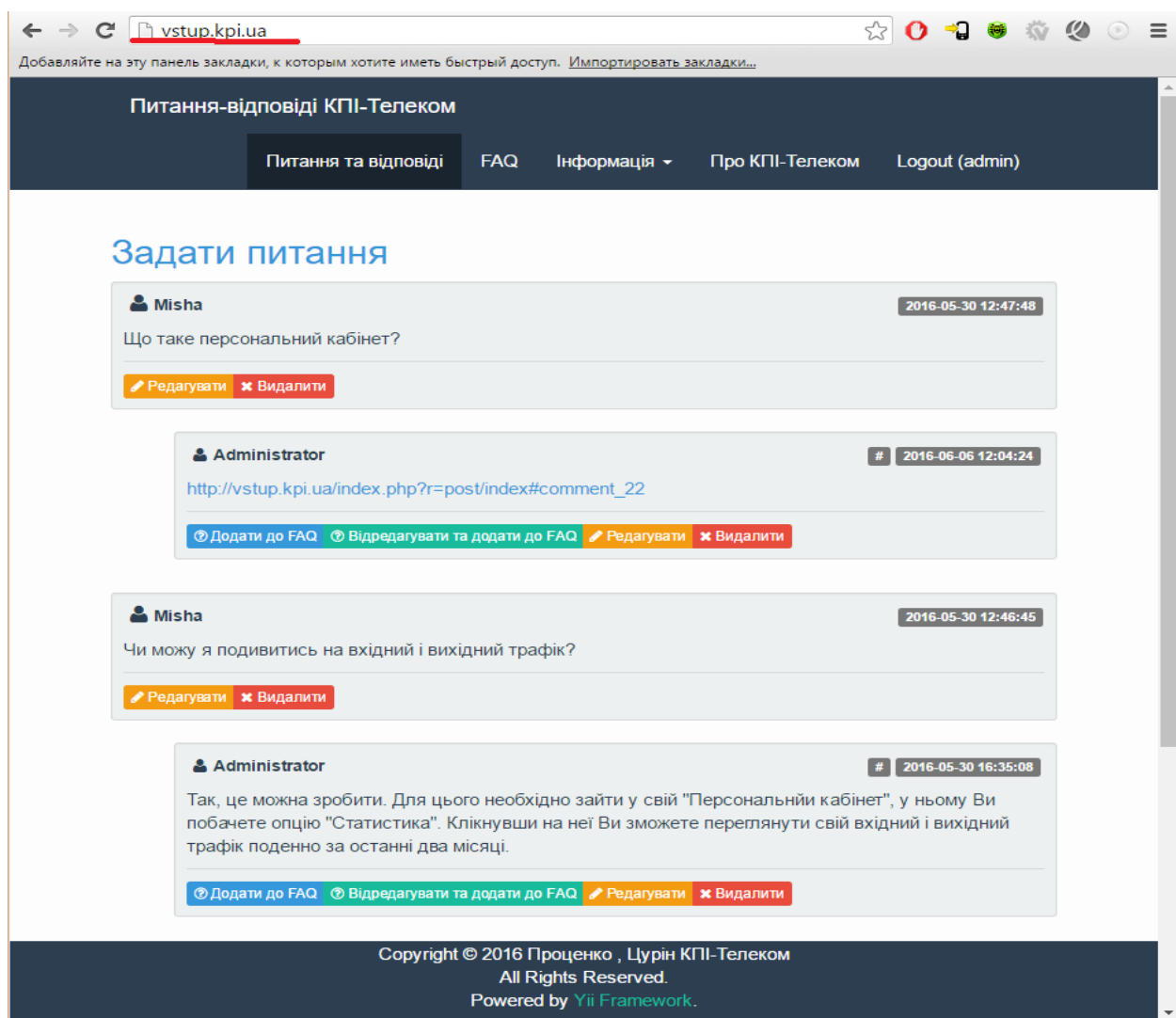


Рисунок 3.31 - Створений сайт завантажений на хостинг

На цьому роботу над сайтом завершено.

### 3.9. Висновки до розділу 3

Як завершальний етап у роботі з фреймворками, було створено web-сайт за допомогою Yii 1.1., для якого характерні усі основні особливості сайту, побудованого за допомогою обраного Frameworku. Він має трирівневу 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.

Завершальним етапом створення сайту стало завантаження його на хостинг . Тепер сайт доступний з мережі Інтернет і скористуватися ним може будь-який користувач.

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

## **4. ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ**

У даному розділі проводиться оцінка основних характеристик програмного продукту, призначеного для створення сайту «Питання-відповіді КПП-Телеком». Сайт буде створюватися за допомогою web-фреймворку .

Фінальний сайт призначений для відповідей на питання стосовно роботи інтернет-провайдера КПП-Телеком. 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 двох група: 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-економічного аналізу розробки.

Відповідно до цього необхідно обирати і систему показників якості програмного продукту.

Технічні вимоги до продукту наступні:

– програмний продукт повинен функціонувати на персональних комп'ютерах із стандартним набором компонент;

– забезпечувати цілодобову можливість клієнтам КПП-Телеком поставити своє запитання стосовно послуг, що надає КПП-Телеком;

– 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 1.1;
- 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).

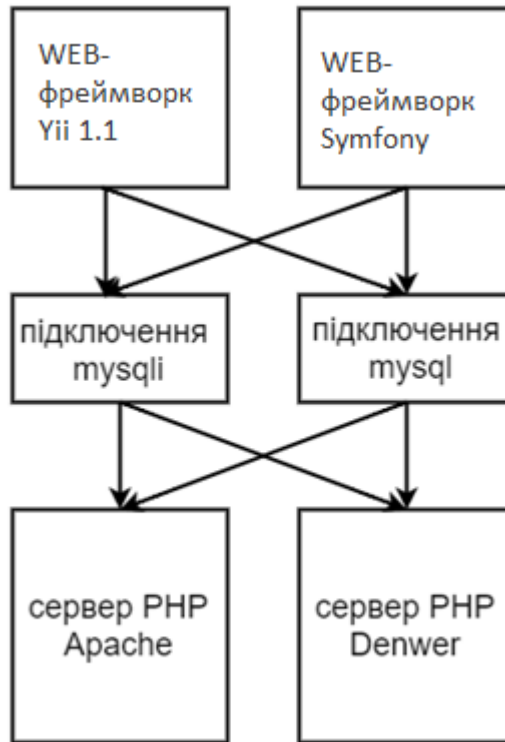


Рисунок 4.1 – Морфологічна карта

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

Таблиця 4.1 – Позитивно-негативна матриця

Основні функції	Варіанти реалізації	Переваги	Недоліки
F1	A	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.	Не дуже потужний роутинг і наявність великої кількості віджетів, на настройку яких витрачається багато часу.
	B	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 проєктів, відсутність російської документації.
F2	A	Більша оптимізація ніж у аналога, більша швидкість доступу.	Неможливе з'єднання з старими версіями баз даних.
	B	Підтримка старих версій MySQL баз даних та PHP.	Менша швидкість.
F3	A	Можливість використання різних версій PHP.	PHP та MySQL необхідно встановлювати додатково.
	B	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:

1. F1a – F2б – F3а
2. F1б – F2б – F3а

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 мати 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	середні	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.

