

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА ПРОГРАМУВАННЯ ТА МАТЕМАТИКИ

**Пояснювальна записка  
до дипломної роботи**

**бакалавр**

(освітньо-кваліфікаційний рівень)

**на тему «Веб-додаток з опитування та дослідження стану співробітників»**

Виконав: студент 4 курсу, групи ІПЗ-17д  
напряму підготовки 121 „Інженерія  
програмного забезпечення”

\_\_\_\_\_ Макаров С.К.  
(підпис)

Керівник,  
доцент, к.т.н. \_\_\_\_\_ Іванов В.Г.  
(підпис)

Рецензент,  
доцент каф. ПМ,  
ст. викл. \_\_\_\_\_ Батурін О.І.  
(підпис)

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ  
бакалаврської роботи студента гр. ПЗ-17д Макарова С.К.

Науковий керівник

доцент, к.т.н. \_\_\_\_\_ Іванов .В.Г

ПБ, посада

Оцінка наукового керівника: \_\_\_\_\_

Рецензент Батурін О.І., ст. викл. каф. ПМ СНУ ім. В. Даля

ПБ, місто роботи, посада

Оцінка рецензента: \_\_\_\_\_

Кінцева оцінка за результатами захисту: \_\_\_\_\_

Голова ЕК,  
проф. кафедри ПМ  
д.т.н., доцент

\_\_\_\_\_

підпис

Захожай О.І.

**СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ**

Факультет інформаційних технологій та електроніки

Кафедра програмування та математики

Освітньо-кваліфікаційний рівень бакалавр

Напрямок підготовки 121 „Інженерія програмного забезпечення”

ЗАТВЕРДЖУЮ

Завідувач кафедри ПМ,

д.т.н., доцент

\_\_\_\_\_ Лифар В.О.

« \_\_\_ » \_\_\_\_\_ 2021 р.

**З А В Д А Н Н Я**

НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Макарова Сергія Костянтиновича

**1. Тема роботи Веб-додаток з опитування та дослідження стану співробітників.**

**керівник роботи к.т.н., доцент Іванов Віталій Геннадійович**

затверджені наказом вищого навчального закладу від «22» квітня 2021 року

№ 68/14.04

2. Строк подання студентом роботи 12 червня 2021 р.

3. Вихідні дані до роботи

Об'єктом досліджень є розробка веб додатку з опитування та аналізу стану працівників фірми.

3.1 Літературні джерела:

Переваги побудови динамічних сайтів - hackernoon.com [Електронний ресурс] - Режим доступу: <https://hackernoon.com/the-advantages-ofbuilding-dynamic-websites-bc914071a155>

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Вступ

4.2 Аналіз предметної галузі (огляд літератури), з висвітленням наступних питань:

Основні завдання веб додатку.

Аналіз конкурентів.

4.3 Основна частина, в якій висвітлити:

Вибір і опис методології проектування.

Технології розробки сайту.

Вибір засобів розробки.

Проектування схеми і опис бази даних.

Вибір СУБД.

Проектування макету сайту

Реалізація веб сайту.

4.4 Висновки

4.4 Перелік використаних джерел

5. Перелік графічного матеріалу немає

6. Дата видачі завдання 23 квітня 2021 року.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Одержання завдання на виконання роботи	23.04.21	
2	Укладання і погодження з керівником плану і етапів виконання роботи	23.04.21	
3	Узагальнення даних літературних джерел, укладання розділу «Аналіз предметної галузі»	26.04.21	
4	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху	02.05.21	
5	Розробка технічного завдання	09.05. 21	
6	Укладання програмного продукту	11.05.21	
7	Укладання, оформлення та погодження пояснювальної записки з керівником	25.05.21	
8	Здача готової пояснювальної записки на кафедрі	06.06.21	
9	Укладання доповіді і презентації	07.06.21	

Студент

\_\_\_\_\_ Макаров С.К.  
(підпис)

Керівник роботи

\_\_\_\_\_ Іванов В. Г.  
(підпис)

## РЕФЕРАТ

Текст – 65., рис. – 30, літературних джерел – 16

Продуктивність роботи фірми напряму залежить від продуктивності роботи працівників, що в свою чергу залежить від робочої атмосфери в колективі та від стану конкретного працівника. Цими питаннями займаються HR – human resources, у їх роботі потрібно досліджувати стан працівників різними методами, у тому числі анонімними та звичайними різного роду опитуваннями. Тому питання розробки веб додатка для опитування та дослідження стану працівників фірми є актуальним.

Метою роботи є розробка веб-додатку для проведення опитувань серед співробітників фірми та аналізом результатів

Ключові слова: веб додаток, опитування, PHP, jQuery, YII

## ЗМІСТ

ВСТУП .....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ.....	9
1.1. Основні поняття.....	9
1.2. Основні відмінності web- додатку від web-сайту .....	9
1.3. Переваги та недоліки Web-додатку .....	12
1.4. Архітектура та принципи роботи типового web-додатку. ....	14
1.5. ВИСНОВОК ДО РОЗДІЛУ 1.....	15
РОЗДІЛ 2. АНАЛІЗ ТА ПОРІВНЯННЯ ОБРАНИХ ІНСТРУМЕНТІВ І ТЕХНОЛОГІЙ .....	16
2.1. HTML.....	16
2.2. CSS .....	19
2.3. JavaScript.....	20
2.4. jQuery .....	22
2.5. PHP .....	23
2.6. Фреймворки у веб-розробці.....	24
2.7. Open Server .....	36
2.8. ВИСНОВОК ДО РОЗДІЛУ 2.....	39
РОЗДІЛ 3. РОЗРОБКА ВЕБ ДОДАТКА.....	41
3.1. Постановка задачі.....	41
3.2. Підготовка до розробки додатку.....	42
3.3. Розгортання та налаштування фреймворку Yii2 на проєкті .....	45
3.4. ВИСНОВОК ДО РОЗДІЛУ 3.....	63
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65

## ВСТУП

### Актуальність.

Продуктивність роботи фірми напряму залежить від продуктивності роботи працівників, що в свою чергу залежить від робочої атмосфери в колективі та від стану конкретного працівника. Цими питаннями займаються HR – human resources, у їх роботі потрібно досліджувати стан працівників різними методами, у тому числі анонімними та звичайними різного роду опитуваннями. Тому питання розробки веб додатка для опитування та дослідження стану працівників фірми є актуальним.

Об'єкт дослідження – створення веб додатку.

Предмет дослідження – розроблення веб додатку для опитування та дослідження стану працівників фірми.

Мета дослідження: розробити веб додаток для опитування та дослідження стану працівників фірми.

### Задачі дослідження:

- 1) Проаналізувати предметну галузь обраної області.
- 2) Визначити існуючі підходи та технології для створення веб застосувань.
- 3) Проаналізувати інструменти розробника для створення веб застосувань.
- 4) Порівняти інструменти за якісними та кількісними характеристиками.
- 5) Реалізувати веб застосування



## РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

### 1.1. Основні поняття

Веб-додаток - це будь-яка комп'ютерна програма, яка виконує певну функцію, використовуючи веб-браузер у якості свого клієнта, в незалежності від пристрою та платформи, на яких відкрито браузер. Dodatok може бути таким же простим, як дошка оголошень або контактна форма на веб-сайті, або настільки ж складний, як текстовий процесор або багатокористувацький мобільний ігровий додаток, який ви завантажуєте на свій телефон.

Веб-сайт - це група глобально доступних взаємопов'язаних веб-сторінок, які мають єдине доменне ім'я. Він може бути розроблений і підтримуватися особою, бізнесом або організацією. Веб-сайт має на меті служити різноманітним цілям. Приклад: сайти-візитки, блоги, новостні портали та ін. Основні функції веб-сайту:

1. Ефективний спосіб демонструвати свої продукти та послуги.
2. Розробка сайту допоможе створити соціальний доказ.
3. Допомагає у брендінгу бізнесу.
4. Допомагає досягти ділових цілей.
5. Дозволяє розширити підтримку клієнтів

### 1.2. Основні відмінності web- додатку від web-сайту

#### 1.2.1. Інтерактивність

Перша відмінність - це різні степені взаємодії із сторінкою. У той час, як веб-сайт містить тексти та візуальний контент, з яким користувач не може взаємодіяти, веб-додатки надають користувачеві можливість не лише читати, а й змінювати, додавати інформацію на сторінках.

Інтернет-магазин, який дозволяє користувачам купувати товари, шукати їх каталозі, можна назвати веб-додатком. Ще один цікавий приклад - це соціальні мережі. Вони включають у себе функції блогу, чати, контент на вибір користувача та можливість ділитися цим контентом.

Сьогодні велика кількість веб-сайтів мають інтерактивність. Тому що користувачеві це подобається. Для цього власники веб-сайтів додають на свої веб-сайти невеликі веб-програми.

На сайті деяких ресторанів є доступ до Google Maps, який допомагає користувачеві знайти дорогу до ресторану. У той ж час, більшість сайтів більш інформативні ніж інтерактивні. Таким чином відвідувачі сайтів більш захоплені переглядом, читанням або прослухуванням інформації. А відвідувачі веб-додатку направлені на взаємодію з користувачами.

### **1.2.2. Інтеграція**

Інтеграція - це процес, що містить об'єднання простих компонентів у одному складному. Розробники можуть інтегрувати веб-додатки та сайти з програмами, включаючи ERP, CRM. Однак у більшості випадків інтеграція відбувається саме з веб-додатками, тому що їх складним функціям дуже часто потрібна інформація з інших систем. Найпопулярніший вид інтеграції в електронній комерції - це інтеграція веб-додатків із системою управління взаємодією з клієнтами (CRM). Це допомагає зберігати дані про покупців, інформацію про замовлення та покращити продажі в цілому. Завдяки інтеграції, інформація про користувачів, автоматично збирається та зберігається в CRM системі. Така інтеграція може дозволити команді відділу продажів дізнатися більше про поведінку клієнтів і ефективніше працювати з негативними відгуками. Доступна можливість взаємодії з інформацією про клієнта може принести збільшення в продажі та вдосконалити процес роботи в Інтернет-магазині.

У деяких випадках власники веб-сайтів використовують інтеграцію з CRM для того, щоб представити відвідувачам більше персоналізованого контенту. Але, на відміну від веб-додатків, подібна інтеграція з веб-сайтами - це скоріше опціональна функція, а не частина основного функціоналу.

### **1.2.3. Авторизація**

Цей процес включає в себе введення користувальницьких даних для отримання доступу до веб-сайтів або систем. Ця функція важлива для систем, які вимагають особисту інформацію про клієнтів. На цьому етапі важливо приділити особливу увагу безпеці. Дуже важливо мінімізувати вірогідність доступу до особистих даних користувача стороннім людям.

На відміну від сайтів, веб-додатки частіше вимагають авторизації, тому що вони пропонують користувачам більше можливостей, ніж веб-сайти. Наприклад, при використанні соціальних мереж, системи попереджують вас про слабкість пароля. Ігнорування подібних повідомлень може привести до того, що хакери можуть отримати доступ до вашого аккаунту.

Більшість інформаційних сайтів використовують авторизацію. У деяких випадках авторизація використовується для того, щоб дати більше можливостей, які недоступні неавторизованим користувачем. Якщо незареєстровані

користувачі можуть лише переглядати статті, зареєстровані користувачі можуть залишити коментарі, ділитися статтями у соціальних мережах та ін. Це також відмінне рішення для блокування спаму. Таким чином, авторизація необхідна як для веб-сайтів, так і для веб-додатків. Але в цей ж час авторизація потрібна веб-додаткам в цілях безпеки інформації.

У сучасному світі інтернету майже не залишилось сайтів, які б не мали ніяких інтерактивних елементів. І навпаки, багато веб-додатків часто включають

у себе пошук інформації. Тим не менш, веб-сайти як і раніш являються інформаційними джерелами, а веб-додатки залишаються користувальницькими інструментами.

Можна сказати що всі Web-додатки є Web-сайтами, та не всі Web-сайти є

Web-додатками. Те що робить їх такими унікальними, так це той факт, що неможливо розробити Web-додаток з чистого листа самостійно. У будь-якому разі доведеться використовувати не тільки звичні мови розмітки та програмування, як HTML, CSS та JavaScript, але й цілі бібліотеки технологій від сторонніх розробників. Є велика кількість компаній, які сконцентровані на створенні подібних технологій для стартапів і малого бізнесу.

### **1.3. Переваги та недоліки Web-додатку**

#### **1.3.1. Переваги створення Web-додатків**

Веб-додатки розробляються з такими мовами програмування, як HTML та CSS, які добре відомі серед IT-фахівців.

На відміну від нативних додатків, веб-додаток може використовуватись на усіх пристроях. Він запрограмований для роботи в будь-якій операційній системі. Він повинен належним чином адаптуватися до iOS, Android, Windows Phone та інших системах.

Веб-додатки не вимогливі до ресурсів і не пред'являють ніяких вимог доапаратної платформи.

Немає проблем з підтримкою старих версій програм й з сумісністю. Коли з'являється нова версія нативного додатку, користувачам нерідко доводиться вирішувати проблеми, зв'язаних з оновленням вже встановленої на їх пристроях копії. У випадку з браузерним додатком таких проблем не буває – існує лише одна версія, в якій працюють усі користувачі і у випадку

виходу нової, всі без виключень переходять на неї, іноді навіть не помічаючи цього.

Їх не потрібно завантажувати та встановлювати з магазинів додатків, таких як Вони також можуть відкривати веб-сайти. Це означає, що вони не потребують оновлення, як це роблять звичайні програми.

Високий рівень розвитку надійності сітьових з'єднань та web-технологій. Розробка веб-додатків - дешевший вид розробки додатків. Він складається із створення посилання або декількох посилань між додатком та URL-адресою. Розробка нативного або інтерпретованого додатка тягне за собою більш високу вартість, але шанси на успіх додатку набагато більше.

Web-додатки дозволяють своїм користувачам бути насправді мобільними. Ви можете зберігати результат своєї роботи на сервері, і у випадку необхідності мати доступ до них із будь якої точки планети, де є вихід у мережу Інтернет.

### **1.3.2. Недоліки створення Web-додатків**

Як було сказано вище, один веб-додаток може використовуватись на всіх пристроях. Але, звичайно ж, веб-сайт повинен бути запрограмований таким чином, щоб він відображався незалежно від операційної системи пристрою.

Якщо це не адаптивний веб-сайт, у вас можуть виникнути проблеми під час його відображення на iOS, Android або Windows Phone.

Підключення до Інтернету буде абсолютно необхідним для його запуску.

В іншому випадку ви не зможете переглядати веб-сайт, і веб-додаток вам не принесе користі. На превеликий жаль, наприклад, у нашій країні, інтернет є не всюди. Що ускладнює роботу з web-додатками, наприклад, під час подорожі.

Кожне перезавантаження (або оновлення сторінки) викликає помітну затримку, викликану необхідністю встановити HTTP- з'єднання, обробити запит на сервері, передати по мережі у відповідь HTTP-повідомлення і перезавантажити сторінку браузером. Це створює переривчастий режим роботи та уповільнює його.

Крім того, існують деякі обмеження доступу щодо певних апаратних функцій пристрою, на якому він працює.

Існує багато велика кількість додатків, які не можуть бути реалізовані в Web, наприклад в браузері неможливо створювати складні трьохмірні моделі.

#### **1.4. Архітектура та принципи роботи типового web-додатку.**

Web-додаток являється клієнт-серверним, в якому клієнтом виступає

браузер, а сервером web-сервер. Логіка web-додатку розподілена між сервером та клієнтом. Збереження даних переважно здійснюється на сервері. Обмін інформації проходить у мережі Інтернет.

В web-додатках існує дві програми, працюючі одночасно: код, який знаходиться на сервері і відповідає на HTTP запити; код, який знаходиться в браузері і реагує на введення даних користувачем.

Серверний код не видимий для клієнта, може відповідати тільки на HTTP запити конкретного URL, а не на будь-який тип користувацького введення. Зазвичай цей код написаний на наступних мовах програмування та фреймворках: JavaScript(Node.js), Python(Django), PHP, Ruby On Rails, Java, C# та інші.

Клієнтський код розбирається браузером користувача. На відміну від серверного він може реагувати на користувацькі введення, доступний для перегляду та редагування користувачеві у повному обсязі. Не може читати

файли сервера напряму, повинен звертатися з сервером через HTTP запити.  
Мови програмування: HTML, CSS, JavaScript.

### **1.5. ВИСНОВОК ДО РОЗДІЛУ 1**

Можна зробити висновки щодо особливостей розробки Web-додатку. Саме завдяки подібним проектам можна легко позбавитись будь-яких проблем, пов'язаних із представленням інформації сучасному користувачу. Розробка Web-додатку займає не так багато часу і ресурсів, щоб віддавати перевагу звичайним сайтам. Така розробка може бути відкрита на будь-якому пристрої, тому база потенційних користувачів автоматично збільшується. Можна забути про проблеми з оновленням програмного забезпечення, релізом нових версій сервісу. Тому що Web-додаток можна легко покращувати, не докладаючи до цього багато зусиль.

Гнучкість технологій розробки дозволяє створити Web-додаток для будь-якої сфери діяльності.

## **РОЗДІЛ 2. АНАЛІЗ ТА ПОРІВНЯННЯ ОБРАНИХ ІНСТРУМЕНТІВ І ТЕХНОЛОГІЙ**

### **2.1. HTML**

HTML або "Мова розмітки гіпертексту" – це код, який використовується для створення веб-сторінок та веб-додатків, доступ до яких здійснюється в Інтернеті. HTML був розроблений на зорі всесвітньої павутини наприкінці 1980-х - початку 1990-х. Мова дозволяє розробникам веб-сайтів розповісти інтернет-браузеру, як відображати елементи, такі як зображення, текст, форми та інтерактивні функції.

Творці веб-сайтів регулярно використовують HTML разом із каскадними таблицями стилів (CSS) та JavaScript, що є двома іншими мовами програмування, для створення сайтів та програм, до яких користувачі можуть отримати доступ, використовуючи Інтернет-браузерні програми, такі як Chrome, Firefox, Safari та Edge. Стандарти для HTML, а також CSS історично підтримували консорціум всесвітньої веб-сторінки (W3C).

Саме в 1980 році Тім Бернерс-Лі, працюючи в Європейській організації ядерних досліджень, почав працювати над прототипом HTML. До кінця десятиліття Бернерс-Лі створив всюдищу мову, а також Інтернет-браузер та серверне програмне забезпечення.

Протягом наступних двох десятиліть Всесвітній консорціум із веб-сторінок (W3C) продовжить розробку за міжнародним стандартом коду. У 1995 році була опублікована друга версія HTML, і протягом наступних кількох років наступні декілька додаткових оновлень.



В той час як HTML побудовано на основі правил SGML (стандартна узагальнена мова розмітки, це деяка метамова на якій можна визначати мову розмітки для документів), XHTML побудовано на основі правил XML, суворішої підмножини правил SGML. Оскільки XHTML документи мають бути коректними XML документами, їх обробку можна здійснювати стандартними інструментами обробки XML документів на відміну від HTML, який вимагає порівняно складніших, важчих і повільніших синтаксичних аналізаторів. XHTML можна розглядати як, багато в чому, перетин HTML і XML, оскільки цей стандарт є переформулюванням HTML засобами XML. XHTML 1.0 став рекомендацією консорціуму W3C 26 січня 2000 року. XHTML 1.1 став рекомендацією W3C 31 травня 2001 року.

Мова HTML інтерпретується браузером і відображається у вигляді документа, зручному для людини. HTML є додатком SGML (стандартної узагальненої мови розмітки) і відповідає міжнародному стандарту ISO 8879. HTML-документ є текстовим файлом розмічений за допомогою спеціальних (природно, текстових) команд. Текстовий формат представлення веб-документів був вибраний виходячи з основних вимог до веб-документу: простота, можливість безпосередньої інтерпретації в будь-якій операційній системі, мінімальний розмір файлу, зручність редагування і інтерпретації.

Мова розмітки гіпертекстових документів HTML дозволяє визначити різні типи елементів, що забезпечують функціональність документа: текстові фрагменти із заданими параметрами форматування, списки, таблиці, зображення, гіперпосилання і т.д. Елементи HTML оголошуються за допомогою команд розмітки, званих тегами (від англійського tag – ярлик). Усі HTML-теги, що зустрічаються в тексті документа інтерпретуються браузером при відображенні документа.

HTML - мова розмітки, це фраза, яка використовується для опису того, як програміст може використовувати код для позначення тексту. Мови

розмітки, такі як HTML та XML, читаються людьми та відрізняються від машинних мов, які записуються шістнадцятковим чи двійковим кодом.

Використовуючи HTML для створення сторінки на веб-сайті, автор використовуватиме декілька ключових компонентів або атрибутів, які зможе прочитати будь-який Інтернет-браузер, що отримує доступ до веб-сайту. Багато з цих атрибутів будуть в поєднанні зі «початковим тегом» та «кінцевим тегом».

HTML 5 є найбільш актуальним на сьогоднішній день стандартом цієї мови розмітки. Саме на цьому стандарті базується найбільша кількість Web-сторінок у інтернеті. В останній версії для HTML, що представляє собою HTML 5, ми отримуємо підтримку мультимедіа (аудіо та відео). Сьогодні основним складовим елементом веб-сторінок є HTML 5. Він надав статичний вигляд веб-сторінок. А всі інші стилі та функції додаються за допомогою CSS, PHP, Javascript. Тому перш ніж вивчати ці мови, ви повинні вивчити HTML.

Можна сказати, що HTML – це основний будівельний блок Web-розробки. Всі сучасні Web-сайти, сервіси створення Web-сайтів, такі як WordPress, Wix, Weebly використовують HTML 5 для створення Web-сайту.

ТМЛ-елемент виділяється з іншого тексту в документі за допомогою "тегів", які складаються з імені елемента оточеного "<" і ">". Ім'я елемента всередині тегу не чутливе до регістру. Тобто, воно може бути написано в верхньому або нижньому регістрі, або змішано. Наприклад, тег <title> може бути записаний як <Title>, <TITLE>, або будь-яким іншим способом.

Програміст веб-сайту може використовувати атрибути та елементи HTML для визначення практично кожної грані сторінки та того, як вона буде відображатися користувачеві. Програміст може використовувати CSS та

JavaScript, а також додаткові мови, такі як PHP, jQuery та XML для створення веб-сайту.

Новачки-програмісти часто вивчать HTML як свою першу мову програмування та продовжать навчання з передовими уроками JavaScript та CSS.

Вивчення основ HTML може виявитись корисним для майбутніх програмістів, а також тих, хто бажає зрозуміти, що потрібно для створення веб-сторінок, які він чи вона читає щодня на комп'ютері чи смартфоні.

## 2.2. CSS

CSS - це мова каскадних таблиць стилів (англ. cascading style sheets), яка використовується для опису стилів багаторазового використання для подання документів, написаних мовою розмітки. Його концепція була створена компанією Hakon Wium Lie у 1994 році. У грудні 1996 року W3C розробила специфікацію для CSS і сьогодні дозволяє веб-розробникам змінювати макет і зовнішній вигляд своїх веб-сторінок. Наприклад, CSS може використовуватися для зміни шрифту, який використовується в певному HTML-елементі, а також його розміру та кольору. Один CSS-файл може бути пов'язаний з декількома сторінками, що дозволяє розробнику одночасно змінювати зовнішній вигляд усіх сторінок. Наведене нижче поле містить основний приклад використання коду CSS для визначення шрифтів, кольору гіперпосилань та кольору осилання, коли курсор миші наводить курсор миші. У цьому конкретному прикладі ми змінюємо лише теги HTML `<a>` та `<body>`, а не створюємо нові селектори класу чи id.

Переваги CSS:

- CSS економить час - ви можете написати CSS один раз, а потім повторно використовувати той же аркуш на декількох HTML-

сторінках. Ви можете визначити стиль для кожного елемента HTML і застосувати його до якомога більше веб-сторінок.

- Сторінки завантажуються швидше - якщо ви використовуєте CSS, вам не потрібно щоразу писати атрибути тегів HTML. Просто напишіть одне правило CSS тегу та застосуйте його до всіх випадків цього тегу. Так менше коду означає швидше завантаження.
- Простота обслуговування - Щоб зробити глобальну зміну, просто змініть стиль, і всі елементи на всіх веб-сторінках будуть оновлені автоматично.
- Покращені стилі до HTML - CSS має набагато ширший набір атрибутів, ніж HTML, тому ви можете надати набагато кращий погляд на свою сторінку HTML порівняно з атрибутами HTML.
- Сумісність декількох пристроїв - аркуші стилів дозволяють оптимізувати вміст для більш ніж одного типу пристроїв. Використовуючи один і той же документ HTML, різні версії веб-сайту можуть бути представлені для портативних пристроїв, таких як КПК та мобільних телефонів або для друку.

### 2.3. JavaScript

JavaScript – скриптова мова, що найчастіше використовується при створенні сценаріїв поведінки браузера, що вбудовуються у веб-сторінки.

Назва «JavaScript» є зареєстрованою торговою маркою компанії Sun Microsystems, Inc.