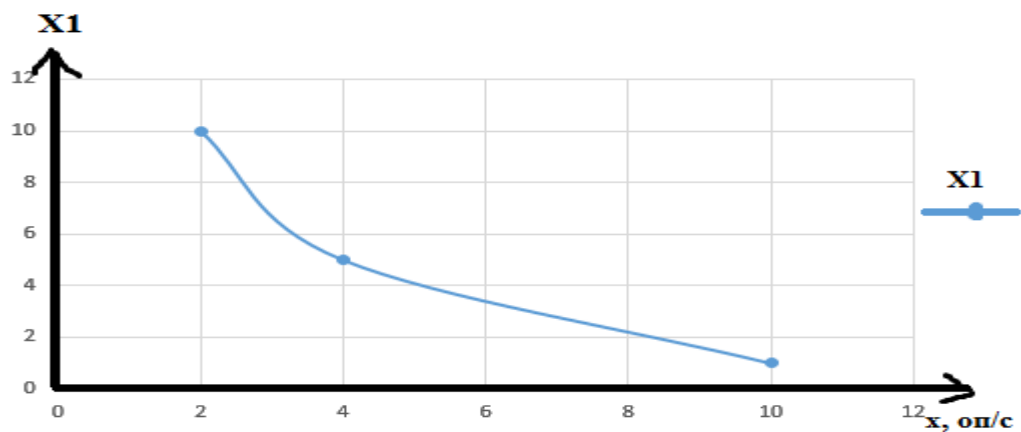


Рисунок 4.2 – X1, завантаженість підключення до Інтернету

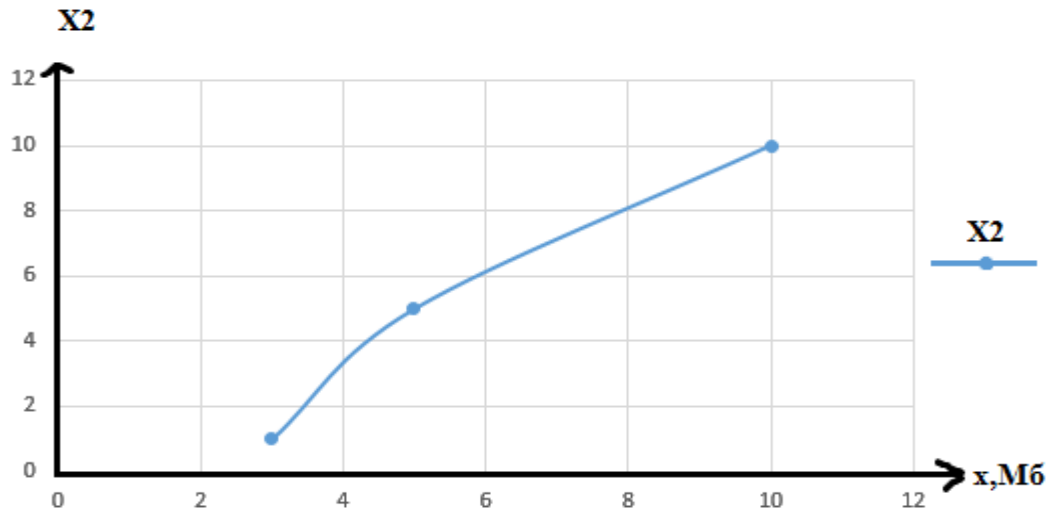


Рисунок 4.3 –  $X_2$ , об'єм оперативної пам'яті, що використовується

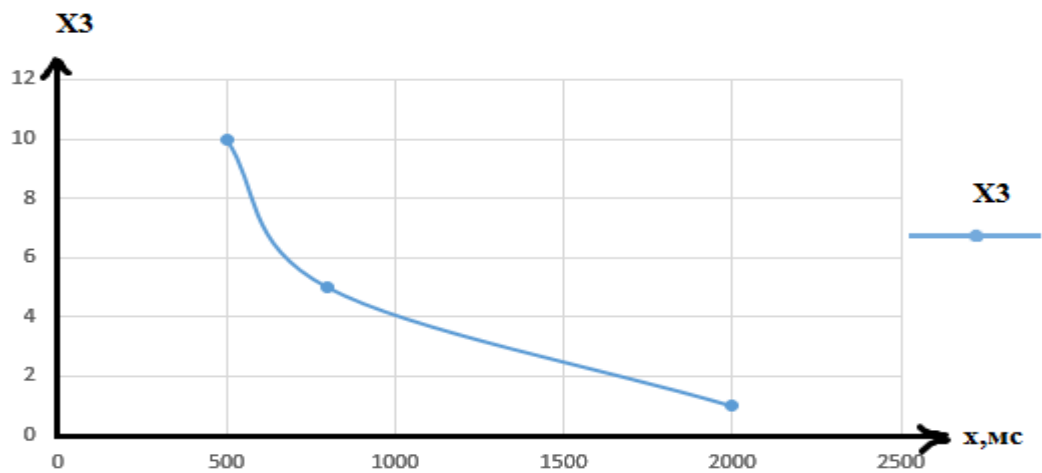


Рисунок 4.4 –  $X_3$ , час необхідний на підключення до медіа серверу

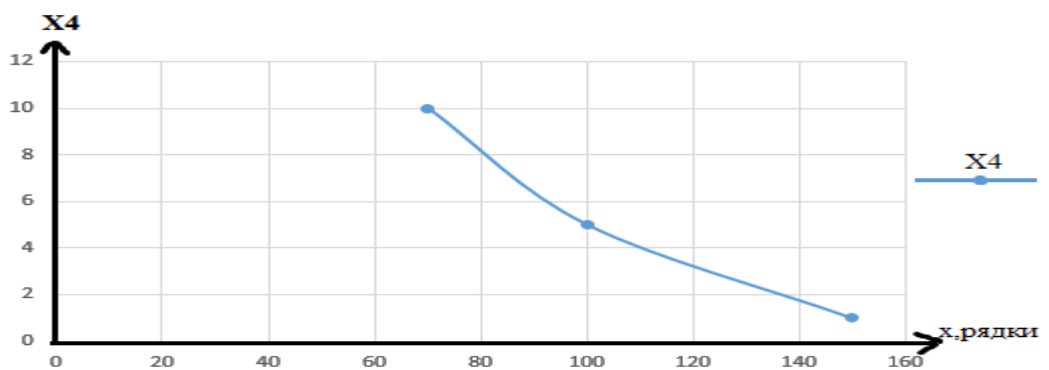


Рисунок 4.5 –  $X_4$ , кількість функцій, доступних для віддаленого доступу

### **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 $\Delta_i$	$\Delta_i^2$
			1	2	3	4	5	6	7			
X1	Час відповіді на запит клієнта	С	1	2	1	2	1	1	1	9	-8,5	72,25
X2	Кількість модулів на сайті	Кількість модулів	4	3	4	3	4	4	3	25	7,5	56,25
X3	Об'єм даних, що передається через Інтернет	Кб	2	1	2	1	2	3	2	13	-4,5	20,25
X4	Об'єм оперативної пам'яті, що використовується	Мб	3	4	3	4	3	2	4	23	5,5	30,25
	Разом		10	10	10	10	10	10	10	70	0	179

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

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

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70, \quad (4.1)$$

де  $N$  – число експертів,  $n$  – кількість параметрів;

б) середня сума рангів:



$$T = \frac{1}{n} R_{ij} = 17,5 \quad (4.2)$$

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

$$\Delta_i = R_i - T \quad (4.3)$$

Сума відхилень по всіх параметрах повинна дорівнювати 0;

г) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^N \Delta_i^2 = 179. \quad (4.4)$$

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3-n)} = \frac{12 \cdot 179}{7^2(4^3-4)} = 0,73 > W_k = 0,67 \quad (4.5)$$

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

Скориставшись результатами ранжування, 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:

$$a_{ij} = \begin{cases} 1,5 & \text{при } X_i > X_j \\ 1.0 & \text{при } X_i = X_j \\ 0.5 & \text{при } X_i < X_j \end{cases} \quad (4.6)$$

З отриманих числових оцінок переваги складемо матрицю  $A = \| a_{ij} \|$ . Для кожного параметра зробимо розрахунок вагомості  $K_{ei}$  за наступними формулами:

$$K_{vi} = \frac{b_i}{\sum_{i=1}^n b_i}, \text{ де } b_i = \sum_{j=1}^N a_{ij}. \quad (4.7)$$

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:

$$K_{vi} = \frac{b'_i}{\sum_{i=1}^n b'_i}, \text{ де } b'_i = \sum_{j=1}^N a_{ij} b_j. \quad (4.8)$$

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	$b_i$	$K_{Bi}$	$b_i^1$	$K_{Bi}^1$	$b_i^2$	$K_{Bi}^2$
X1	1,0	1,5	1,5	1,5	5,5	0,344	21,25	0,360	77,875	0,361
X2	0,5	1,0	0,5	0,5	2,5	0,156	9,25	0,157	34,125	0,158
X3	0,5	1,5	1,0	1,5	4,5	0,281	16,25	0,275	59,125	0,274
X4	0,5	1,5	0,5	1,0	3,5	0,219	12,25	0,208	44,875	0,207
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 4.6):

$$K_K(j) = \sum_{i=1}^n K_{Bi,j} B_{i,j}, \quad (4.9)$$

де  $n$  – кількість параметрів;  $K_{gi}$  – 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

Основні функції	Варіант реалізації функції	Параметри	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	А	X2	5	5	0,158	0,79
		X3	600	7	0,274	1,918
	Б	X2	8	8	0,158	1,264
		X3	1500	2,5	0,274	0,685
F2	Б	X1	3	6,5	0,361	2,35
F3	А	X4	100	5	0,207	1,035

За даними з таблиці 4.6 за формулою

$$K_K = K_{TY}[F_{1k}] + K_{TY}[F_{2k}] + \dots + K_{TY}[F_{zk}], \quad (4.10)$$

визначаємо рівень якості кожного з варіантів:

$$K_{K1} = 0,79 + 1,918 + 2,35 + 1,035 = 6,093$$

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

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

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

де  $T_p$  – 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:  $K_{\text{п}} = 0.81$ . Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для обох завдань рівний 1:  $K_{\text{СК}} = 1$ . Оскільки при розробці кожного з завдань використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта  $K_{\text{СТ}} = 0.8$ . 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,  $K_P = 0.5$ ,  $K_M = 1$ ,  $K_{СК} = 1$ ,  $K_{СТ} = 0.8$ ,  $K_{СТ.М} = 1.6$  (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 однакою 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.

$$C_{\text{ч}} = \frac{9000}{21 \cdot 8} = 53.57 \text{ грн.}$$

Тоді, розрахуємо заробітну плату за формулою

$$C_{\text{ЗП}} = C_{\text{ч}} \cdot T_i \cdot K_{\text{д}}, \quad (4.13)$$

де  $C_{\text{ч}}$  – величина погодинної оплати праці програміста;  $T_i$  – трудомісткість відповідного завдання;  $K_{\text{д}}$  – норматив, який враховує додаткову заробітну плату.

Зарплата розробника:

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

Відрахування на єдиний соціальний внесок становить 22%:

$$C_{\text{ВІД}} = C_{\text{ЗП}} \cdot 0.22 = 18279.28 \cdot 0.22 = 4021.44 \text{ грн.} \quad (4.14)$$

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:

$$C_T = 12 \cdot M \cdot K_3 = 12 \cdot 9000 \cdot 0,2 = 21600 \text{ грн.} \quad (4.15)$$

З урахуванням додаткової заробітної плати:

$$C_{\text{ЗП}} = C_T \cdot (1 + K_3) = 21600 \cdot (1 + 0.2) = 25920 \text{ грн.} \quad (4.16)$$

Відрахування на єдиний соціальний внесок:

$$C_{\text{ВІД}} = C_{\text{ЗП}} \cdot 0.22 = 25920 \cdot 0,22 = 5702.4 \text{ грн.} \quad (4.17)$$

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

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

де  $K_{\text{ТМ}}$ – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;  $K_A$ – річна норма амортизації;  $C_{\text{ПР}}$ – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

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

де  $K_P$ – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

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

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.

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

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_{\text{С}} \cdot K_{\text{З}} \cdot C_{\text{ЕН}} = 1706,4 \cdot 0,5 \cdot 0,2 \cdot 2.0218 = 345,00 \text{ грн.}, \quad (4.21)$$

де  $N_{\text{С}}$  – середньо-споживча потужність приладу;  $K_{\text{З}}$  – коефіцієнт зайнятості приладу;  $C_{\text{ЕН}}$  – тариф за 1 КВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_{\text{Н}} = C_{\text{ПР}} \cdot 0.67 = 12000 \cdot 0,67 = 8040 \text{ грн.} \quad (4.22)$$

Тоді річні експлуатаційні витрати будуть:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_{\text{А}} + C_{\text{Р}} + C_{\text{ЕЛ}} + C_{\text{Н}} \quad (4.23)$$

$$C_{\text{ЕКС}} = 25920 + 5702.4 + 3450 + 690 + 345 + 8040 = 44147.4 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 44147.4 / 1706.4 = 25.87 \text{ грн/час.} \quad (4.24)$$

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

$$C_{\text{М}} = C_{\text{М-Г}} \cdot T \quad (4.25)$$

$$C_{\text{М}} = 25.87 \cdot 284,352 = 7356.19 \text{ грн.};$$

Накладні витрати складають 67% від заробітної плати:

$$C_{\text{Н}} = C_{\text{ЗП}} \cdot 0,67 \quad (4.26)$$

$$C_{\text{Н}} = 18279.28 \cdot 0,67 = 12247.12 \text{ грн.};$$

Отже, вартість розробки ПП за варіантами становить:

$$C_{\text{ПП}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_{\text{М}} + C_{\text{Н}} \quad (4.27)$$

$$C_{\text{ПП}} = 18279.28 + 4021.44 + 7356.19 + 12247.12 = 41904 \text{ грн.};$$



## 4.5. Вибір кращого варіанта ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{\text{TEP}j} = K_{Kj} / C_{\Phi j}, \quad (4.28)$$

$$K_{\text{TEP}1} = 6.093 / 41904 = 0,145 \cdot 10^{-3};$$

$$K_{\text{TEP}2} = 5.334 / 41904 = 0,127 \cdot 10^{-3};$$

Оскільки обидва варіанти однаково трудомісткі, найбільш ефективним є перший варіант реалізації програми (через більший рівень якості) з коефіцієнтом техніко-економічного рівня  $K_{\text{TEP}1} = 0,145 \cdot 10^{-3}$ .

## 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  $K_{TEP1} = 0,145 \cdot 10^{-3}$ ;

Цей варіант реалізації програмного продукту має такі параметри:

- WEB-фреймворк Yii 1.1;
- 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

В дипломній роботі було розглянуто аналіз фреймворків, як засобів створення WEB-додатків. Після їх детального огляду можна сказати, що 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 фреймворку 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 убезпечитися 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 «Питання-відповіді КПП-Телеком», фреймворк Yii було обрано для створення проекту. Вибір був здійснений згідно з критеріями які були оглянуті в розділі 1. Головними критеріями при його виборі стали: підтримка інтернаціоналізації, низький поріг входження, захист

від SQL ін'єкцій та велика кількість російськомовної та україномовної документації. Як підсумок варто додати, що Yii також має багато додаткових можливостей які виділяють цей фреймворк на тлі інших і роблять розробку з його допомогою дуже швидкою і приємною .

Завершальним етапом по роботі з фреймворками, стало створено web-сайту за допомогою Yii 1.1., для якого характерні усі основні особливості сайту, побудованого за допомогою обраного Frameworku. Він має трирівневу 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.

Фінальним етапом створення сайту стало завантаження його на хостинг . Тепер сайт доступний з мережі Інтернет і скористуватися ним може будь-який користувач.

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

Також було створено завдання і приклад для лабораторної роботи №2б з курсу WEB – проектування . Яка надалі буде використовуватися для перевірки якості знань студентів з курсу .

Якщо підсумувати всі описані вище стадії роботи над то можна зробити таке заключення,що в роботі було проведено аналіз критеріїв, що дозволив виявити найбільш оптимальний інструмент для створення сайту «Питання-відповіді КПП-Телеком». Результати роботи можуть бути використані веб-розробниками при проектуванні і розробці сайтів. І дозволять уникнути можливого помилкового вибору фреймворка для свої майбутніх проектів. А також допоможуть новачкам з орієнтуватися у великій кількості фреймворків і вибрати той , що найкраще підійде для створення їхнього першого WEB-додатку.