JavaScript був створений, щоб "оживити веб-сторінки". Програми цією мовою називаються скриптами. Їх можна записати прямо в HTML веб-сторінки та запустити автоматично під час завантаження сторінки. Сценарії надаються та виконуються як звичайний текст. Для запуску їм не потрібна

спеціальна підготовка чи компіляція. У цьому аспекті JavaScript сильно відрізняється від іншої мови під назвою Java.

Коли був створений JavaScript, він спочатку мав іншу назву: "LiveScript". Але Java була дуже популярною в той час, тому було вирішено, що позиціонування нової мови як "молодшого брата" Java допоможе.

Але в міру розвитку, JavaScript став цілком незалежною мовою зі своєю специфікацією під назвою ECMAScript, і тепер він взагалі не має відношення до Java.

JavaScript найчастіше використовується в якості мови сценаріїв на стороні клієнта. Це означає, що код JavaScript записується на сторінку HTML. Коли користувач вимагає HTML-сторінки з JavaScript у ній, скрипт надсилається до браузера, і браузер повинен вирішувати щось робити.

Те, що сценарій знаходиться на сторінці HTML, означає, що ваші сценарії можуть бачити та копіювати всі, хто переглядає вашу сторінку. Тим не менш, на мій погляд, ця відкритість є великою перевагою, адже зворотний бік полягає в тому, що ви можете переглядати, вивчати та використовувати будь-який JavaScript, який ви зустрінете на WWW.

JavaScript може використовуватися в інших контекстах, ніж веб-браузер. Netscape створив JavaScript на стороні сервера як CGI-мову, яка може робити приблизно так само, як Perl або ASP. Немає жодної причини, через яку JavaScript не можна було б використовувати для написання реальних, складних програм. Однак цей сайт стосується виключно використання JavaScript у веб-браузерах.

Сьогодні JavaScript може виконуватись не тільки в браузері, але і на сервері або фактично на будь-якому пристрої, який має спеціальну програму під назвою механізм JavaScript. У браузері є вбудований «движок», який

іноді називають "віртуальною машиною JavaScript". У різних двигунах є різні "кодові назви". Наприклад:

## 2.4. jQuery

Простіше кажучи, jQuery - це набір попередньо написаних кодів JavaScript (відомий як бібліотека JavaScript), який ви можете додати до власних проектів.

Переваги jQuery:

1. jQuery робить програмування JavaScript більш швидким та ефективним.
2. jQuery є відкритим кодом (тобто кожен може внести свій внесок у або змінити) та має велике співтовариство користувачів, тобто його постійно підтримують та додають.
3. jQuery має велику документацію (включаючи приклади коду!)
4. jQuery добре працює з будь-якими іншими бібліотеками JavaScript, якими ви можете користуватися.
5. jQuery має дуже багато плагінів, які дозволяють розширити функціональність jQuery, коли це необхідно на проекті.

Оскільки jQuery корисний для спрощення окремих функцій (наприклад, вище), його можна розширити ще більше у вигляді плагінів - колекцій коду JS з бібліотеки jQuery, які поєднують ці окремі функції та створюють надійні функції веб-сайту та інструменти (знову ж таки, не кодуючи їх з нуля).

Плагіни створюються користувачами jQuery на основі коду в бібліотеці Query, а потім можуть публічно публікуватися в Інтернеті. Хоч і плагіни можна знайти в багатьох місцях, ті, які знайдені в офіційному

сховищі інтерфейсу jQuery (User Interface), можна сміливо вважати якісною роботою, оскільки їх курирує професійна спільнота jQuery.

## 2.5. PHP

PHP («препроцесор гіпертексту») – скриптова мова програмування, яка була створена для генерації сторінок мовою HTML на стороні веб-сервера і також для роботи з базами даних. В даний час підтримується переважною більшістю провайдерів хостингу.

Препроцесор — програма, яка виконує попередню обробку даних, для того, щоб вони могли використовуватись іншою програмою, наприклад, такою як компілятор. Про дані на виході препроцесора говорять, що вони знаходяться в препроцесованій формі, якій властива обробка подальшими програмами (компілятор). Результат і вид обробки залежать від класифікації препроцесора: так, деякі з них можуть тільки виконати просту текстову підстановку, а інші здатні змагатися з іншими скриптовими мовами програмування. Найчастіший випадок використання препроцесора – обробка первинного коду перед його відправленням на наступний крок компіляції. Мови програмування C/C++ і комп'ютерної верстки використовують препроцесори, що значно розширюють їхні можливості. Назва широко поширеної скриптової мови програмування PHP рекурсивним акронімом «PHP: Hypertext Preprocessor».

В області програмування для мережі Internet, PHP – одна з найпопулярніших скриптових мов (разом з JSP, Perl, Ruby, Python і мовами, використовуваними в ASP.NET) завдяки своїй простоті, швидкості виконання, багатій функціональності і розповсюдженню початкових кодів на основі ліцензії PHP. PHP відрізняється наявністю ядра і модулів, що підключаються, «розширень»: для роботи з базами даних, сокетом, графікою,

криптографією, документами формату PDF і т.д. Будь-який охочий може розробити своє власне розширення і підключити його. Існують сотні розширень, проте в стандартне постачання входить лише декілька десятків тих, що добре зарекомендували себе. Інтерпретатор PHP підключається до веб-серверу або через модуль, створений спеціально для цього сервера (наприклад, для Apache або IIS), або як CGI-додаток.

Окрім цього, він може використовуватися для вирішення адміністративних завдань в операційних системах UNIX, GNU/Linux, Microsoft Windows, Mac OS X і AmigaOS. Проте в такій якості він не набув поширення, віддаючи пальму першості Perl, Python і VBScript.

Синтаксис PHP подібний синтаксису мови Сі. Деякі елементи, такі як асоціативні масиви і цикл `foreach`, запозичені з Perl.

Сьогодні PHP знайшо використання серед тисячі програмістів. Кілька мільйонів сайтів повідомляють про роботу з PHP, що складає більше ніж п'ята частина доменів Інтернету.

Команда розробників мови PHP складається з безлічі людей, що добровільно працюють над ядром і розширеннями PHP, і суміжними проектами, такими, як PEAR або документація мови.

Назва PHP – рекурсивна аббревіатура, що означає «PHP: Hypertext Preprocessor» (раніше акронім розшифровувався як «Personal Home Page Tools»). Спочатку PHP створювався як надбудова над Perl для полегшення розробки веб-сторінок.

## **2.6. Фреймворки у веб-розробці**

### **2.6.1 Поняття фреймворку, веб-фреймворку**

Фреймворки - це програмні продукти, які спрощують створення і підтримку технічно складних або навантажених проектів. Фреймворк, як



правило, містить тільки базові програмні модулі, а все специфічні для проекту компоненти реалізуються розробником на їх основі. Тим самим досягається не тільки висока швидкість розробки, а й велика продуктивність і надійність рішень.

Веб-фреймворк - це платформа для створення сайтів і веб-додатків, що полегшує розробку і об'єднання різних компонентів великого програмного проекту. За рахунок широких можливостей в реалізації бізнес-логіки і високої продуктивності ця платформа особливо добре підходить для створення складних сайтів, бізнес-додатків і веб-сервісів.

### **2.6.2 Важливість фреймворків у розробці**

З точки зору бізнесу розробка на фреймворку майже завжди економічно ефективніше і якісніше за результатом, ніж написання проекту на чистому мовою програмування без використання будь-яких платформ. Розробка без використання платформи може бути правильним рішенням тільки в двох випадках - або проект зовсім простий і не вимагає подальшого розвитку, або дуже навантажений і вимагає дуже низкоуровневої оптимізації (наприклад, веб-сервіси з десятками тисяч звернень в секунду). У всіх інших випадках розробка на програмній платформі швидше і якісніше.

Якщо порівнювати фреймворки з іншими класами платформ - SaaS, CMS або CMF - то фреймворки значно ефективніше використовувати в проектах зі складною бізнес-логікою і з високими вимогами до швидкості роботи, надійності і безпеки. Але в простих і типових проектах без значущих вимог швидкість і вартість розробки на фреймворку буде вище, ніж на SaaS або CMS.

Одним з головних переваг у використанні фреймворків є те, що фреймворк визначає уніфіковану структуру для побудованих на його базі додатків. Тому додатки на фреймворках значно простіше супроводжувати і допрацьовувати, так як стандартизована структура організації компонентів

зрозуміла всім розробникам на цій платформі і не потрібно довго розбиратися в архітектурі, щоб зрозуміти принцип роботи програми або знайти місце реалізації того чи іншого функціоналу. Більшість фреймворків для розробки веб-додатків використовує парадигму MVC (модель-уявлення-контролер) - тобто дуже в багатьох фреймворках ідентичний підхід до організації компонентів програми та це ще більше спрощує розуміння архітектури додатку навіть на незнайомому розробнику фреймворку.

Проектування архітектури ПЗ при розробці на фреймворку теж дуже спрощується - в методології фреймворків звичайно закладені кращі практики програмної інженерії та просто дотримуючись цих правил можна уникнути багатьох проблем і помилок в проектуванні. По суті, фреймворк - це безліч конкретних і абстрактних класів, пов'язаних між собою і впорядкованих згідно з методологією фреймворка. Конкретні класи зазвичай реалізують взаємні відносини між класами, а абстрактні класи являють собою точки розширення, в яких закладений у фреймворк базовий функціонал може бути використаний «як є» або адаптований під завдання конкретного додатка. Для забезпечення розширення можливостей в більшості фреймворків використовуються техніки об'єктно-орієнтованого програмування: наприклад, частини програми можуть успадковуватися від базових класів фреймворка або окремі модулі можуть бути підключені як домішки.

Екосистеми веб-фреймворків також багаті на готові реалізації багатьох функціональних можливостей. Розробникам при роботі над типовими завданнями не треба «винаходити велосипеди», так як вони можуть скористатися вже створеною спільнотою реалізацією. А це не тільки скорочує витрати часу і грошей, але і дозволяє домогтися більш високої стабільності рішення - компонент, який використовується і допрацьовується тисячами інших розробників зазвичай більш якісно реалізований і краще

протестований на всіляких сценаріях, ніж рішення, яке може в адекватні терміни розробити один розробник або навіть невелика команда.



### 2.6.3 Топ-10 фреймворків PHP

Рис. 2.3 Логотип Laravel

Представлений у 2011 році, **Laravel** став найпопулярнішим в світі безкоштовним фреймворком PHP з відкритим кодом, оскільки він може безпечно обробляти складні веб-програми, значно швидшими темпами, ніж інші фреймворки. Laravel спрощує процес розробки, полегшуючи звичайні завдання, такі як маршрутизація, сеанси, кешування та аутентифікація.

#### **Переваги Laravel:**

Laravel підходить при розробці додатків зі складними вимогами до бекенда, будь то малий чи великий проект. Встановлення Laravel було полегшене введенням Homestead, попередньо упакованого в коробці.

Цей фреймворк PHP повний функцій, які допоможуть вам налаштувати складні програми. Серед них: безперебійна міграція даних, підтримка архітектури MVC, безпека, маршрутизація, механізм перегляду шаблонів та автентифікація.

Laravel відрізняється високою експресією, а його швидкість та безпека відповідають очікуванням сучасного веб-додатку. Для розробників, які хочуть створити B2B або корпоративні веб-сайти, які розвиватимуться зі зміною веб- тенденцій, Laravel - це їх вибір.



Рис. 2.4 Логотип CodeIgniter

Відомий своїм невеликим розміром (розміром є лише близько 2 Мб, включаючи документацію) **CodeIgniter** - це програма PHP, яка підходить для розробки динамічних веб-сайтів. Він пропонує численні попередньо вбудовані модулі, які допомагають створити надійні та багаторазові компоненти.

### Переваги CodeIgniter

CodeIgniter - це легкий і простий фреймворк PHP, який легко встановлювати, на відміну від інших фреймворків. Завдяки простому процесу налаштування та високоілюстрованій документації, він ідеально підходить для новачків. Основні особливості включають архітектуру MVC, неперевершене поводження з помилками, вбудовані засоби захисту, а також просту і відмінну документацію. Крім того, він створює масштабовані додатки.

У порівнянні з іншими фреймворками CodeIgniter значно швидше. Оскільки він також забезпечує високу продуктивність, це хороший вибір, коли ви хочете розробити легкі програми для роботи на скромних серверах. Одне застереження: випуски CodeIgniter дещо нерегулярні, тому фреймворк не є чудовим варіантом для програми, яка вимагає високого рівня безпеки.



Рис. 2.5 Логотип Symfony

Фреймворк **Symfony** був запущений в 2005 році, і хоча він існує набагато довше, ніж інші фреймворки цього списку, але він є надійною і зрілою платформою. Symfony - це обширний фреймворк PHP MVC і єдиний фреймворк, який, як відомо, дотримується стандартів PHP та веб-стандартів.

### Переваги Symfony:

Symfony - ідеальний вибір для розробки масштабних корпоративних проектів. Встановлення та налаштування на більшості платформ дуже просте. Одна з його ключових особливостей це багаторазові PHP компоненти. Він також може похвалитися незалежністю від «движка» бази даних. Також він стабільний, відповідає більшості найкращих практик та моделей веб-дизайну, а також дозволяє інтегруватися з іншими бібліотеками постачальників.

Symfony також дуже гнучкий і може інтегруватися з більш великими проектами, такими як Drupal. Symfony та Laravel мають багато спільних і унікальних особливостей, через що важко сказати, який з цих двох



фреймворків краще.

Рис. 2.6 Логотип CakePHP

**CakePHP** допоможе розробити візуально вражаючі, завантажені функції. Крім того, CakePHP є однією з найпростіших фреймворків для вивчення, особливо через її CRUD (створювати, читати, оновлювати та видаляти). CakePHP вийшов на ринок на початку 2000-х, і відтоді він набув кращих показників роботи та багатьох нових компонентів.

### Переваги CakePHP:

CakePHP легкий і простий в установці, оскільки вам потрібен лише веб-сервер і копія фреймворку. Це робить хорошим вибором для



комерційних програм завдяки функціям безпеки, які включають запобігання ін'єкцій SQL, перевірку вхідних даних, захист від підробки веб-сайтів (CSRF) та захист міжсхемових скриптів (XSS). Деякі ключові особливості включають сучасний фреймворк, швидку побудову, належне успадкування класів, перевірку та безпеку. Крім того, CakePHP надає чудову документацію, безліч порталів підтримки та преміальну підтримку через корпорацію Cake Development.

Рис. 2.7 Логотип Yii

Фреймворк **Yii** – насправді простий і еволюційний. Це високоефективний, створений на компонентах PHP фреймворк для розробки сучасних веб-додатків. Yii підходить для всіх видів веб-додатків. З цієї причини це універсальна структура веб-програмування.

### **Переваги Yii:**

У Yii простий процес установки. Крім того, його надійні функції безпеки роблять його придатними для високобезпечних починань, таких як проекти електронної комерції, портали, CMS, та багато інших.

Він відрізняється високою швидкістю та продуктивністю, він дуже розширюється, і це дозволяє розробникам уникати складності написання повторюваних операторів SQL, оскільки вони можуть моделювати дані бази даних з точки зору об'єктів.

Yii має основну команду розробників та експертів, які сприяють її розвитку. Завдяки масовій спільноті, яка використовує його, ви можете розміщувати проблеми на форумах Yii та отримувати допомогу. Yii надзвичайно розширюваний, і ви можете налаштувати майже кожен фрагмент коду ядра. Однак якщо ви це вперше використовуєте, будьте готові

до крутої кривої навчання.

Рис. 2.8 Логотип Zend

Фреймворк **Zend** - це повноцінна об'єктно-орієнтована структура, а той факт, що вона використовує такі функції, як інтерфейси та успадкування, робить розширюваною. Він був побудований за спритною методологією, яка допомагає вам доставляти високоякісні додатки для корпоративних клієнтів. Zend дуже настроюється і дотримується кращих практик РНР - важливий момент для розробників, які хочуть додати функціональні можливості для проекту.

### Переваги Zend:

Zend Framework - це чудова придатність для складних проектів на рівні підприємств. Він є кращою основою для великих відділів ІТ та банків. Деякі ключові особливості включають компоненти MVC, простий хмарний API,



шифрування даних та управління сесіями.

Рис. 2.9 Логотип Phalcon

Він може інтегруватися із зовнішніми бібліотеками, а ви можете використовувати лише потрібні компоненти. Фреймворк Zend оснащений надзвичайно гарною документацією та має велику базу спільноти. Однак якщо ви розробник мобільних додатків, приготуйтеся до складностей.

Повноцінна структура РНР, що використовує схему дизайну веб-архітектури MVC, **Phalcon** спочатку був написаний на С та С ++ та випущена

у 2012 році. Оскільки він постачається як розширення С, вам не доведеться



турбуватися про вивчення мови програмування на С.

### **Переваги Phalcon:**

Phalcon простий у встановленні та підходить для створення веб-додатків, які легко налаштовуються, і які відповідають настановам щодо розвитку підприємства.

Основні характеристики включають підвищену швидкість виконання, управління активами, універсальний автозавантажувач та найвищу безпеку та кешування.

На відміну від інших фреймворків, Phalcon оптимізує продуктивність завдяки ефективному використанню пам'яті. Якщо ви хочете створити швидкий веб-сайт, спробуйте Phalcon. З негативного боку, розробники Phalcon трохи повільно виправляють помилки, що може не збігатися з сьогоденішньою потребою у високому рівні безпеки.

Рис. 2.10 Логотип Swift 1

Swift - це високоефективний фреймворк програмного забезпечення мікросервісу PHP. Він публікується протягом багатьох років і став найкращим вибором для php.

### **Переваги Swift:**

Поставляється з ефективним пулом з'єднань Mysql / Redis / Rps та підключенням всіх відключень. Розробники не дбають про об'єднання з'єднань, і відповідні компоненти були реалізовані.



AOP може використовуватися для всіх об'єктів, якими керує контейнер Framework. Використання AOP дозволяє контролювати поведінку об'єктів екземпляра, не змінюючи внутрішню частину екземпляра.

Сервіс RPC поділений на RPC Server та RPC Client, і фреймворк забезпечує більш елегантний спосіб використання RPC-сервісів, таких як Dubbo. За допомогою службових мережевих фреймворків, таких як Istio / Envoy, і надає набір компонентів для швидкого побудови мікросервісного управління для



Рис. 2.11 Логотип RHPixie

малого та середнього бізнесу, включаючи реєстрацію та виявлення сервісів, долари обслуговування, зупинку обслуговування та центри налаштування.

Представлений в 2012 році і подібно до FuelPHP, **RHPixie** реалізує модель дизайну HMVC. Його метою було створити високоефективну основу для веб-сайтів, лише для читання.

### **Переваги RHPixie:**

Початок роботи з RHPixie, який підходить для веб-сайтів у соціальних мережах, налаштованих веб-додатків та служб розробки веб-додатків, легко.

Основні характеристики включають архітектуру HMVC, стандартну ORM (об'єктно-реляційне відображення), перевірку вводу, можливості авторизації, автентифікацію та кешування.

PHPixie побудований за допомогою незалежних компонентів. З цієї причини ви можете використовувати його без самих фреймворків. Зауважте, що в PHPixie є порівняно мало модулів. Крім того, йому не вистачає підтримки для компонентів, незалежно виготовлених від залежностей. Оскільки він порівняно новий, він менш популярний і має меншу спільноту користувачів, ніж інші фреймворки.

**Slim** - ще один популярний мікро-фреймворк PHP, який допомагає розробникам

швидко створювати прості, але потужні веб-програми та API.

#### **Переваги Slim:**

Так само як і PHPixie, Slim легко вивчити. Розробники PHP використовують Slim для розробки API RESTful та веб-сервісів.

Основні функції включають маршрутизацію URL-адреси, шифрування сеансів та файлів cookie, кешування HTTP на стороні клієнта тощо.



Це найкращий фреймворк для невеликого веб-додатка, який не обов'язково потребує повного пакету PHP. Крім того, активне обслуговування та дружня документація роблять Slim супер зручним для користувачів.

#### **2.6.4 Фреймворк Yii**

Зважаючи на вище наданий список, та на наявність досвіду роботи з фреймворком Yii, було вирішено вибрати його для створення веб-додатку.

Назва фреймворка - це акронім «Yes It Is!!». Серед PHP-фреймворків Yii виділяється вельми гарною продуктивністю при відносно легкому освоєнні фреймворку.

Фреймворк активно розвивається спільнотою. Yii не виглядає «монстром»

в порівнянні з фреймворками Symfony та Zend Framework, кодова базаю яких завжди є дуже об'ємна. Фреймворк Yii досить простий в освоєнні та використанні, що дозволяє швидко розроблювати проекти на ньому. Однак при виборі в якості платформи для розробки веб-проекту цей фреймворк, треба брати до уваги той факт, що швидкість розробки на ньому типових вирішень всеодно буде нижче, ніж розробка на CMS. Як і всі фреймворки, Yii «заточений» під розробку технічно складних веб-проектів: бізнес-додатків, веб-сервіси, а також сайтів із складною бізнес-логікою та вимогливих до швидкої роботи.

Основні переваги і можливості фреймворка Yii:

1. Забезпечує високу продуктивність щодо інших php-фреймворків.
2. Заснований на парадигмі MVC (Модель-Представлення-Контролер).
3. Є інтерфейси DAO і ActiveRecord для роботи з базами даних (використовується PDO).
4. Підтримує інтернаціоналізацію.
5. Дозволяє кешувати як сторінки цілком, так і окремі фрагменти.
6. Здійснює перехоплення і обробка помилок.
7. Має функціонал роботи з формами, забезпечує їх побудова та валідацію.
8. Реалізовано аутентифікація і авторизація.
9. Зручний для реалізації AJAX-інтерфейсів, інтегрується з jQuery.
10. У фреймворк вбудовані генератори базового PHP-коду для CRUD-операцій (скаффолдинг).
11. Підтримує теми оформлення.

12. Має можливість підключення сторонніх бібліотек.
13. Працює з міграціями баз даних (генерація, застосування і відкат).
14. Дозволяє здійснювати автоматичне тестування і вести розробку в стилі TDD.
15. Підтримує стиль REST.

## 2.7. Open Server

Для створення web-додатку було обрано серверну платформу Open Server. Open Server - це портативний локальний WAMP / WNMP сервер, який має багатофункціональну керуючу програму і великий вибір підключаємих компонентів. Open Server - це перший проект, який наповнений професійними інструментами та створений спеціально для веб-розробників з урахуванням їх рекомендацій і побажань.

Для налагодження скриптів у різному оточенні Open Server пропонує на вибір відразу два види HTTP серверів, різні версії PHP і СУБД модулів, а так само можливість швидкого перемикання між ними.

Основні компоненти:

- OSPanel **5.3.5**
- Apache **2.2.31 / 2.4.38 / 2.4.41**
- Bind **9.14.5**
- ConEmu **19.07.14**
- FTP FileZilla **0.9.60**
- Ghostscript **9.27**
- Git **2.23.0**
- HeidiSQL **10.2.0.5599**
- Nginx **1.17.3**
- NNCron Lite **1.17**
- Sendmail **32**

- **Wget 1.20.3**

Системы управління базами даних:

- **MariaDB 5.5.63 / 10.0.38 / 10.1.38 / 10.2.22 / 10.3.13**
- **Memcached 1.2.6 / 1.4.5**
- **MongoDB 2.4.14 / 2.6.12 / 3.0.15 / 3.2.22 / 3.4.19 / 3.6.11 / 4.0.6 / 4.2.0**
- **MySQL 5.1.73 / 5.5.62 / 5.6.43 / 5.7.25 / 8.0.15**
- **PostgreSQL 9.2.24 / 9.3.25 / 9.4.21 / 9.5.16 / 9.6.12 / 10.7 / 11.2**
- **Redis 2.8.2402 / 3.0.504 / 3.2.100**

PHP модулі:

- **PHP 5.2.17**
- **PHP 5.3.29**
- **PHP 5.4.45**
- **PHP 5.5.38**
- **PHP 5.6.40**
- **PHP 7.0.33**
- **PHP 7.1.32**
- **PHP 7.2.22**
- **PHP 7.3.9**

PHP

додатки:

- **Adminer 4.7.3**
- **PHPMemcachedAdmin 1.3**
- **PHPMyAdmin 4.9.0.1**
- **PHPPgAdmin 7**
- **PHPRedisAdmin 1.11.4**