**ПЕРЕЛІК ПОСИЛАНЬ :**

1. КОЛИЧЕСТВО ПОЛЬЗОВАТЕЛЕЙ ИНТЕРНЕТА 15 ЛЕТ НАЗАД И СЕГОДНЯ [Электронный ресурс] – Режим доступа до ресурсу: <http://bit.ua/2015/05/internet-changes/>.
2. Интернет в цифрах: мировая статистика за 2012 год [Электронный ресурс] – Режим доступа до ресурсу: <https://habrahabr.ru/company/webnames/blog/166457/>.
3. Digital, Social and Mobile in 2015 report indicates [Электронный ресурс] – Режим доступа до ресурсу: <http://wearesocial.com/sg/special-reports/digital-social-mobile-2015>.
4. Веб-фреймворки и с чем их едят [Электронный ресурс] – Режим доступа до ресурсу: <http://iwsn.ru/blog/show/veb-freymvorki-i-s-chem-ih-edyat>.
5. Фреймворки в веб-разработке [Электронный ресурс] – Режим доступа до ресурсу: : [https://web-creator.ru/articles/about\\_frameworks](https://web-creator.ru/articles/about_frameworks).
6. Результаты тестирования шести ведущих фреймворков на производительность [Электронный ресурс] – Режим доступа до ресурсу: <http://www.alrond.com/ru/2007/jan/25/rezultaty-testirovaniya-6-frameworks/>.
7. Адаптивные CSS-фреймворки, сетки, классы видимости [Электронный ресурс] – Режим доступа до ресурсу: <http://klondike-studio.ru/blog/responsive-css-framework/>.
8. Обзор CSS-фреймворков [Электронный ресурс] – Режим доступа до ресурсу: <http://iantonov.me/page/obzor-css-frejmworkov>.
9. Використання PHP фреймворків в розробці сайту [Электронный ресурс] – Режим доступа до ресурсу: <http://ukrbukva.net/page,5,39718-Ispol-zovanie-PHP-freiyworkov-v-razrabotke-saiyta.html>.
10. Сравнение каркасов веб-приложений [Электронный ресурс] – Режим доступа до ресурсу: [https://ru.wikipedia.org/wiki/Сравнение\\_каркасов\\_веб-приложений](https://ru.wikipedia.org/wiki/Сравнение_каркасов_веб-приложений).

11. Обзоры Web-фреймворков [Электронный ресурс] – Режим доступа до ресурсу: <https://praktikatech.wordpress.com/category/обзоры-web-фреймворков/>
12. Полное руководство по Yii [Электронный ресурс] – Режим доступа до ресурсу: <http://www.yiiframework.com/doc/guide/1.1/ru/index>.
13. Каркас веб-приложений [Электронный ресурс] – Режим доступа до ресурсу: [https://ru.wikipedia.org/wiki/Каркас\\_веб-приложений](https://ru.wikipedia.org/wiki/Каркас_веб-приложений).
14. Что такое фреймворк? [Электронный ресурс] – Режим доступа до ресурсу: <http://www.dbhelp.ru/what-is-framework/page/>.
15. Десять причин избегать тяжеловесных фреймворков, а также лишних зависимостей в проекте [Электронный ресурс] – Режим доступа до ресурсу: <http://eax.me/avoid-frameworks/>.
16. 5 самых популярных фреймворков 2014 года [Электронный ресурс] – Режим доступа до ресурсу: <http://lpgenerator.ru/blog/2016/03/10/5-samyh-populyarnyh-frejmworkov-2014-goda/>.
17. Ruby on Rails [Электронный ресурс] – Режим доступа до ресурсу: <http://www.rubyonrails.ru/>.
18. What is a Web Framework? [Электронный ресурс] – Режим доступа до ресурсу: <https://jeffknupp.com/blog/2014/03/03/what-is-a-web-framework/>.
19. Популярные frontend и backend фреймворки [Электронный ресурс] – Режим доступа до ресурсу: <http://web-diz.com.ua/poleznosti/populyarnye-frontend-i-backend-freymvorki/>.
20. О фреймворках [Электронный ресурс] – Режим доступа до ресурсу: <http://web-elive.com/stati/raznoe/o-frejmworkax/>.
21. Веллинг Л. Разработка веб-приложений с помощью PHP и MySQL / Л. Веллинг, Л. Томсон. – Москва: Вильямс, 2010. – 848 с.
22. Шлоснейгл Д. Профессиональное программирование на PHP / Джордж Шлоснейгл. – Москва: Вильямс, 2006. – 624 с.
23. Кузнецов М. В. PHP 5 на примерах / М. В. Кузнецов, И. В. Симдянов, С.

- В. Голышев. – Санкт-Петербург: БХВ-Петербург., 2005. – 576 с.
24. Кухарчик А. С. PHP: обучение на примерах / А.С.Кухарчик. – Минск: Новое знание, 2004. – 240 с
25. Аткинсон Л. PHP 5. Библиотека профессионала / Л. Аткинсон, З. Сураски. – Москва: Вильямс, 2006. – 944 с.
26. Мазуркевич А. М. PHP: Настольная книга программиста / А. М. Мазуркевич, Д. С. Еловой. – Минск: Новое знание, 2004. – 480 с.
27. Суэринг С. PHP и MySQL. Библия программиста / С. Суэринг, Д. Парк, Т. Конверс. – Москва: Вильямс, 2010. – 912 с.
28. Дронов В. А. PHP 5/6, MySQL 5/6 и Dreamweaver CS4. Разработка интерактивных Web-сайтов / Владимир Александрович Дронов. – Санкт-Петербург: БХВ-Петербург., 2009. – 544 с.

## Додаток А

Налаштування Yii для роботи сайту:

### **main.php:**

```
<?php
return array(
    'basePath'=>dirname(__FILE__).DIRECTORY_SEPARATOR.'..',
    'defaultController' => 'post',
    'homeUrl' => '?r=post/index',
    'name'=>'Питання-відповіді КПП-Телеком',

    // preloading 'log' component
    'preload'=>array('log',
    ),
    'aliases' => array(
'bootstrap' => realpath(__DIR__ . '/../../vendor/crisu83/yiistrap'),
'vendor' => realpath(__DIR__ . '/../../vendor'),
    ),

    // autoloading model and component classes
    'import'=>array(
        'application.models.*',
        'application.components.*',
        'bootstrap.helpers.*',
'bootstrap.behaviors.*',
'bootstrap.components.*',
'bootstrap.form.*',
'bootstrap.widgets.*',
    ),
'modules'=>array(

    'gii'=>array(
        'class'=>'system.gii.GiiModule',
        'password'=>'090909',
        'ipFilters'=>array('127.0.0.1', '::1'),
'generatorPaths' => array('bootstrap.gii'),
```



```

    ),
),
// application components
'components'=>array(
    'user' => array(
        'class' => 'WebUser',
    ),
    'bootstrap' => array(
        'class' => 'bootstrap.components.TbApi',
    ),
    'authManager' => array(
        'class' => 'PhpAuthManager',
        'defaultRoles' => array('guest'),
    ),
    'db'=>require(dirname(__FILE__).'/database.php'),
    'errorHandler'=>array(
        // use 'site/error' action to display errors
        'errorAction'=>'site/error' ,
    ),
    'log'=>array(
        'class'=>'CLogRouter',
        'routes'=>array(
            array(
                'class'=>'CFileLogRoute',
                'levels'=>'error, warning',
            ),
        ),
    ),
);

```

**database.php:**

```
<?php
// This is the database connection configuration.
return array(
    'connectionString' => 'mysql:host=localhost;dbname=guestbook',
    'emulatePrepare' => true,
    'username' => 'root',
    'password' => "",
    'charset' => 'utf8',
);
```

**auth.php:**

```
<?php
return array(
    'guest' => array(
        'type' => CAuthItem::TYPE_ROLE,
        'description' => 'Guest',
        'bizRule' => null,
        'data' => null
    ),
    'administrator' => array(
        'type' => CAuthItem::TYPE_ROLE,
        'description' => 'Administrator',
        'children' => array(
            'Guest',
        ),
        'bizRule' => null,
        'data' => null
    ),
);
?>
```

Налаштування доступу на основі ролей:

### PhpAuthManager.php:

```
<?php
/*Description of PhpAuthManager
Завантажує правила для RBAC из файла applocation/config/auth.php */
class PhpAuthManager extends CPhpAuthManager
{
    public function init()
    {
        if ($this->authFile === null)
        {
            $this->authFile = Yii::getPathOfAlias('application.config.auth') . '.php';
        }
        parent::init();
        if (!Yii::app()->user->isGuest)
        {
            $this->assign(Yii::app()->user->role, Yii::app()->user->id);
        }
    }
}??>
```

### UserIdentity.php:

```
<?php
/**
 * Description of UserIdentity
 * Разширює класс CUserIdentity, и перевизначає метод authenticate.
 */
class UserIdentity extends CUserIdentity
{
    private $_id;
    public function authenticate()
    {
        $record = User::model()->findByAttributes(array('login' => $this->username));
        if ($record === null)
            $this->errorCode = self::ERROR_USERNAME_INVALID;
        else if ($record->password !== $this->password)
```

```
$this->errorCode = self::ERROR_PASSWORD_INVALID;
else
{
    $this->_id = $record->id;
    $this->errorCode = self::ERROR_NONE;
}
return!$this->errorCode;
}
public function getId()
{
    return $this->_id;
}
}
?>
```

**WebUser.php:**

```

<?php
/* Description of WebUser
 * Реалізує отримання ролі з БД при використанні Yii::app()->user->role
 * Для цього розширює клас CWebUser.
 */
class WebUser extends CWebUser
{
    private $_model = null;
    function getRole()
    {
        if ($user = $this->getModel())
        {
            return $user->role;
        }
    }
    private function getModel()
    {
        if (!$this->isGuest && $this->_model === null)
        {
            $this->_model = User::model()->findByPk($this->id, array('select' => 'role'));
        }
        return $this->_model;
    }
}
?>

```

MVC реалізація сторінки сайту “Питання-відповіді”:

Моделі:

**Post.php:**

```

<?php
class Post extends CActiveRecord
{
    public $verifyCode;
    public function tableName()

```

```
{
    return 'post';
}
public function rules()
{
    return array(
        array('author, content', 'required'),
        array('author', 'length', 'max'=>255),
        array('content', 'length', 'max' => 1000, 'min' => 5),
        array('id, author, content, created', 'safe', 'on'=>'search'),
        array('verifyCode', 'captcha', 'allowEmpty' => !extension_loaded('gd')),
    );
}

public function relations()
{
    return array(
    );
}

public function attributeLabels()
{
    return array(
        'id' => 'ID',
        'author' => 'Author',
        'content' => 'Content',
        'created' => 'Created',
    );
}

public function search()
{
    $criteria=new CDbCriteria;

    $criteria->compare('id',$this->id);
    $criteria->compare('author',$this->author,true);
    $criteria->compare('content',$this->content,true);
```

```
$criteria->compare('created',$this->created,true);

return new CActiveDataProvider($this, array(
    'criteria'=>$criteria,));}

public static function model($className=__CLASS__)
{
    return parent::model($className);
}

protected function beforeSave()
{
    if (parent::beforeSave())
    {
        if ($this->isNewRecord)
        {
            $this->created = date('Y-m-d H:i:s');
        }
        return true;
    }
    else
    {
        return false;
    }
}
}
```

## Comment.php:

```
<?php
class Comment extends CActiveRecord
{

    public static function model($className = __CLASS__)
    {
        return parent::model($className);
    }

    public function tableName()
    {
        return 'comment';
    }

    public function rules()
    {
        return array(
            array('content', 'required'),
            array('content', 'length', 'max' => 1000, 'min' => 5),
            array('post_id', 'required', 'on' => 'create'),
        );
    }

    public function relations()
    {
        return array(
            'posts' => array(self::BELONGS_TO, 'Post', 'post_id'),
        );
    }

    public function safeAttributes()
    {
        return array('content', 'post_id', 'created');
    }

    protected function beforeSave()
    {
        if (parent::beforeSave())
```



```

{
    if ($this->isNewRecord)
    {
        $this->created = date('Y-m-d H:i:s');
    }
    return true;
}
else
{
    return false;
}
}
?>

```

Контроллери:

### **PostController.php:**

```

<?php
class PostController extends Controller
{
    public $layout='//layouts/main';
    public function filters()
    {
        return array(
            'accessControl', // perform access control for CRUD operations
        );
    }
    public function accessRules()
    {
        return array(
            array('allow', // allow all users to perform 'index' and 'view' actions
                'actions'=>array('index','view','create','captcha'),
                'users'=>array('*'),
            ),

```

```

        array('allow', // allow admin user to perform 'admin' and 'delete' actions
            'actions'=>array('delete','update'),
            'users'=>array('admin'),
        ),
        array('deny', // deny all users
            'users'=>array('*'),
        ),
    );
}

```

```

public function actions()
{
    return array(
        'captcha' => array(
            'class' => 'CCaptchaAction',
            'width' => '150',
            'height' => '50',
            'backColor' => 0x000000,
            'foreColor' => 0x66ff66,
        ),
    );
}

```

```

public function actionCreate()
{
    $model=new Post;
    // $this->performAjaxValidation($model);
    if(isset($_POST['Post']))
    {
        $model->attributes=$_POST['Post'];
        if($model->save())
            $this->redirect(array('index'));
    }

    $this->render('create',array(
        'model'=>$model,

```

```

        ));
    }
    public function actionUpdate($id)
    {
        $model=$this->loadModel($id);
        if(isset($_POST['Post']))
        {
            $model->attributes=$_POST['Post'];
            if($model->save())
                $this->redirect(array('index','id'=>$model->id));
        }

        $this->render('update',array(
            'model'=>$model,
        ));
    }
    public function actionDelete($id)
    {
        $this->loadModel($id)->delete();

        // if AJAX request (triggered by deletion via admin grid view), we should not redirect the browser
        if(!isset($_GET['ajax']))
            $this->redirect(array('/post/index'));
    }
    public function actionIndex()
    {
        $criteria = new CDbCriteria;
        $criteria->order = 'id DESC';
        $pages = new CPagination(Post::model()->count($criteria));
        $pages->pageSize = 5;
        $pages->applyLimit($criteria);
        $model= Post::model()->findAll($criteria);
        $this->render('index', array(
            'model' => $model,
            'pages' => $pages,

```

```

    ));
}
public function commentSearch($num)
{
    $criteria = new CDbCriteria;
    $criteria->condition = 'post_id=:postID';
    $criteria->params = array('postID' => $num);
    $comment = Comment::model()->find($criteria);
    return $comment;
}
public function loadModel($id)
{
    $model=Post::model()->findByPk($id);
    if($model===null)
        throw new CHttpException(404,'The requested page does not exist. ');
    return $model;
}
protected function performAjaxValidation($model)
{
    if(isset($_POST['ajax']) && $_POST['ajax']=== 'post-form')
    {
        echo CActiveForm::validate($model);
        Yii::app()->end();
    }
}
}

```

### **CommentController.php:**

```

<?php
class CommentController extends CController
{
    public $_postId;
    public $defaultAction = 'create';

    public function filters()
    {

```

```

return array(
    'accessControl', // perform access control for CRUD operations
);
}

public function accessRules()
{
    return array(
        array('allow', // allow all users to perform 'index' and 'view' actions
            'actions'=>array('index','view','create'),
            'users'=>array('*'),
        ),
        array('allow', // allow admin user to perform 'admin' and 'delete' actions
            'actions'=>array('admin','delete','update'),
            'users'=>array('admin'),
        ),
        array('deny', // deny all users
            'users'=>array('*'),
        ),
    );
}

public function actionCreate()
{
    $comment = new Comment('create');
    if (isset($_GET['id']))
    {
        if (isset($_POST['Comment']))
        {
            $comment->post_id = $_GET['id'];
            $comment->attributes = $_POST['Comment'];
            if ($comment->validate())
            {
                if ($comment->save())
                {
                    $this->redirect(array('/post/index', 'id' => $comment->id));
                }
            }
        }
    }
}

```

```

    }
    $this->render('create', array('comment' => $comment));
}
}
public function actionUpdate()
{
    $comment = $this->loadModel();
    if (isset($_POST['Comment']))
    {
        $comment->attributes = $_POST['Comment'];
        if ($comment->save())
            $this->redirect(array('/post/index', 'id' => $comment->post_id));
    }
    $this->render('update', array('comment' => $comment));
}
public function actionDelete()
{
    $this->loadModel()->delete();
    if (!isset($_GET['ajax']))
        $this->redirect(array('/post/index'));
}
public function loadModel()
{
    if ($this->_model === null)
    {
        if (isset($_GET['post_id']))
        {
            $criteria = new CDbCriteria();
            $criteria->condition = 'post_id=:postID';
            $criteria->params = array('postID' => ($_GET['post_id']));
            $this->_model = Comment::model()->find($criteria);
        }
        if ($this->_model === null)
            throw new CHttpException(404, 'The requested page does not exist777');
    }
}

```

```

        return $this->_model;
    }
} ?>

```

Представлення:

### **index.php:**

```

<div id="content" class="row">
    <!-- Задати питання -->
    <h2 class="create-new-post">
        <a href="index.php?r=post/create" >Задати питання</a>
    </h2><!-- create-new-post (End) -->
    <!-- Відповідає за відображення питань -->
    <?php
    if (!empty($model))
        foreach ( $model as $key => $val)
        {
            $this->renderPartial('_post', array(
                'model' => $val,
            )) } ?>
    <!-- Навігація по сторінках -->
    <div class="pagination_container text-center">
        <?php
        $this->widget('CLinkPager', array(
            'pages' => $pages,
            'maxButtonCount' => 3,
            'header' => ",
            'nextPageLabel' => 'Next',
            'prevPageLabel' => 'Prev',
            'selectedPageCssClass' => 'active',
            'hiddenPageCssClass' => 'disabled',
            'htmlOptions' => array(
                'class' => 'pagination pagination-sm',
            )
        ));
    ?>

```

```
</div><!-- pagination (End) -->
```

```
</div><!-- content (End) -->
```

## **\_post.php:**

```
<!-- Перевірка наявності коментаря -->
```

```
<?php $comment = PostController::commentSearch($model->id); ?>
```

```
<div class="post">
```

```
<div id="post_<?php echo $model->id; ?>" class="well well-sm">
```

```
<!-- Заголовок питання автор/дата створення-->
```

```
<div class="header_post">
```

```
<span class="author"><i class="fa fa-user fa-lg"></i><?php echo $model->author; ?></span>
```

```
<div class="pull-right">
```

```
<span class="label label-default"><?php echo $model->created; ?></span>
```

```
</div>
```

```
</div>
```

```
<!-- Контент питання -->
```

```
<div class="post-content">
```

```
<p><?php echo $model->content; ?></p>
```

```
</div>
```

```
<!-- post-content (End) -->
```

```
<!-- Якщо користувач адмін йому доступні операції над постом -->
```

```
<?php if (Yii::app()->user->checkAccess('administrator')): ?>
```

```
<?php
```

```
$this->renderPartial('_post/adminwork', array(
```

```
    'model' => $model,
```

```
    'comment' => $comment,
```

```
));
```

```
?>
```

```
<?php endif; ?>
```

```
</div>
```

```
<!-- Якщо до цього питання є коментарій то виводимо його -->
```

```
<?php if (isset($comment->content)): ?>
```

```
<?php
```

```
$this->renderPartial('_post/comment', array(
```

```
    'model' => $model,
```

```
    'comment' => $comment,
```



```
));
?> <?php endif; ?></div>
```

### create.php:

```
<?php
/* @var $this PostController */
/* @var $model Post */
/* @var $form CActiveForm */
?>
<div id="content" class="row">
  <?php $form=$this->beginWidget('CActiveForm', array(
    'id'=>'post-form',
    'enableAjaxValidation'=>false,
    'htmlOptions'=>array(
      'class'=>'form-horizontal',
    ),
  )); ?>
  <div class="header_form">
    <div class="bg-warning rules">
      <p>Вимоги до змісту повідомлення:</p>
      <ol>
        <li>Максимально 1000 символів!</li>
        <li>Прямі та непрямі образи кого-небудь, зокрема, політиків, чиновників, адміністраторів ресурсу, користувачів ресурсу, у тому числі за національною, етнічною, расовою або релігійною приналежністю, а також шовіністичні висловлювання.</li>
        <li>Нецензурні вирази, висловлювання порнографічного, еротичного чи сексуального характеру.</li>
        <li>Будь-яка образлива поведінка по відношенню до авторів статей і всіх учасників ресурсу.</li>
        <li>Реклама, комерційні повідомлення, а також повідомлення, які не мають інформаційного навантаження і не стосуються тематики ресурсу.</li>
        <li>Будь-які повідомлення, які заборонені законодавством України.</li>
      </ol>
    </div><!-- create-rules (End) -->
  <div class="bg-danger">
    <?php echo $form->errorSummary($model); ?>
  </div>
```

```

</div>
<?php if (Yii::app()->user->checkAccess('administrator')): ?>
    <div class="col-xs-12 col-sm-5 col-md-5 col-lg-5">
        <div class="form-group">
            <div class="col-xs-12 col-sm-3 col-md-2 col-lg-2">
                <div class="row">
                    <?php echo $form->labelEx($model,'author',array('class'=>'control-label')); ?>
                </div>
            </div>
            <div class="col-xs-12 col-sm-9 col-md-9 col-lg-9 col-md-offset-1 col-lg-offset-1">
                <div class="row">
                    <?php echo $form->textField($model,'author',array('class'=>'form-control',
'value'=>'Administrator')); ?>
                </div>
            </div>
        </div>
    </div>
<?php endif; ?>

<?php if (Yii::app()->user->isGuest): ?>
    <div class="col-xs-12 col-sm-5 col-md-5 col-lg-5">
        <div class="form-group">
            <div class="col-xs-12 col-sm-2 col-md-2 col-lg-2">
                <div class="row">
                    <?php echo $form->labelEx($model,'author',array('class'=>'control-label')); ?>
                </div>
            </div>
            <div class="col-xs-12 col-sm-10 col-md-9 col-lg-9 col-md-offset-1 col-lg-offset-1">
                <div class="row">
                    <?php echo $form->textField($model,'author',array('class'=>'form-control')); ?>
                </div>
            </div>
        </div>
    </div>
<?php endif; ?>

```

```

<div class="col-xs-12 col-sm-6 col-md-6 col-lg-6 col-sm-offset-1 col-md-offset-1 col-lg-offset-1">
  <div class="form-group ">
    <div class="col-xs-6 col-sm-6 col-md-4 col-lg-4 ">
      <div class="row">
        <?php $this->widget('CCaptcha', array('clickableImage'=>true, 'showRefreshButton'=>true, 'buttonLabel'
=> '<i class="fa fa-refresh fa-2x icon-captcha" aria-hidden="true"></i>')); ?>
      </div>
    </div>
    <div class="col-xs-6 col-sm-6 col-md-8 col-lg-8">
      <div class="row">
        <?php echo CHtml::activeTextField($model, 'verifyCode',array('class' => "form-control cntr-
captcha",'placeholder'=>"Введіть текст з картинки")); ?>
      </div>
    </div>
  </div>
  <div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">
    <div class="form-group">
      <?php echo $form->labelEx($model,'content',array('class'=>'control-label')); ?>
      <?php echo $form->textArea($model,'content',array('class'=>'form-
control','rows'=>"10",'maxlength'=>'1000')); ?>
    </div>
  </div>
  <div class="col-xs-12 col-sm-12 col-md-6 col-lg-6 pull-right">
    <div class="form-group">
      <?php
      echo CHtml::submitButton($model->isNewRecord ? 'Додати запитання' : 'Зберегти',
      array('class' => 'btn btn-success col-xs-12 col-sm-12 col-md-6 col-lg-6 pull-right'));
      ?>
    </div>
  </div>
</div>
<?php $this->endWidget(); ?>