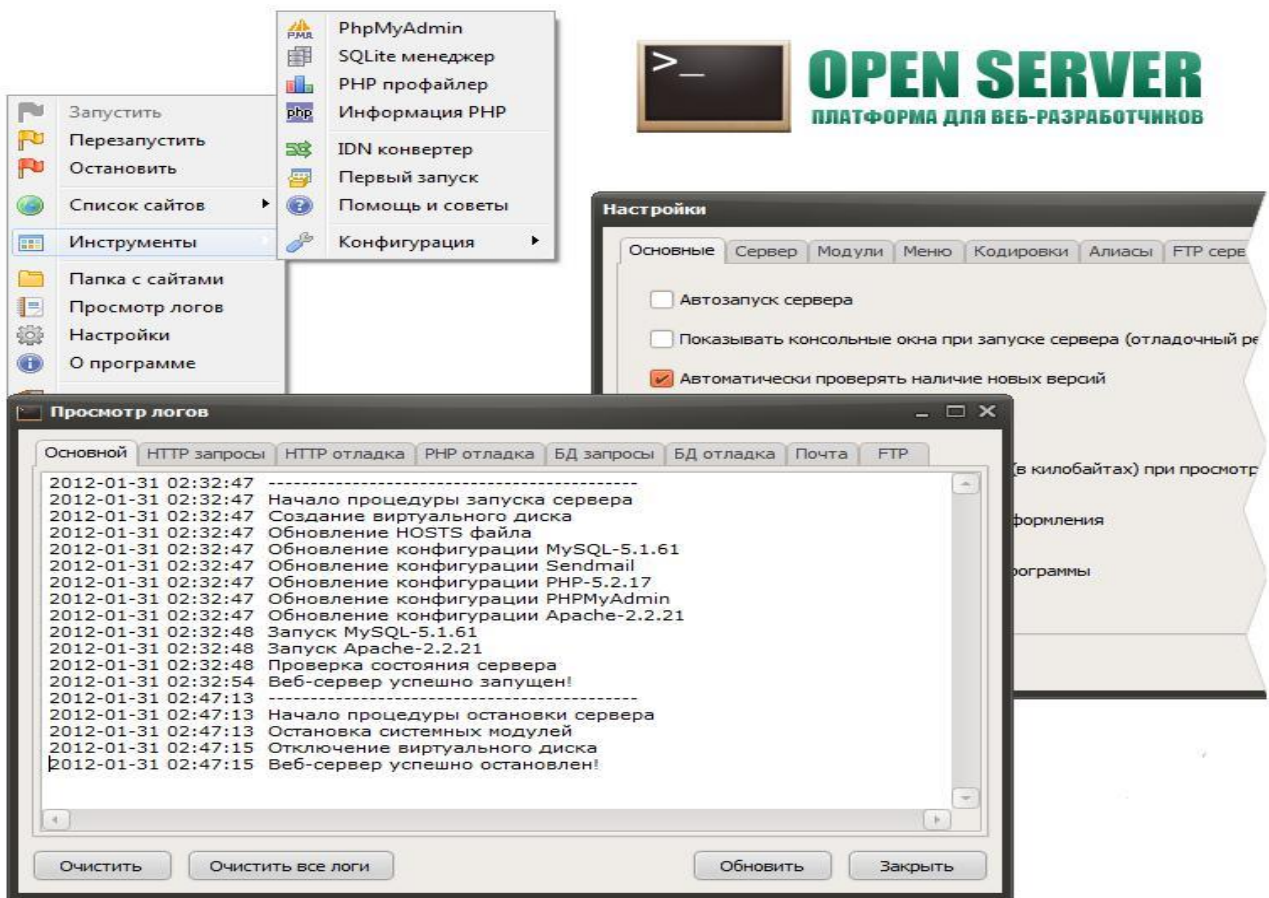


Рис. 2.14 Инструменты Open Server

Open Server - це єдиний проект у цьому сегменті, в якому є Nginx! Також тут реалізовано зручне підключення правил рерайтингу через файли .htaccess в корені домену, а PHP працює в режимі True FastCGI.

Всі компоненти цього проекту взяті з офіційних репозиторіїв і завжди оновлюються до актуальних версій з кожним оновленням пакета.

Перш за все треба відзначити, що Open Server - це цілком і повністю портативний сервер. Ніяких системних сервісів, куп сміття в реєстрі і system32. Ви можете завжди брати його з собою на флешці та запускати на робочій / домашній машині без побоювання що у вас щось не запрацює.

У разі відсутності на комп'ютері потрібних системних компонентів Open Server встановить їх сам, досить вибрати в меню [Інструменти - Перший запуск] якщо сервер запускається на комп'ютері вперше.

Не буду описувати основні можливості програми, оскільки в більшості своїй вони стандартні для такого роду софта. Само собою, що за допомогою Open Server можна запустити / зупинити сервер або відкрити потрібний домен. Набагато цікавіше поговорити про специфічні «фічі», які роблять Open Server особливим, справді особливим:

- детальний перегляд логів всіх компонентів в реальному часі;
- вибір HTTP, СУБД і PHP модулів в будь-якому поєднанні;
- підтримка SSL і кириличних доменів з коробки;
- підтримка алієсів, або по іншому - доменних покажчиків, а також дуже зручна форма їх налаштування;

- створення локального піддомена без втрати видимості основного домену

в мережі інтернет;

- доступ до доменів (в один клік) і швидкий доступ до шаблонів конфігурації модулів;

- багатомовний інтерфейс (Російська, Українська, Білоруська, Англійська);

Програма постійно удосконалюється, усі адекватні прохання з боку користувачів Open Server детально вивчаються і більшість з них реалізується.

## **2.8. ВИСНОВОК ДО РОЗДІЛУ 2**

Цей розділ демонструє технології, які будуть використовуватись при розробці Web-додатку «Система контролю задоволеності співробітників». Було розглянуті технології HTML, CSS, JavaScript, бібліотека jQuery. Також

було розглянуто мова програмування PHP та її основні фреймворки. Після опису зрівняння фреймворку було описано обраний фреймворк для створення додатку, його переваги та недоліки. Також були описані обрані інструменти для розробки, такі як IDE PhpStorm та серверна платформа Open Server, які завжди залишаться в наборі моїх інструментів для розробки, адже вони дуже зручні та наповнені потрібним функціоналом для вдалої і швидкої розробки. Головне одразу правильно обрати технології, щоб у подальшому не виникало складнощів з удосконаленням проекту.



## РОЗДІЛ 3. РОЗРОБКА ВЕБ ДОДАТКА

### 3.1. Постановка задачі

Для реалізації проекту нам потрібно створити конструктор опитувань, в якому можна створювати питання різних типів: вибір із списку, шкала, вільна відповідь, група вільних відповідей, випадючий список, матриця, ранжування, дата, ел.пошта, номер телефону, завантаження файлу. Для кожного типу питання має бути можливість ввести текст питання, додати варіанти відповіді (якщо тип питання може мати таку функцію). Також для кожного питання має бути налаштування обов'язковості відповіді на нього. Звісно кожне питання та варіант відповіді повинен видалятися за потреби. Типи питань як «вибір із списку» та «випадаючий список» мають мати налаштування мульти-вибору (коли користувач хоче вибрати декілька варіантів відповіді).

Після збереження опитування ми маємо потрапляти на сторінку «усіх опитувань» де можна продивитися загальні відомості про статус опитування, статус розсилання цього опитування, кількість відправлених \ отриманих \ продивлених листів, кількість пройдених опитувань, дата створення опитування, та дата останньої відповіді. Для кожного опитування має бути функція редагування, перегляд сконструйованого опитування (те, як користувач буде його бачити), статистика, та архівація, на випадок, якщо опитування не потрібно, але статистика має залишитись на майбутнє, або якщо при створенні нового опитування потрібно буде вибрати шаблон з готового опитування (також потрібна функція при створенні опитування).

Для кожного опитування має бути сторінка налаштування розсилки для цього опитування. На цій сторінці користувач зможе вибрати тип розсилки (email\sms) та прикріпити таблицю excel зі списком опитуваних, з певною

послідовністю назв колонок: прізвище, ім'я, по батькові, телефон, е-мейл, стать, канал розсилання (email/sms). Після додавання даних розсилки, на цій сторінці має бути можливість видалити прикріплені дані, якщо їх потрібно змінити (до тих пір поки розсилання не почалось), також перехід на сторінку, де можна продивитися завантажені дані. Звісно на цій же сторінці (налаштування розсилки) має бути перехід на сторінку налаштування змісту листа\смс-повідомлення. Після прикріплених даних розсилання та налаштування листа\смс-повідомлення має активуватись кнопка активації розсилання, після якої листи\смс-повідомлення будуть відправлені одержувачам з даних розсилки. Також має бути реалізована система оновлення статусів відправлених листів і звісно статистика, яка буде доступна після закінчення опитування (закінчування опитування має здійснюватись по натисканню кнопки, після чого опитування буде неактивне і навіть ті хто мають посилання не зможуть більше його пройти).

### **3.2. Підготовка до розробки додатку**

Для початку розробки попередньо було встановлено PHPStorm, Open Server, інтерфейс яких було проілюстровано у розділі вище.

Open Server – це дуже зручна платформа, адже у ній є все що потрібно, а користуватись нею – одне задоволення.

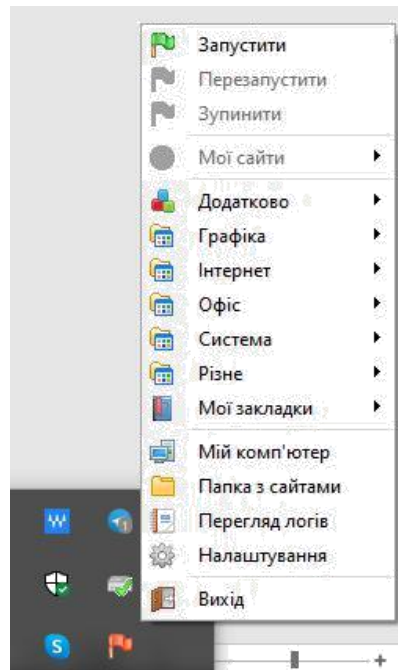


Рис. 3.1 Панель керування Open Server

Для початку нам потрібен локальний сервер, на якому ми розмістимо наш додаток. В Open Server – це питання однієї хвилини, потрібно тільки запустити платформу і відкрити панель керування, а після натиснути «Запустити».

Далі потрібно створити директорію на сервері щоб розмістити там проект. Для цього у папці Domains, що розташована в директорії Open Server створюємо папку constructor, що і буде доменом додатку на локальному сервері (constructor, тому-що основна увага додатка приділена створенню конструктора опитувань).

Далі потрібно перезапустити сервер. На наступній ілюстрації видно що домен «constructor» додався до списку «Мої сайти».

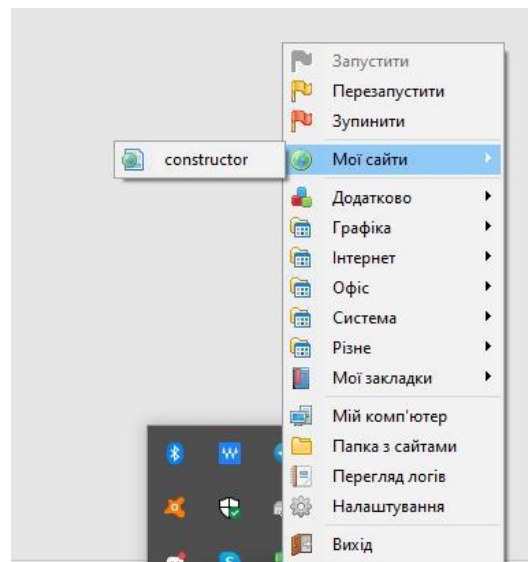


Рис. 3.2 Список доменів серверу

Щоб перевірити працездатність домену потрібно натиснути в панелі керування на наш домен, або у адресному рядку браузера ввести «constructor». Для наочної перевірки створимо в папці проекту файл `index.php` і викличимо мовну конструкцію PHP «echo» що відображає текст у HTML розмітці. Оскільки налаштування серверу не змінювалися, то початково, при посиланні на домен, відкриється файл `index.php`. Для перевірки роботи виведемо на екран привітання: «Привіт!»

```

index.php x
1 <?php
2 echo 'Привіт!';

```

Рис. 3.3 Код привітання

Весь php код має починатись з відкриваючого тегу «<?php» та закінчуватись закриваючим тегом «?>», але якщо у файлі крім php коду немає іншого, то закриваючий тег є не обов'язковим.

Отже тепер перевіримо працездатність серверу та домену, ввівши у адресний рядок браузера «constructor».