```

</div>

**update.php:**

```

<!-- CONTENT -->
<div id="content" class="row">
    <?php $form=$this->beginWidget('CActiveForm', array(
        'id'=>'post-form',
        'enableAjaxValidation'=>false,
        'htmlOptions'=>array(
            'class'=>'form-horizontal',
        ),
    )); ?>
    <div class="header_form">
        <div class="bg-warning rules">
            <p>Вимоги до змісту повідомлення:</p>
            <ol>
                <li>Прямі та непрямі образи кого-небудь, зокрема, політиків, чиновників, адміністраторів ресурсу, користувачів ресурсу, у тому числі за національною, етнічною, расовою або релігійною приналежністю, а також шовіністичні висловлювання.</li>
                <li>Нецензурні вирази, висловлювання порнографічного, еротичного чи сексуального характеру.</li>
                <li>Будь-яка образлива поведінка по відношенню до авторів статей і всіх учасників ресурсу.</li>
                <li>Реклама, комерційні повідомлення, а також повідомлення, які не мають інформаційного навантаження і не стосуються тематики ресурсу.</li>
                <li>Будь-які повідомлення, які заборонені законодавством України.</li>
            </ol>
        </div><!-- create-rules (End) -->
        <div class="bg-danger">
            <?php echo $form->errorSummary($model); ?>
        </div>
    </div>
    <?php if (Yii::app()->user->checkAccess('administrator')): ?>
        <div class="col-xs-12 col-sm-5 col-md-5 col-lg-5">
            <div class="form-group">
                <div class="col-xs-12 col-sm-3 col-md-2 col-lg-2">
                    <div class="row">
                        <?php echo $form->labelEx($model,'author',array('class'=>'control-label')); ?>
                    </div>
                </div>
            </div>
        </div>
    </if>

```

```

</div>

<div class="col-xs-12 col-sm-9 col-md-9 col-lg-9 col-md-offset-1 col-lg-offset-1">
    <div class="row">
        <?php echo $form->textField($model,'author',array('class'=>'form-control',
'value'=>'Administrator')); ?>
    </div>
</div>
</div>
</div>
</div>
<?php endif; ?>
<div class="col-xs-12 col-sm-6 col-md-6 col-lg-6 col-sm-offset-1 col-md-offset-1 col-lg-offset-1">
    <div class="form-group ">
        <div class="col-xs-6 col-sm-6 col-md-4 col-lg-4 ">
            <div class="row">
                <?php $this->widget('CCaptcha', array('clickableImage'=>true, 'showRefreshButton'=>true, 'buttonLabel'
=> '<i class="fa fa-refresh fa-2x icon-captcha" aria-hidden="true"></i>')); ?>
            </div>
        </div>
        <div class="col-xs-6 col-sm-6 col-md-8 col-lg-8">
            <div class="row">
                <?php echo CHtml::activeTextField($model, 'verifyCode',array('class' => "form-control cntn-
captcha",'placeholder'=>"Введіть текст з картинки")); ?>
            </div>
        </div>
    </div>
</div>
</div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">
    <div class="form-group">
        <?php echo $form->labelEx($model,'content',array('class'=>'control-label')); ?>
        <?php echo $form->textArea($model,'content',array('class'=>'form-
control','rows'=>"10",'maxlength'=>'1000')); ?>
    </div>
</div>
<div class="col-xs-12 col-sm-12 col-md-6 col-lg-6 pull-right">
    <div class="form-group">
        <?php
            echo CHtml::submitButton($model->isNewRecord ? 'Додати запитання' : 'Зберегти',

```

```

        array('class' => 'btn btn-success col-xs-12 col-sm-12 col-md-6 col-lg-6 pull-right'));
    ?>
</div>
</div>
<?php $this->endWidget(); ?>
</div><!-- content (End) -->

```

## adminwork.php:

```

<div class="post_footer">
    <div class="post_control btn-group">
        <a href="index.php?r=post/update&id=?php echo $model->id; ?>" class="btn btn-warning btn-xs"><i
class="fa fa-pencil"></i> Редагувати</a>

        <a href="index.php?r=post/delete&id=?php echo $model->id; ?>" class="btn btn-danger btn-xs"><i
class="fa fa-times"></i> Видалити</a>

        <?php if (!isset($comment->content)): ?>
            <a href="index.php?r=comment/create&id=?php echo $model->id; ?>" class="btn btn-default btn-
xs"><i class="fa fa-reply"></i>Відповісти</a>

        <?php endif; ?>
    </div>
</div>

```

## comment.php:

```

<div id="comment_<?php echo $comment->id; ?>" class="comment well well-sm">
    <div class="header_post">
        <span class="author"><i class="fa fa-user"></i>Administrator</span>
        <div class="pull-right">
            <?php if (Yii::app()->user->checkAccess('administrator')): ?>
                <a href="#comment_<?php echo $comment->id; ?>" class="label label-default" title="Посилання на цей
коментар">#</a>
            <?php endif; ?>
            <span class="label label-default"><?php echo $comment->created; ?></span>
        </div>
    </div>
<!-- Контент коментаря -->
<div class="content_post">
    <p><?php echo $comment->content; ?></p>
</div>
<?php if (Yii::app()->user->checkAccess('administrator')): ?>

```

```

<!-- Панель керування якщо адмін -->
<div class="post_footer">
    <div class="post_control btn-group">
        <a href="index.php?r=faq/redirn&post_id=?php echo $model->id; ?>" class="btn btn-info btn-xs"><i
class="fa fa-question-circle-o"></i> Додати до FAQ</a>
        <a href="index.php?r=faq/redir&post_id=?php echo $model->id; ?>" class="btn btn-success btn-xs"><i
class="fa fa-question-circle-o"></i> Відредагувати та додати до FAQ</a>
        <a href="index.php?r=comment/update&post_id=?php echo $model->id; ?>" class="btn btn-warning btn-
xs"><i class="fa fa-pencil"></i> Редагувати</a>
        <a href="index.php?r=comment/delete&post_id=?php echo $model->id; ?>" class="btn btn-danger btn-
xs"><i class="fa fa-times"></i> Видалити</a>
    </div>
</div><?php endif; ?>
</div><!-- comment (End) -->

```

### **create.php(для відповіді):**

```

<div id="content" class="row">
    <?php $form=$this->beginWidget('CActiveForm', array(
        'id'=>'post-form',
        'enableAjaxValidation'=>false,
        'htmlOptions'=>array(
            'class'=>'form-horizontal',
        ),
    )); ?>

    <div class="header_form">
        <div class="bg-warning rules">
            <p>Вимоги до змісту повідомлення:</p>
            <ol>
                <li>Прямі та непрямі образи кого-небудь, зокрема, політиків, чиновників, адміністраторів
ресурсу, користувачів ресурсу, у тому числі за національною, етнічною, расовою або релігійною
приналежністю, а також шовіністичні висловлювання.</li>
                <li>Нецензурні вирази, висловлювання порнографічного, еротичного чи сексуального
характеру.</li>
                <li>Будь-яка образлива поведінка по відношенню до авторів статей і всіх учасників
ресурсу.</li>
            </ol>
        </div>
    </div>

```



```
<li>Реклама, комерційні повідомлення, а також повідомлення, які не мають інформаційного навантаження і не стосуються тематики ресурсу.</li>
```

```
<li>Будь-які повідомлення, які заборонені законодавством України.</li>
```

```
</ol>
```

```
</div><!-- create-rules (End) -->
```

```
<div class="bg-danger">
```

```
<?php echo $form->errorSummary($comment); ?>
```

```
</div>
```

```
</div>
```

```
<div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">
```

```
<div class="form-group">
```

```
<?php echo $form->labelEx($comment,'content'); ?>
```

```
<?php echo $form->textArea($comment,'content',array('class'=>'form-control','rows'=>"10",'maxlength'=>'1000')); ?>
```

```
</div>
```

```
</div>
```

```
<div class="col-xs-12 col-sm-12 col-md-6 col-lg-6 pull-right">
```

```
<div class="form-group">
```

```
<?php
```

```
echo CHtml::submitButton($comment->isNewRecord ? 'Відповісти' : 'Зберегти',
array('class' => 'btn btn-success col-xs-12 col-sm-12 col-md-6 col-lg-6 pull-right'));
```

```
?>
```

```
</div>
```

```
</div>
```

```
<?php $this->endWidget(); ?>
```

```
</div><!-- content (End) -->
```

### **update.php(редагування відповіді):**

```
<div id="content" class="row">
```

```
<?php $form=$this->beginWidget('CActiveForm', array(
```

```
'id'=>'post-form',
```

```
'enableAjaxValidation'=>false,
```

```
'htmlOptions'=>array(
    'class'=>'form-horizontal',
),
)); ?>
```

```
<div class="header_form">
    <div class="bg-warning rules">
        <p>Вимоги до змісту повідомлення:</p>
        <ol>
            <li>Прямі та непрямі образи кого-небудь, зокрема, політиків, чиновників, адміністраторів ресурсу, користувачів ресурсу, у тому числі за національною, етнічною, расовою або релігійною приналежністю, а також шовіністичні висловлювання.</li>
            <li>Нецензурні вирази, висловлювання порнографічного, еротичного чи сексуального характеру.</li>
            <li>Будь-яка образлива поведінка по відношенню до авторів статей і всіх учасників ресурсу.</li>
            <li>Реклама, комерційні повідомлення, а також повідомлення, які не мають інформаційного навантаження і не стосуються тематики ресурсу.</li>
            <li>Будь-які повідомлення, які заборонені законодавством України.</li>
        </ol>
    </div><!-- create-rules (End) -->
    <div class="bg-danger">
        <?php echo $form->errorSummary($comment); ?>
    </div>
</div>

<div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">
    <div class="form-group">
        <?php echo $form->labelEx($comment,'content'); ?>
        <?php echo $form->textArea($comment,'content',array('class'=>'form-control','rows'=>"10",'maxlength'=>'1000')); ?>
    </div>
</div>

<div class="col-xs-12 col-sm-12 col-md-6 col-lg-6 pull-right">
    <div class="form-group">
        <?php
```

```
echo CHtml::submitButton($comment->isNewRecord ? 'Відповісти' : 'Зберегти',
array('class' => 'btn btn-success col-xs-12 col-sm-12 col-md-6 col-lg-6 pull-right'));
?>
</div>
</div>

<?php $this->endWidget(); ?>
</div><!-- content (End) -->
```

## Додаток Б

Створення завдання та прикладу лабораторної роботи №26 з курсу WEB-проекування.

Завдання: Створити сайт з інформаційним наповненням згідно варіанту завдання з використанням фреймворку Yii 1.1.