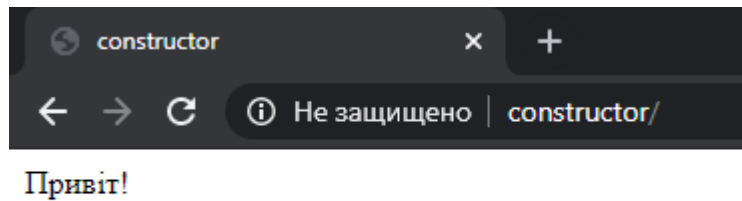
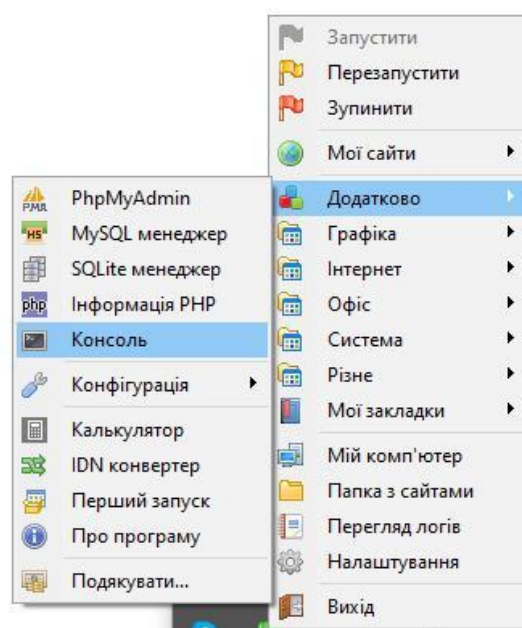


Рис. 3.4 Перевірка роботи серверу

Як видно на ілюстрації вище наш сайт працює, отже можна починати процес розробки проекту.

### 3.3. Розгортання та налаштування фреймворку Yii2 на проєкті

Для установки фреймворку нам буде потрібен пакетний менеджер для мови програмування PHP - Composer. З офіційно сайту завантажимо пакетний менеджер та встановимо його на комп'ютер. Для перевірки



роботи Composer

```

cmd
<1> cmd
Microsoft Windows [Version 10.0.18362.592]
User@DESKTOP-BIHVT94 d:\programms\openserver\ospanel
> composer

  _____
 /_   _/   _/   _/   _/   _/   _/   _/   _/   _/   _/   _/   _/   _/
/  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /
/___/___/___/___/___/___/___/___/___/___/___/___/___/___/___/___/___/___/

Composer version 1.7.1 2018-08-07 09:39:23

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
      --ansi                Force ANSI output
      --no-ansi            Disable ANSI output
  -n, --no-interaction     Do not ask any interactive question
      --profile            Display timing and memory usage information
      --no-plugins        Whether to disable plugins.
  -d, --working-dir=WORKING-DIR
                          If specified, use the given directory as working directory.
  -v|vv|vvv, --verbose    Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
                          3 for debug

Available commands:
  about                    Shows the short information about Composer.
  archive                  Creates an archive of this composer package.
  browse                   Opens the package's repository URL or homepage in your browser.
  check-platform-reqs     Check that platform requirements are satisfied.
  clear-cache              Clears composer's internal package cache.
  clearcache               Clears composer's internal package cache.
  config                   Sets config options.
  create-project           Creates new project from a package into given directory.
  depends                  Shows which packages cause the given package to be installed.
  diagnose                 Diagnoses the system to identify common errors.
  dump-autoload            Dumps the autoloader.

```

Рис. 3.5 Відкриття консолі

відкриємо консоль в інструментах Open Server:

Тепер введемо команду «composer». Як видно на рис. 3.6, ми отримали інформацію про версію та існуючі команди Composer, отже він вдало встановлений, і можна перейти до встановлення фреймворку

Рис. 3.6. результат команди

Фреймворк Yii2 має два набори пакетів: Basic та Advanced. Basic – має реалізацію звичайного додатку, Advanced – приклад більш складного додатку, який поділяється на backend, frontend, console тобто проект розділяється на три застосунки, з своїм налаштуванням. Мій проект буде складатися окремо з адмін-панелі та відображення опитувань, а також матиме консольні команди для оновлення статусу розсилання, тому мені буде потрібен Advanced.

Для завантаження Yii2 Advanced відкриваємо консоль, переходимо в папку нашого проекту та викликаємо команду «composer create-project --prefer-dist yiisoft/yii2-app-advanced yii2-advanced» і чекаємо закінчення завантаження.

backend	30.01.2020 15:53	Папка с файлами	
common	30.01.2020 15:53	Папка с файлами	
console	30.01.2020 15:53	Папка с файлами	
environments	30.01.2020 15:53	Папка с файлами	
frontend	30.01.2020 15:53	Папка с файлами	
vagrant	30.01.2020 15:53	Папка с файлами	
vendor	30.01.2020 16:02	Папка с файлами	
.bowerrc	30.01.2020 15:53	Файл "BOWERRC"	1 КБ
.gitignore	30.01.2020 15:53	Текстовый докум...	1 КБ
codeception.yml	30.01.2020 15:53	Файл "YML"	1 КБ
composer.json	30.01.2020 15:59	Файл "JSON"	2 КБ
composer.lock	30.01.2020 16:02	Файл "LOCK"	138 КБ
docker-compose.yml	30.01.2020 15:53	Файл "YML"	1 КБ
init	30.01.2020 15:53	Файл	9 КБ
init.bat	30.01.2020 15:53	Пакетный файл ...	1 КБ
LICENSE.md	30.01.2020 15:53	Файл "MD"	2 КБ
README.md	30.01.2020 15:53	Файл "MD"	3 КБ
requirements.php	30.01.2020 15:53	Файл "PHP"	6 КБ
Vagrantfile	30.01.2020 15:53	Файл	3 КБ
yii.bat	30.01.2020 15:53	Пакетный файл ...	1 КБ

Рис. 3.7 Результат завантаження

На ілюстрації вище можна спостерігати те що завантаження відбулося успішно і ми маємо раніше описану структуру.

Далі потрібно ініціювати проект. Для цього в консолі переходимо у папку з проектом та викликаємо команду «init» та обираємо [0] Development і вводимо «yes» (якщо обрати Development то це означає що будуть

відобразитись усі помилки коду, Production – відключення усіх виведень помилок). (рис. 3.8)

Наступним кроком буде створення бази даних для нашого проекту. Для цього в Open Server є веб інтерфейс для адміністрування базами

```
User@DESKTOP-BIHVT94 d:\Programms\OpenServer\OSPanel\domains
> cd constructor

User@DESKTOP-BIHVT94 d:\Programms\OpenServer\OSPanel\domains\constructor
> init
Yii Application Initialization Tool v1.0

Which environment do you want the application to be initialized in?

[0] Development
[1] Production

Your choice [0-1, or "q" to quit] 0

Initialize the application under 'Development' environment? [yes|no] yes
```

даних –

Рис. 3.8 Ініціалізація проекту

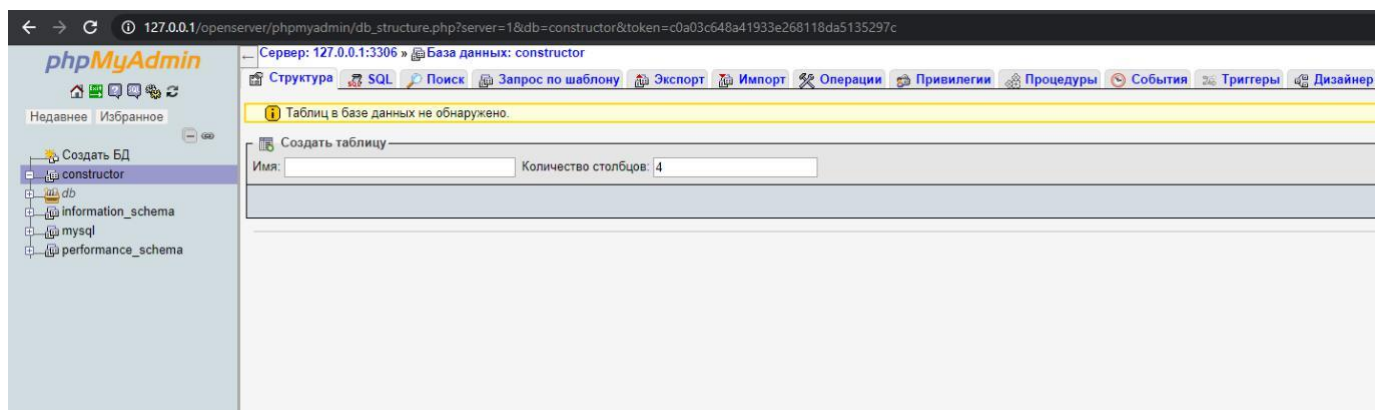


Рис. 3.9 Інтерфейс phpMyAdmin

phpMyAdmin. Через панель керування відкриваємо цей інтерфейс та створюємо базу даних (далі БД) і називаємо так як і наш проект «contstructor». При створенні вибираємо кодування `utf8mb4_general_ci`, що рекомендується при створенні БД з вмістом кирилиці.



Тепер потрібно прописати параметри для підключення до БД. Для цього в файлі `common/config/main-local.php` прописуємо параметри, зображені на наступній ілюстрації.

Після цього в консолі прописуємо команду «yii migrate», що створить у БД усі таблиці, які є в папці міграцій. Стандартно - це таблиця міграцій (для відроблених міграцій) та таблиця користувачів, з колонками: емейл, логін, зашифрований пароль та ін.

```
'components' => [  
    'db' => [  
        'class' => 'yii\db\Connection',  
        'dsn' => 'mysql:host=localhost;dbname=constructor',  
        'username' => 'root',  
        'password' => '',  
        'charset' => 'utf8',  
    ],  
],
```

Отже тепер фреймворк встановлено, і ми можемо отримати доступ до застосувань frontend і backend за посиланнями <http://constructor/frontend/web> та <http://constructor/backend/web> відповідно, що відповідає шляху до файлів `index.php` у обох застосунках. Для зручності подальшої розробки у корні проекту та в директоріях `web` кожного застосунку створимо файл конфігурації серверу

`.htaccess` з відповідними параметрами (ілюстрація вмісту файлів наведена нижче). Це робиться для того щоб посилання були зручнішими та зрозумілими для користувачів. Відтепер доступ до <http://constructor/frontend/web> буде здійснюватись за посиланням <http://constructor>, відповідно <http://constructor/admin> до «бекенду». (див. рис. 3.10 та 3.11)

Тепер за допомогою компоненту Yii2 «urlManager» налаштуємо перехід на контролери застосунків, щоб наші посилання були ще зручнішими. Наприклад доступ до контроллеру SiteController і його екшену (action) actionIndex в директорії backend буде здійснюватися за посиланням <http://constructor/admin/site/index>. Для цього в config/main.php обох директив

```

1 RewriteEngine on
2 RewriteCond %{REQUEST_FILENAME} !-f
3 RewriteCond %{REQUEST_FILENAME} !-d
4 RewriteRule . index.php

```

Рис. 3.10 Файл .htaccess в корені проекту 1

Отже тепер наш проект повністю налаштований і готовий для подальшої розробки.

Для реалізації завдання нам потрібно спроектувати відповідну БД. Ми маємо сутність опитування, яке буде мати певну кількість питань, певних типів, та у свою чергу кожне з питань матиме свої варіанти відповідей чи певні налаштування. Збережені відповіді будуть прив'язані до питання та\чи до варіанту відповіді, для зручного маніпулювання даними в фреймворці Yii2. Також потрібна таблиця «клінтів» яка буде заповнюватись при завантаженні Excel файлу за даними для розсилання. При кожному завантаженні файл Excel буде зберігатися на сервері, що займає певне місце, і з часом таких файлів може зібратися дуже велика купа, тому при видаленні опитування мають видалятись усі данні, зв'язані з цим опитуванням (питання, варіанти відповідей, відповіді, файли завантажень, таблиця клієнтів і тд). Тому буде створено таблицю для завантажень даних розсилки, де буде збережено шлях на сервері до файлу, час завантаження та ідентифікатор завантаження, до якого можна буде прив'язати клієнтів з таблиці клієнтів, для зручного маніпулювання даними. Авжеж потрібна таблиця історії відповідей, яка буде пов'язувати «клієнта» та його відповіді в таблиці відповідей. І накінець нам потрібно дві таблиці: перша для збереження налаштувань змісту повідомлень,

та таблиця в якій при віправці листа буде зберігатись його ідентифікатор, ідентифікатор отримувача (з таблиці клієнтів) та ідентифікатор опитування, знову ж таки, для зручного маніпулювання.

В фреймворці Yii використовується архітектурний шаблон MVC, тобто напочатку ми звертаємося до певного контроллера, який має набір екшені. Далі ми звертаємося до окремого екшену, який в свою чергу, за необхідністю,

викликає модель (трохи далі буде детальніше описані моделі) і отримує певні дані для подальшої обробки і накінець, після закінчення усіх вчислень та операцій, екшн звертається до файлу виду ( View) що представляє собою HTML код із вставками PHP. Контроллер передає необхідні данні до файлу виду і ми отримуємо унікальну сторінку, в залежності від ситуації.

Таблиці БД в фреймворці представлені як «моделі», це класи які наслідуються від класу ActiveRecord, який в свою чергу від класу Model. Active record дозволяє звертатись до таблиці БД у представленні ООП. У наступному

### 3.13 Модель таблиці опитування

рисунку створено модель для таблиці сутності «опитування».