Варіанти завдань:

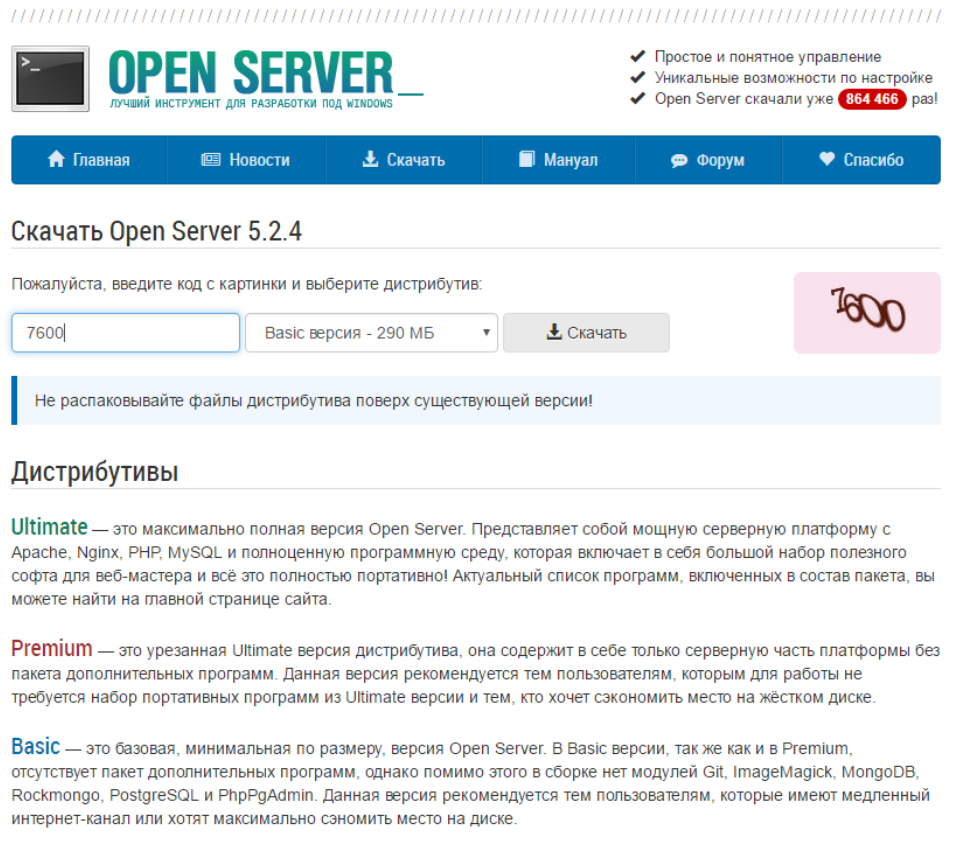
Розділи	Вар 1	Вар 2	Вар 3	Вар 4
Розділ 1	Класифікація Web-сайтів	Реєстрація Web-сайтів у мережі	Захист Web - сайтів	Класифікація Web-сайтів
Розділ 2	Процес розробки Web-сайтів	Рейтинг сайтів	Захист Web-серверів	Рейтинг сайтів
Розділ 3	Web-дизайн як етап розробки	Вразливості веб-серверів	Сканери пошуку вразливостей	Вразливості Web-сайтів
Розділ 4	Методи розробки Web-сайтів	Вразливості Web-сайтів	Мета-тегі та SEO	Web-дизайн як етап розробки
Розділи	Вар 5	Вар 6	Вар 7	Вар 8
Розділ 1	Процес розробки Web-сайтів	Реєстрація Web-сайтів у мережі	Захист Web - сайтів	Процес розробки Web-сайтів
Розділ 2	Вразливості веб-серверів	Сканери пошуку вразливостей	Реєстрація Web-сайтів у мережі	Рейтинг сайтів
Розділ 3	Сканери пошуку вразливостей	Вразливості веб-серверів	Сканери пошуку вразливостей	Вразливості Web-сайтів
Розділ 4	Рейтинг сайтів	Вразливості Web-сайтів	Мета-тегі та SEO	Процес розробки Web-сайтів

Рисунок Б.1 - Варіанти Лабораторної роботи №26

Інформація для виконання роботи.

## Встановлення та налаштування Yii 1.1. :

1. Завантажити та встановити останню Basic версію Open Server : <http://open-server.ru/download/>



**OPEN SERVER**  
ЛУЧШИЙ ИНСТРУМЕНТ ДЛЯ РАЗРАБОТКИ ПОД WINDOWS

- ✓ Простое и понятное управление
- ✓ Уникальные возможности по настройке
- ✓ Open Server скачали уже **864 466** раз!

Главная | Новости | Скачать | Мануал | Форум | Спасибо

### Скачать Open Server 5.2.4

Пожалуйста, введите код с картинки и выберите дистрибутив:

7600 | Basic версия - 290 МБ | Скачать

Не распаковывайте файлы дистрибутива поверх существующей версии!

#### Дистрибутивы

**Ultimate** — это максимально полная версия Open Server. Представляет собой мощную серверную платформу с Apache, Nginx, PHP, MySQL и полноценную программную среду, которая включает в себя большой набор полезного софта для веб-мастера и всё это полностью портативно! Актуальный список программ, включенных в состав пакета, вы можете найти на главной странице сайта.

**Premium** — это урезанная Ultimate версия дистрибутива, она содержит в себе только серверную часть платформы без пакета дополнительных программ. Данная версия рекомендуется тем пользователям, которым для работы не требуется набор портативных программ из Ultimate версии и тем, кто хочет сэкономить место на жестком диске.

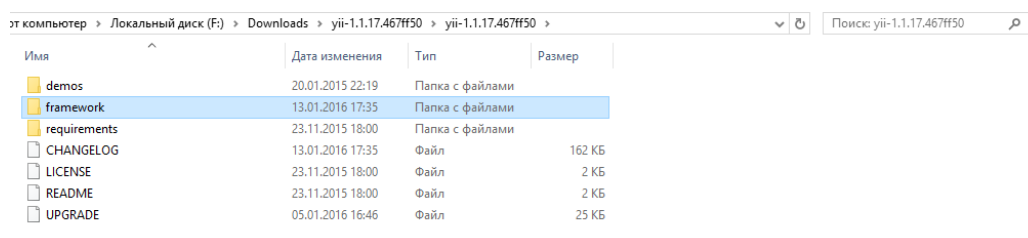
**Basic** — это базовая, минимальная по размеру, версия Open Server. В Basic версии, так же как и в Premium, отсутствует пакет дополнительных программ, однако помимо этого в сборке нет модулей Git, ImageMagick, MongoDB, Rockmongo, PostgreSQL и PhpPgAdmin. Данная версия рекомендуется тем пользователям, которые имеют медленный интернет-канал или хотят максимально сэкономить место на диске.

Рисунок Б.2 - Сторінка завантаження Open Server

2. Завантажити останню версію Yii 1.1 за адресою :

<https://github.com/yiisoft/yii/releases/download/1.1.17/yii-1.1.17.467ff50.zip>

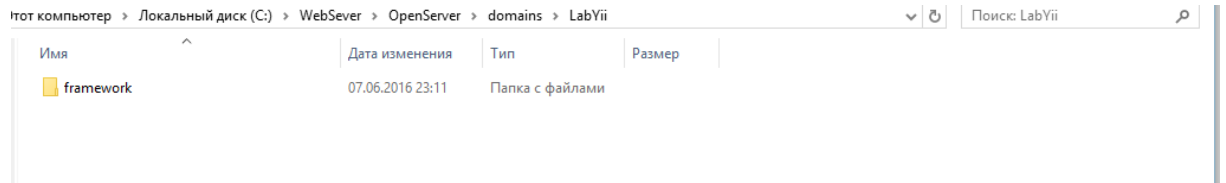
3. Розпакувати завантажений архів фреймворка і скопіювати папку framework до папки з вашим проектом на локальному сервері :



Имя	Дата изменения	Тип	Размер
demос	20.01.2015 22:19	Папка с файлами	
<b>framework</b>	13.01.2016 17:35	Папка с файлами	
requirements	23.11.2015 18:00	Папка с файлами	
CHANGELOG	13.01.2016 17:35	Файл	162 КБ
LICENSE	23.11.2015 18:00	Файл	2 КБ
README	23.11.2015 18:00	Файл	2 КБ
UPGRADE	05.01.2016 16:46	Файл	25 КБ

### Рисунок Б.3 - Розархівований фреймворк

Результатом буде папка з подібним вмістом(шлях до папки:  
C:\WebSever\OpenServer\domains\LabYii ) :



### Рисунок Б.4 - Папка проекту на сервері

#### 4. Встановлення Yii 1.1. :

Запускаємо Командний рядок і вводимо такі команди :

1. `cd C:\`
2. `WebSever\OpenServer\modules\php\PHP-5.4\php.exe`  
`C:\WebSever\OpenServer\domains\LabYii\framework\yiiic webapp`  
`C:\WebSever\OpenServer\domains\LabYii\`
3. Далі відобразиться, повідомлення про те чи бажаємо створити додаток.
4. Вводимо : `yes`

## 5. Отримуємо такий результат :

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.10240]
(c) Корпорація Майкрософт (Microsoft Corporation), 2015 г. Все права захищені.
C:\Users\MISHA>cd c:/
c:/>WebSever\OpenServer\modules\php\PHP-5.4\php.exe C:\WebSever\OpenServer\domains\LabYii\Framework\yiiic webapp C:\WebSever\OpenServer\domains\LabYii
Create a Web application under 'C:\WebSever\OpenServer\domains\LabYii'? (yes|no)
[no]:yes
mkdir C:\WebSever\OpenServer\domains\LabYii\assets
mkdir C:\WebSever\OpenServer\domains\LabYii\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\LabYii\images
generate index-test.php
generate index.php
mkdir C:\WebSever\OpenServer\domains\LabYii\protected
generate protected/.htaccess
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\commands
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\commands\shell
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\components
generate protected/components/Controller.php
generate protected/components/UserIdentity.php
mkdir C:\WebSever\OpenServer\domains\LabYii\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\LabYii\protected\controllers
generate protected/controllers/BarController.php
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\data
generate protected/data/schema.mysql.sql
generate protected/data/schema.sqlite.sql
generate protected/data/testdrive.db
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\extensions
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\messages
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\migrations
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\models
generate protected/models/ContactForm.php
generate protected/models/LoginForm.php
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\runtime
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\tests
generate protected\tests/bootstrap.php
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\tests\fixtures
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\tests\functional
generate protected\tests\functional\SiteTest.php
generate protected\tests\phpunit.xml
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\tests\report
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\tests\unit
generate protected\tests\WebTestCase.php
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\vendor
mkdir C:\WebSever\OpenServer\domains\LabYii\protected\views
mkdir C:\WebSever\OpenServer\domains\LabYii\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\LabYii\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\LabYii\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\LabYii\themes
mkdir C:\WebSever\OpenServer\domains\LabYii\themes\classic
mkdir C:\WebSever\OpenServer\domains\LabYii\themes\classic\views
generate themes\classic\views/.htaccess
mkdir C:\WebSever\OpenServer\domains\LabYii\themes\classic\views\layouts
mkdir C:\WebSever\OpenServer\domains\LabYii\themes\classic\views\site
mkdir C:\WebSever\OpenServer\domains\LabYii\themes\classic\views\system