Відтепер замість довгих та незручних SQL запитів до таблиці опитувань можна звертатись у вигляді ООП. Наприклад, якщо ми хочемо отримати дані

```
$quiz = Quiz::findAll(['status' => 'active']);
```

Рис.3.13 Звернення до таблиці опитувань

усіх опитувань, у яких, наприклад, активний статус (див. рис. 3.14)

Результатом, який ми помістили у змінну `$quiz` є масив об'єктів, тобто екземплярів класу `Quiz`, який, нагадаю ще раз, наслідюється від класу `ActiveRecord`, а той від `Model`. Це означає, якщо отримати окремо об'єкт `Quiz`, до нього, звертаючись у формі ООП можна використати усі методи та функції з наслідуваних класів, а також легко можна звертатись до колонок запису. Наприклад ми отримуємо об'єкт класу `Quiz`, і хочемо отримати його дату створення (колонка в таблиці називатиметься `created_at`), буде виглядати це наступним чином:

```
$quiz = Quiz::findOne();
$created_at = $quiz->created_at;
```

Рис. 3.14 Отримання часу створ.

Ще однією великою перевагою моделей фреймворку – це зв'язки між таблицями. Для того щоб зв'язувати таблиці не потрібно створювати їх зв'язки у самій БД MySQL. Для цього достатньо мати ідентифікатор першої таблиці в колонці другої таблиці (чи навпаки), після чого можна прописати зв'язок таблиць з будь якої сторони. Наприклад, ми матимемо таблицю опитувань (назвемо її `quiz` – один з варіантів слова «опитування» англійською мовою, воно коротке та зручне) і таблицю питань опитування (назвемо `quiz_questions`).

Таблиця `quiz` не повинна мати в собі дані ідентифікаторів усіх своїх питань, з таблиці питань. А ось при створенні питання, воно має бути прив'язано до опитування, тому запис питання буде мати в собі ідентифікатор опитування.

Фреймворк дозволяє легко і зручно описати зв'язки. Якщо ми, маючи об'єкт опитування хочемо отримати усі питання, які містять ідентифікатор цього опитування, то нам потрібно в моделі таблиці `quiz` створити наступну функцію

```
public function getQuestions()
{
    return $this->hasMany(QuizQuestions::className(), ['quiz_id' => 'id'])->orderBy(['question_order' => 'SORT_ASC']);
}
```

### Рис. 3.15 Зв'язок опитування з питаннями

Таким чином, щоб отримати усі записи в таблиці питань, які відносяться до певного опитування, потрібно на об'єкт `quiz` викликати цю функцію (див. рис. 3.16)

Далі отримані питання можна обробити за допомогою мовної конструкції `foreach`, та в свою чергу для кожного питання отримати відповіді (заздалегіть прописавши зв'язки між цими таблицями) (див. рис. 3.17)

Після створення усіх таблиць та написання всіх зв'язків між ними, можна зобразити цю систему наступною діаграмою:

```
// Отримуємо об'єкт Quiz  
$quiz = Quiz::findOne();  
// Отримуємо quiz_questions  
$quiz_questions = $quiz->getQuestions()->all();  
// Або прощений варіант, завдяки "магічному" методу  
$quiz_questions = $quiz->questions;
```

Рис 3.16 Отримання питань опитувань завдяки зв'язкам моделей

```

// Отримуємо об'єкт Quiz
$quiz = Quiz::findOne();
// Отримуємо quiz_questions
$quiz_questions = $quiz->getQuestions()->all();
// Або прощений варіант, завдяки "магічному" методу
$quiz_questions = $quiz->questions;
foreach ($quiz_questions as $question)
{
    $answer = $question->getAnswer()->one();
    // Або
    $answer = $question->answer;
}

```

Рис. 3.17 Отримання відповідей

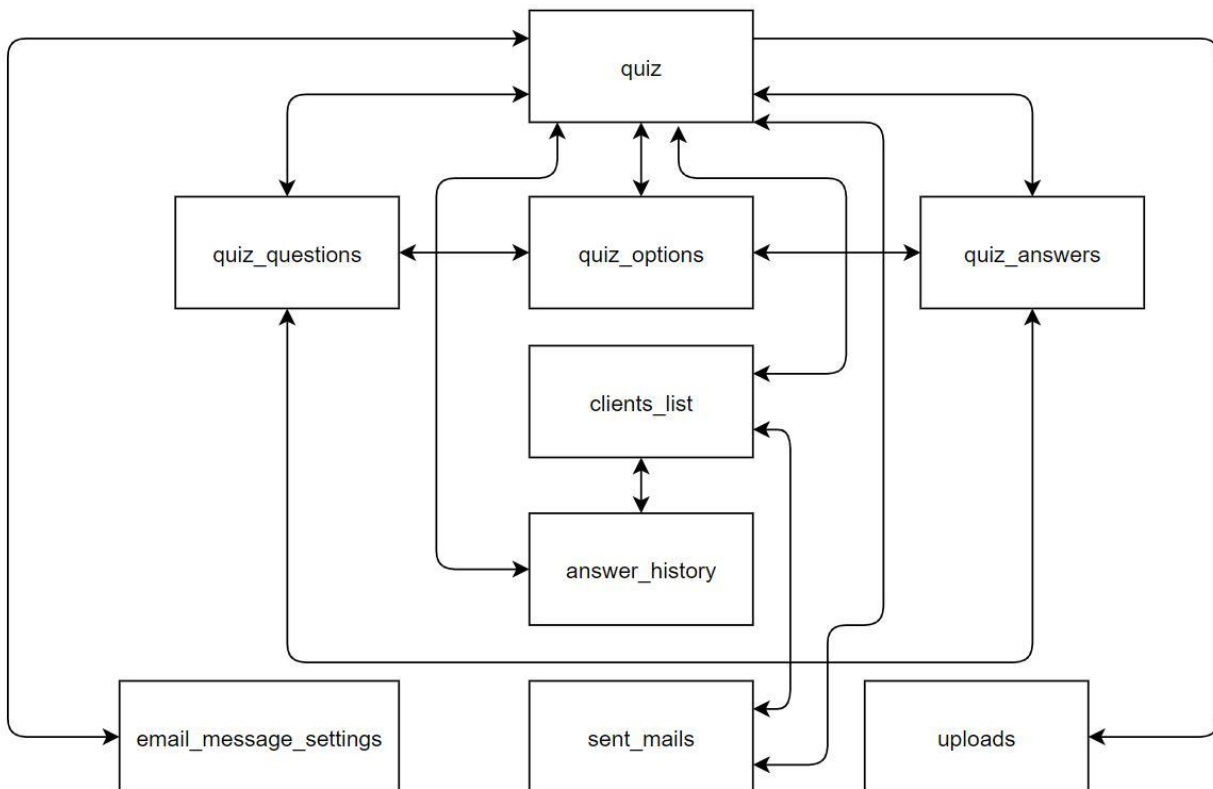
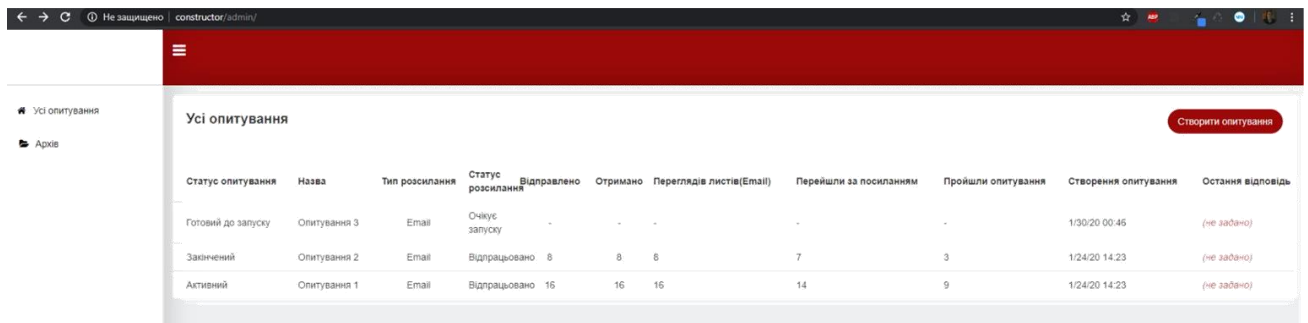


Рис 3.18 Діаграма зв'язків таблиць

### 3.5 Опис готового проекту

Нажаль чи на радість проект вийшов дуже об'ємним, і описувати кожну модель, кожен зв'язок, усі контролери та екшени, усі сторінки виду і тд буде

не дуже доцільно, тому у цьому пункті буде описани вже готовий функціонал проекту, а основний код проекту буде прикладено у вигляді додатків. На наступному рисунку зображена головна сторінка, на якій можна бачити список усіх опитувань, їх основні дані, та можливість перейти до налаштувань окремого опитування, а також кнопка створення нового опитування. (див рис.



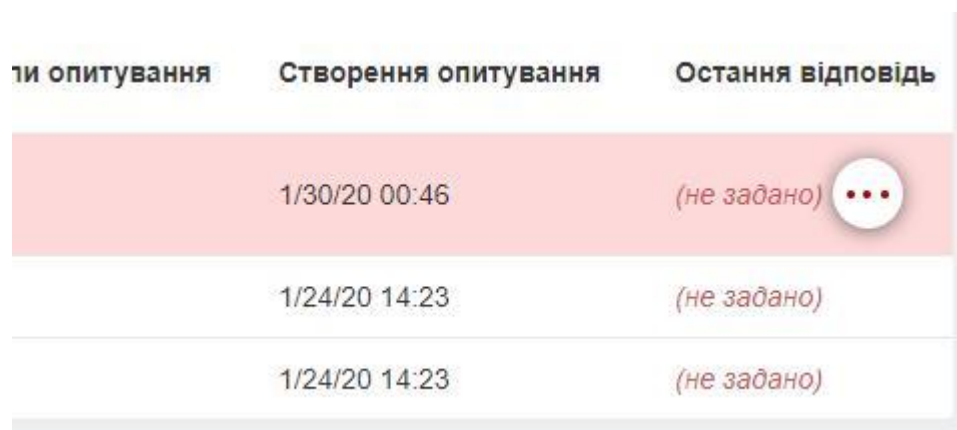
The screenshot shows a web application interface with a red header and a sidebar on the left. The main content area displays a table titled "Усі опитування" (All Surveys). The table has columns for status, name, type, email status, and various counts. A red button "Створити опитування" (Create Survey) is visible in the top right corner of the table area.

Статус опитування	Назва	Тип розсилання	Статус розсилання	Відправлено	Отримано	Переглядів листів(Email)	Перейшли за посиланням	Пройшли опитування	Створення опитування	Остання відповідь
Готовий до запуску	Опитування 3	Email	Очікує запуску	-	-	-	-	-	1/30/20 00:46	(не задано)
Закінчений	Опитування 2	Email	Відпрацьовано	8	8	8	7	3	1/24/20 14:23	(не задано)
Активний	Опитування 1	Email	Відпрацьовано	16	16	16	14	9	1/24/20 14:23	(не задано)

3.19).

Рис. 3.19 Головна сторінка проекту

На наступному рисунку 3.21 зображені функції та сторінки для кожного опитування. При наведенні курсором на необхідне опитування із списку, воно підсвічується і відображається кнопка, при натисканні якої відображається меню (рис. 3.20).



This image is a close-up of the table from Figure 3.19, highlighting the hover effect. The row for "Опитування 3" is highlighted in light red. A circular button with three dots (a menu icon) is visible in the "Остання відповідь" column for this row.


ти опитування	Створення опитування	Остання відповідь
	1/30/20 00:46	(не задано) 
	1/24/20 14:23	(не задано)
	1/24/20 14:23	(не задано)

Рис. 3.20 Hover ефект

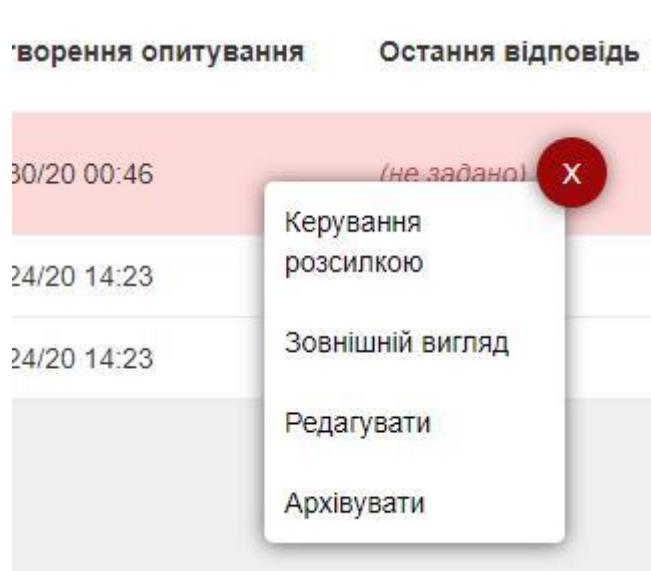


Рис. 3.21 Меню опитування

На ілюстрації вище можна бачити елементи меню опитування, такі як «керування розсилкою», «зовнішній вигляд», «редагувати», «архівувати», призначення яких розкрито у постановці завдання.