Your application has been created successfully under C:\WebSever\OpenServer\domains\LabYii.
c:/>_

```

Рисунок Б.5 - Встановлення Yii 1.1.

6. Для перевірки того, чи дійсно створений додаток. Потрібно ввести в браузері назву папки в яку розпакований фреймворк : LabYii/

Отриманий результат:

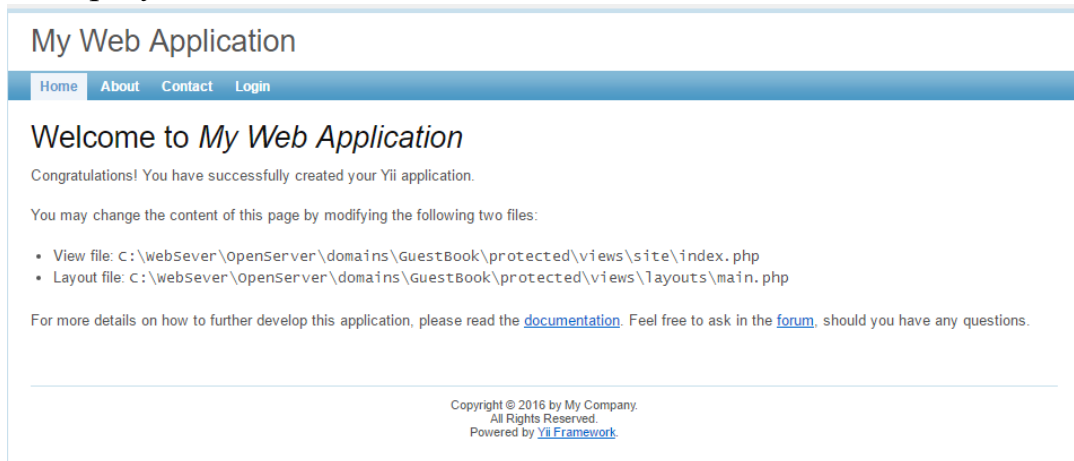


Рисунок Б.6 - Стандартно-згенерований додаток Yii 1.1.

## 5. Налаштування Yii 1.1. :

### Налаштування з'єднання з базою даних

Для створення свого проекту потрібно налаштувати Yii 1.1 для роботи з базою даних. Для цього потрібно відредагувати файл `database.php` (розміщений за адресою: `C:\WebSever\OpenServer\domains\LabYii\protected\config\database.php`).

Для цього потрібно додати наступний код:

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



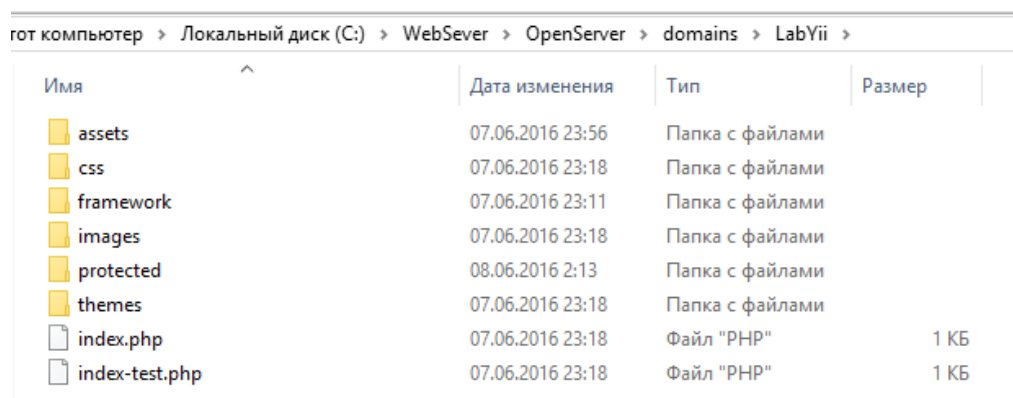
## Активация кодогенератора Gii

Фреймворк доступний разом з 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 main.php( адреса до файлу: C:\WebSever\OpenServer\domains\LabYii\protected\config\main.php) додаємо опис підключення модуля gii:

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

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

## 7. Структура папок в Yii 1.1.



Имя	Дата изменения	Тип	Размер
assets	07.06.2016 23:56	Папка с файлами	
css	07.06.2016 23:18	Папка с файлами	
framework	07.06.2016 23:11	Папка с файлами	
images	07.06.2016 23:18	Папка с файлами	
protected	08.06.2016 2:13	Папка с файлами	
themes	07.06.2016 23:18	Папка с файлами	
index.php	07.06.2016 23:18	Файл "PHP"	1 КБ
index-test.php	07.06.2016 23:18	Файл "PHP"	1 КБ

Рисунок Б.7 - Структура корневої папки Yii 1.1.

Assets - папка для тимчасових файлів, необхідних самому фреймворку,

CSS - папка для стилів,

Framework - безпосередньо, сам фреймворк,

Images - папка для зображень,

Protected - робоча дїрректорїю,

Themes - теми оформлення.

Основна папка, з якою нам доведеться працювати, - це папка Protected.

Розглянемо структуру цієї папки:

Имя	Дата изменения	Тип	Размер
commands	07.06.2016 23:18	Папка с файлами	
components	07.06.2016 23:18	Папка с файлами	
config	07.06.2016 23:18	Папка с файлами	
controllers	08.06.2016 2:13	Папка с файлами	
data	07.06.2016 23:18	Папка с файлами	
extensions	07.06.2016 23:18	Папка с файлами	
messages	07.06.2016 23:18	Папка с файлами	
migrations	07.06.2016 23:18	Папка с файлами	
models	07.06.2016 23:52	Папка с файлами	
runtime	08.06.2016 0:01	Папка с файлами	
tests	07.06.2016 23:18	Папка с файлами	
vendor	07.06.2016 23:18	Папка с файлами	
views	08.06.2016 2:07	Папка с файлами	
.htaccess	07.06.2016 23:18	Файл "HTACCESS"	1 КБ
yiiic	07.06.2016 23:18	Файл	1 КБ
yiiic.bat	07.06.2016 23:18	Пакетный файл ...	1 КБ
yiiic.php	07.06.2016 23:18	Файл "PHP"	1 КБ

Рисунок Б.8 - Структура папки protected.

Commands - папка для консольних додатків, для управління Фрейм-ворк за допомогою консолї.

Components - містить компоненти.

Config - містить конфїгураційні файли.

Controllers - папка для зберїгання контролерів.

Data - папка для зберїгання даних бази даних SQLite

Extensions - папка для розширень фреймворка.

Messages - папка для зберїгання системних повідомлень. У тому числі необхідна для зберїгання асоціацій перекладів в багатомовних приложени-ях.

Migrations - папка для відстеження змін в базї даних.

Models - папка моделей.

Runtime - папка для зберїгання тимчасових файлів.

Tests - використовується для тестування додатків.

Vendor - папка для підключення модулів сторонніх фрейвворков.

Views - папка яка містить views проекту.

### Приклад виконання роботи.

#### 1. Створення Бази даних за допомогою PhpMyAdmin.

Вхід в PhpMyAdmin:

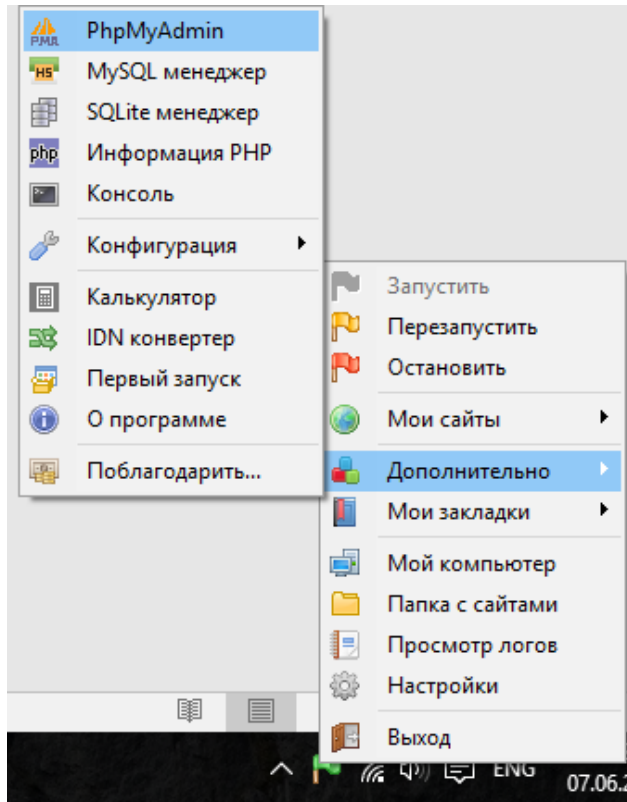


Рисунок Б.9 - Вхід в PhpMyAdmin.

Далі бачимо:

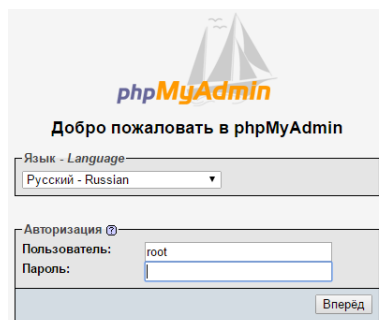


Рисунок Б.10 - Вхід в PhpMyAdmin(продовження).

Натискаємо “Вперед” бачимо:

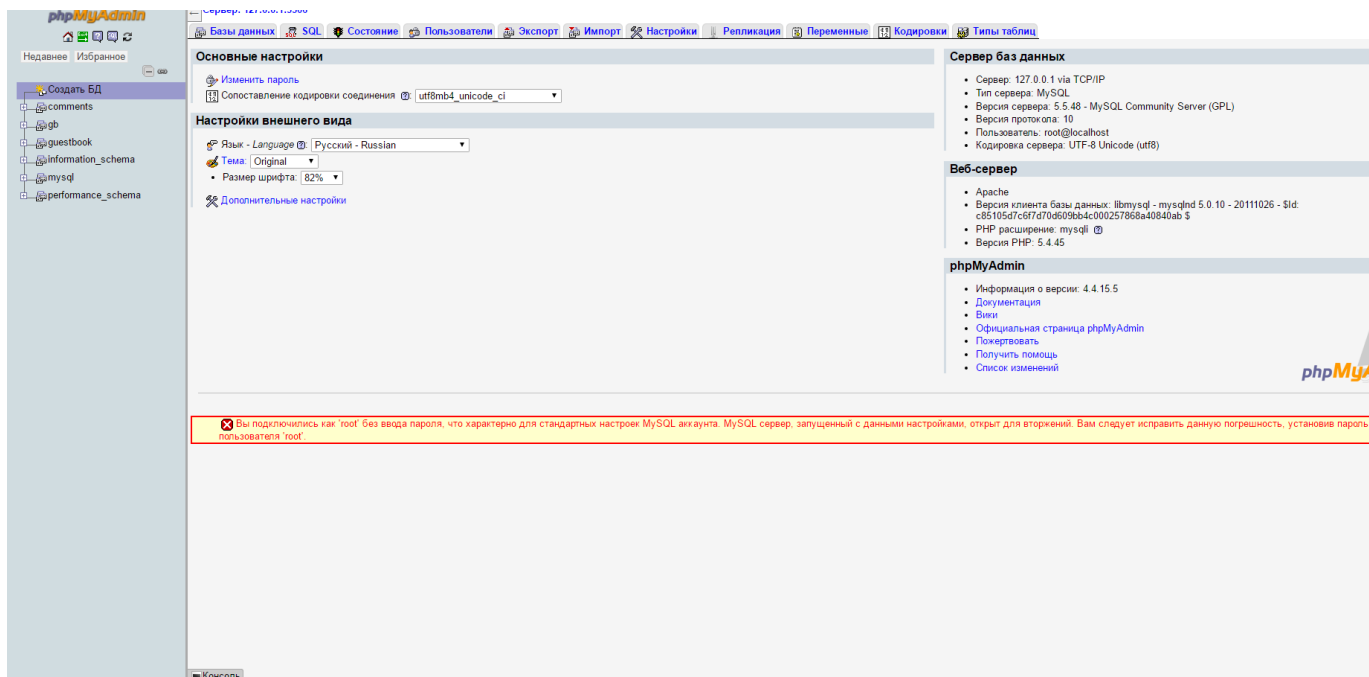


Рисунок Б.11 - Головна сторінка PhpMyAdmin.

Вибираємо пункт “Создать БД”:

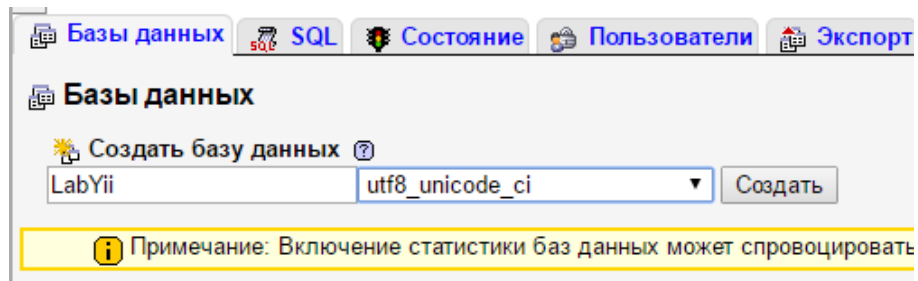


Рисунок Б.12 - Створення БД.

Тиснемо кнопку “Создать” і БД створена.

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

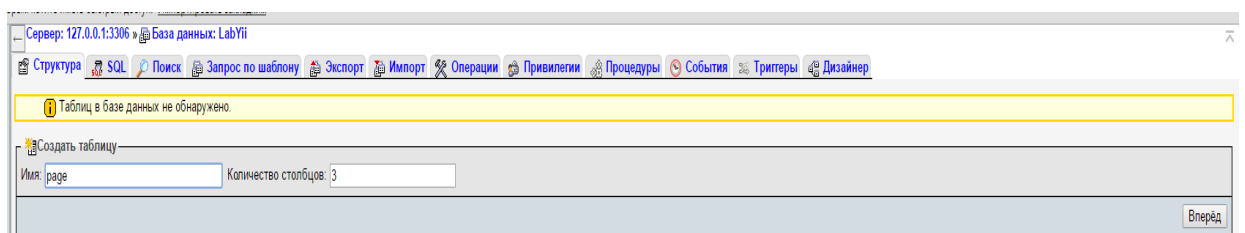


Рисунок Б.13 - Створення таблиць.

Натискаємо “Вперёд” бачимо:

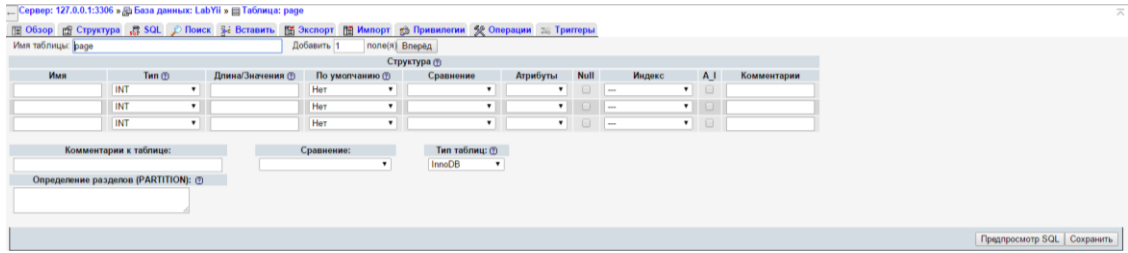


Рисунок Б.14 - Форма створення стовпців таблиці.

Далі вводимо назви стовпців:

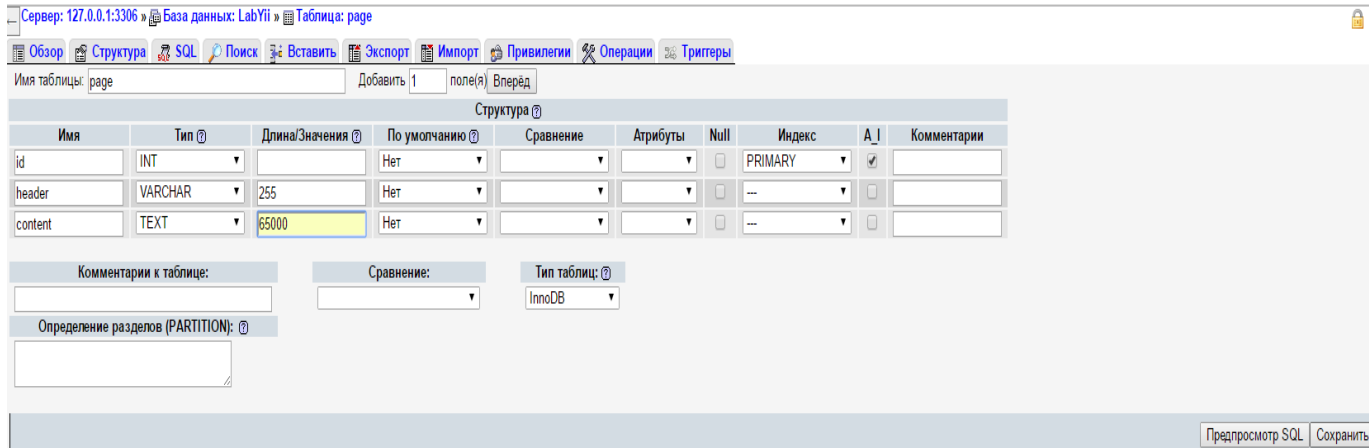


Рисунок Б.15 - Заповнення форми створення стовпців таблиці.

І тиснемо “Сохранить”.

Далі заповняємо таблицю даними згідно з варіантом:

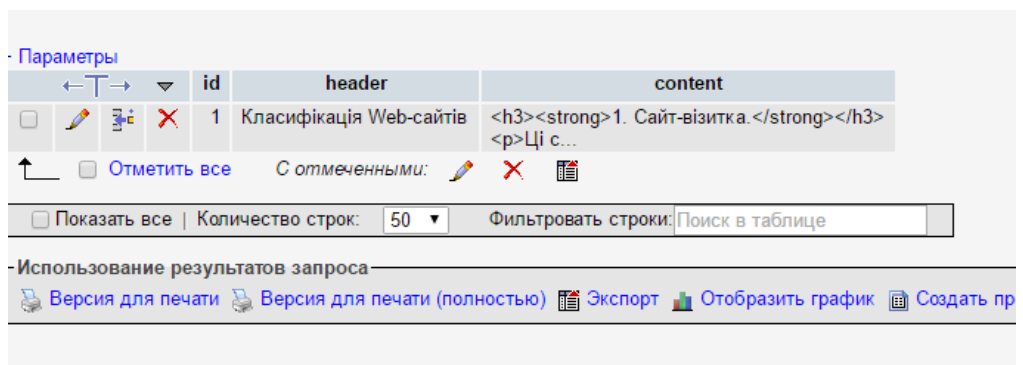


Рисунок Б.16 - Заповнення таблиці інформацією.

## 2. Створення моделі.

Для створення моделей необхідних для сайту скористаємося кодогенератором Gii. Щоб зайти в Gii необхідно перейти за наступним посиланням в браузері: <http://guestbook/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 page. Після цього ми маємо такий результат:

yii code generator [help](#) | [webapp](#) | [yii](#) | [logout](#)

**Generators**

- [Controller Generator](#)
- [Crud Generator](#)
- [Form Generator](#)
- Model Generator**
- [Module Generator](#)

### Model Generator

This generator generates a model class for the specified database table.

Fields with \* are required. Click on the highlighted fields to edit them.

Database Connection \*  
db

Table Prefix  
[empty]

Table Name \*  
page

Model Class \*  
Page

Base Class \*  
 CActiveRecord

Model Path \*  
application.models

Build Relations

Use Column Comments as Attribute Labels

Code Template \*  
default  
(C:\WebServer\OpenServer\domains\LabYii\framework\gii\generators\model\templates\default)

Code File	Generate
<a href="#">models/Page.php</a>	new <input checked="" type="checkbox"/>

Powered by [Yii Framework](#)  
A product of [Yii Software LLC](#)

Рисунок Б.17 - Створення моделі Page.

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

The code has been generated successfully.

Рисунок Б.18 - Успішно створена модель.

### 3. Створення контролера і відображень.

Для генерації контролерів та представлення `index.php` для, цього контролера використаєм Gii.

Заходимо в Gii вибираємо Controller Generator і вводимо назву контролера Page :

The screenshot shows the 'Controller Generator' interface in the Yii Code Generator. On the left, there is a sidebar with 'Generators' and a list of options: Controller Generator, Crud Generator, Form Generator, Model Generator, and Module Generator. The main area is titled 'Controller Generator' and contains the following fields and options:

- Controller ID \***: Page
- Base Class \***: Controller
- Action IDs**: index
- Code Template \***: default (C:\WebSever\OpenServer\domains\LabYii\framework\gii\generators\controller\templates\default)

Below the fields are 'Preview' and 'Generate' buttons. A table shows the generated files:

Code File	Generate
<a href="#">controllers\PageController.php</a>	new <input checked="" type="checkbox"/>
<a href="#">views\page\index.php</a>	new <input checked="" type="checkbox"/>

At the bottom, it says 'Powered by Yi Framework. A product of Yi Software LLC.'

Рисунок Б.19 - Створення контролера і відображень.

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 Б.20 - Успішно створений контроллер.

Перейшовши по ссилці “try it now”, можна побачити згенероване представлення:



Рисунок Б.21 - Предсталення для моделі Page.

Поки що контролер не зв'язаний з таблицею page. Тому сторінка пуста. Щоб відобразити дані з таблиці потрібно зробити такі кроки:

1. Редагуємо контролер PageController.php(шлях до файлу:  
C:\WebSever\OpenServer\domains\LabYii\protected\controllers\PageController.php):

Заміняємо стандартно згенерований код на:

```
<?php

class PageController extends Controller
{
    public function actionIndex()
    {
        $model= Page::model()->findAll($criteria);
        $this->render('index', array(
            'model' => $model,
        ));
    }
}
```

2. Редагуємо представлення index.php(шлях до файлу:  
C:\WebSever\OpenServer\domains\LabYii\protected\views\page\index.php)

Заміняємо стандартно згенерований код на:

```
<div id ="content">
    <!-- Відповідає за відображення -->
    <?php
    if (!empty($model))
        foreach ( $model as $key => $val)
        {
            $this->renderPartial('view', array(
                'model' => $val,
            ));
        }
    ?>
</div><!-- content (End) -->
```

3. Створюємо представлення view.php(шлях до файлу:  
C:\WebSever\OpenServer\domains\LabYii\protected\views\page\view.php):

В якому пишемо наступний код:

```
<div class="page">
    <div class="header">
        <h2><?php echo $model->header; ?></h2>
    </div>
```



```
<div class="content">
    <?php echo $model->content; ?>
</div></div>
```

4. Для перевірки успішності проведених робіт переходимо в браузері по адресі: <http://labyii/index.php?r=Page> і бачимо:

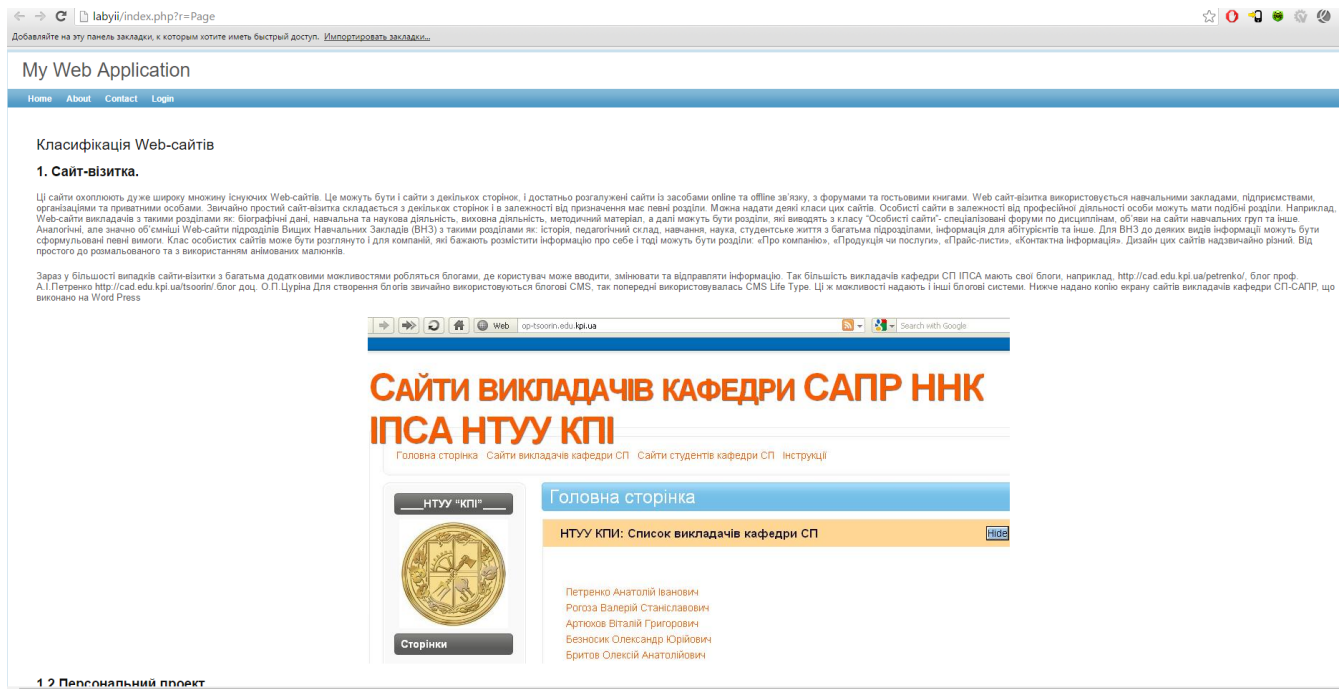


Рисунок Б.22 - Представлення сторінки Page після редагування.

Вимоги до звіту:

1. Варіант завдання.
2. Специфіка реалізації.
3. Особливості роботи з вибраними інструментами ( копії екрану).
4. Зовнішній вигляд ( копії екрану) і код основних сторінок MVC Вашого блогу.
5. Висновки.

Контрольні питання:

- 1 Особливості реалізація взаємодії модулів згідно з концепцією MVC.
- 2 Структура фреймворка Yii 1.1.
- 3 Особливості існуючих фреймворків для php.