Після натискання кнопки «створити опитування» відкриється модальне вікно з формою для введення назви опитування та (по бажанню) вибору шаблону готового опитування. (рис.3.22)



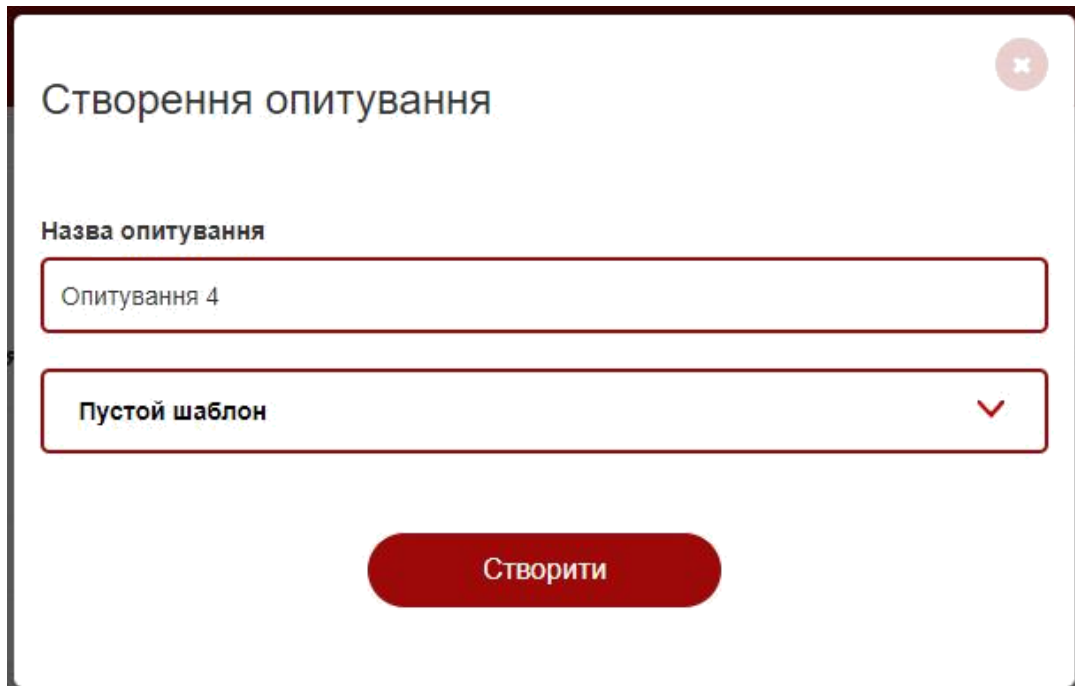


Рис. 3.22 Модальне вікно створення опитування

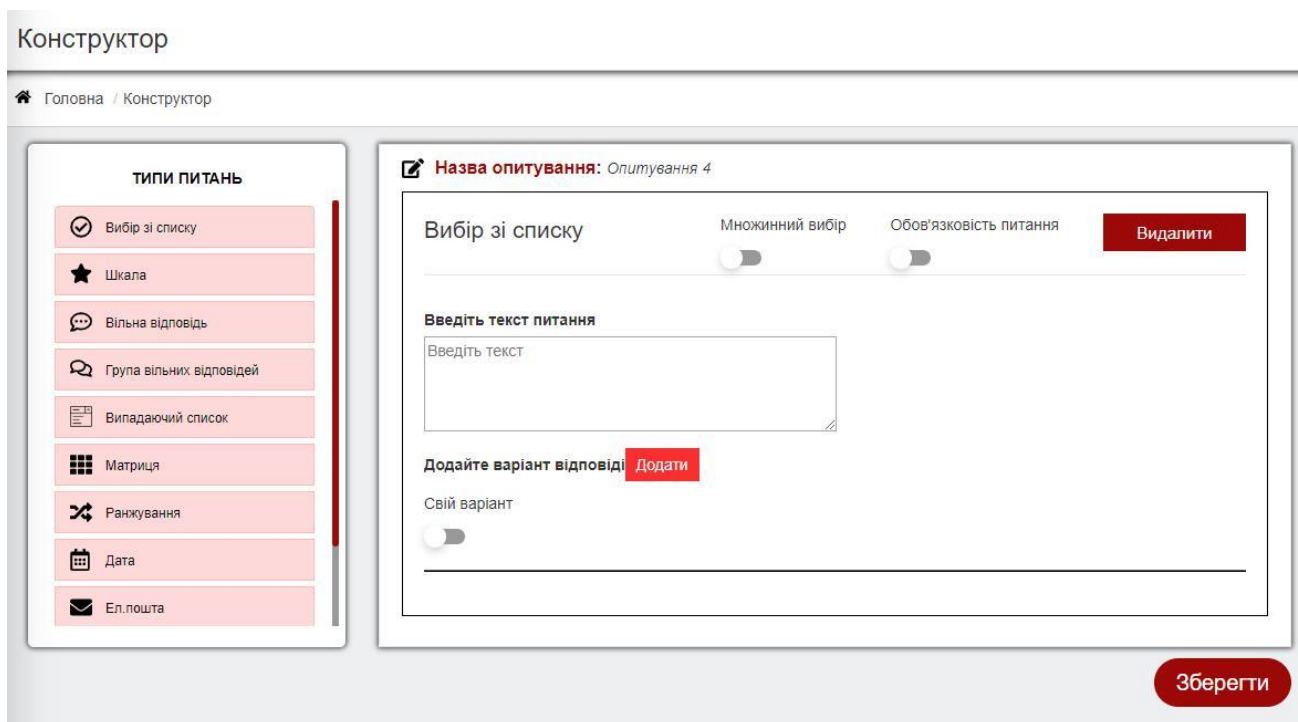


Рис. 3.23 Сторінка конструктору

Після заповнення даних і натискання «створити» ми потрапляємо на сторінку конструктору опитувань. (рис. 3.23)

Шляхом перетягування типу питання у область конструктора, в останній створюється профіль цього питання з усіма налаштуваннями. На наступній

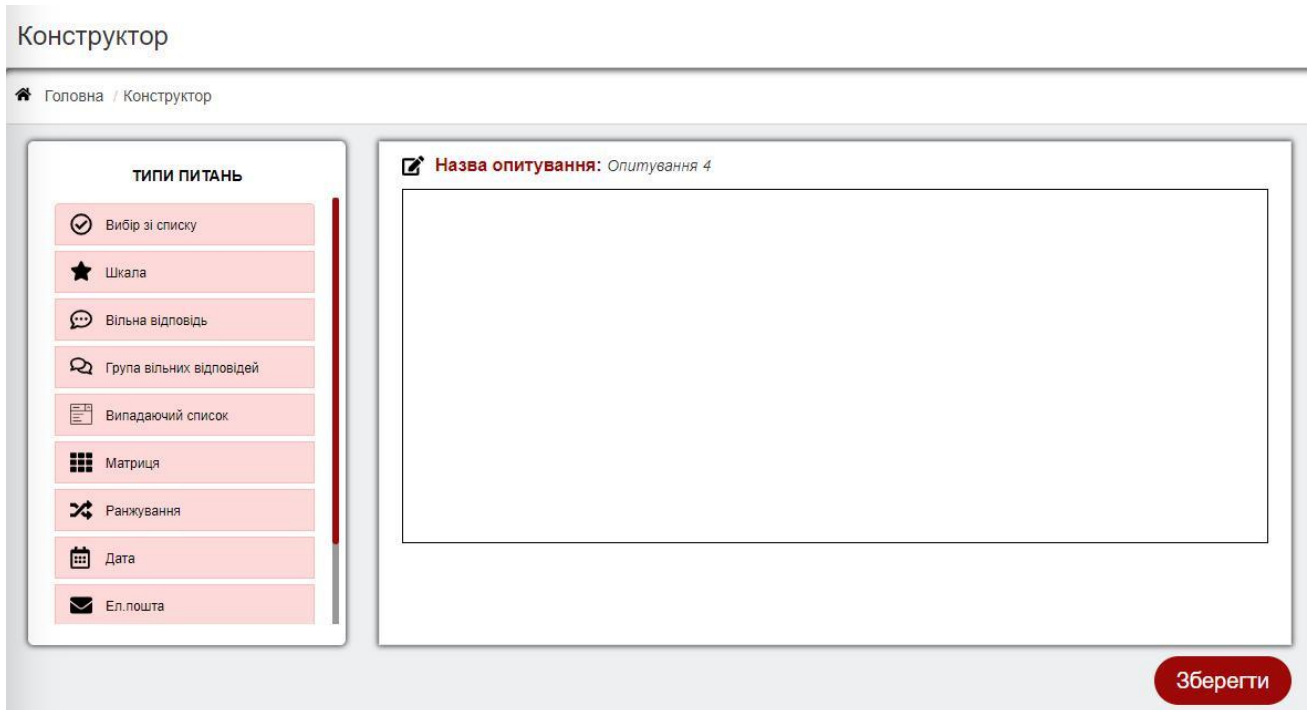


Рис. 3.24 Додавання питання до області конструювання

ілюстрації наведено приклад, у якому до області конструктора додано «Вибір зі списку».

Як видно на рис. 3.24 в налаштуванні питання виконані поставлені задачі, а також додатково було розроблено функцію «свій варіант», якщо при створенні опитування респонденту хочуть надати можливість крім наданих варіантів відповідей ввести свій варіант.

Між собою питання у області конструювання переміщуються за допомогою перетягувань.

Для наочності створимо маленьке тестове опитування, де буде три питання: «Ваша стать», «Ваш вік», «Оцініть вашу задоволеність роботою в компанії».

Перше і друге питання реалізуємо завдяки типу питань «вибір із списку», а третє за допомогою питання «шкала». Варіанти відповіді для першого

← → ↻ ⓘ Не захищено | constructor/site/view-one?id=38

## Опитування 4

1. Ваша стать

Чоловіча  
 Жіноча

2. Ваш вік

18-25  
 25-45  
 45-60

3. Оцініть вашу задоволеність роботою в компанії

0  
1  
2  
3  
4  
5  
6  
7  
8   
9  
10

питання : «чоловіча», «жіноча», для другого «18-25», «25-45», «45-60», а для третього оберемо шкалу від 0 до 10. Таким чином після збереження опитування і перегляді його зовнішнього виду можемо бачити наступне (див. рис. 3.25)

Рис. 3.25 Зовнішній вид сконструйованого опитування

Опитування "Опитування 4"

**Назад**

Прив'яжіть дані для розсилання

Тип розсилання

Email  Phone

Ехсел файл

Выберите файл | Файл не выбран

Зберегти

Рис. 3.26 Форма для завантаження даних розсилки

Далі, якщо перейти у налаштування розсилкою опитування ми побачимо. Оберемо тип «email» та завантажимо ехсел у форматі, зазначеному в постановці задачі. Після збереження сторінка керуванням розсилання оновиться і матиме наступний вигляд:

Лист опитування "Опитування 4"

**Назад**

**Картинка**

Выберите файл | опитування.png

**Тест привітання**

Добрий день!

**Текст листа**

Будь ласка, пройдіть опитування компанії

**Тексти кнопки переходу на опитування**

Пройти опитування

**Зберегти**

Рис. 3.27 Вигляд сторінки після завантаження даних

Як бачимо сторінка має запланований функціонал. Звісно, щоб запустити розсилання потрібно налаштувати зміст листа. Кожен лист матиме однакову структуру, а ось текст і картинку в шапці листа можна налаштовувати за потреби.

Після цього, на сторінці налаштування активується кнопка «запустити розсилання», після натискання якої відпрацює розсилання, а також ця кнопка заміниться на «закінчити опитування».

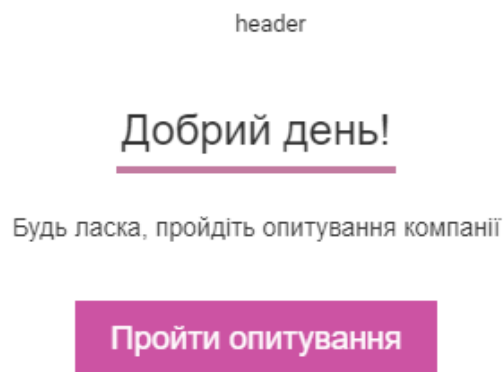
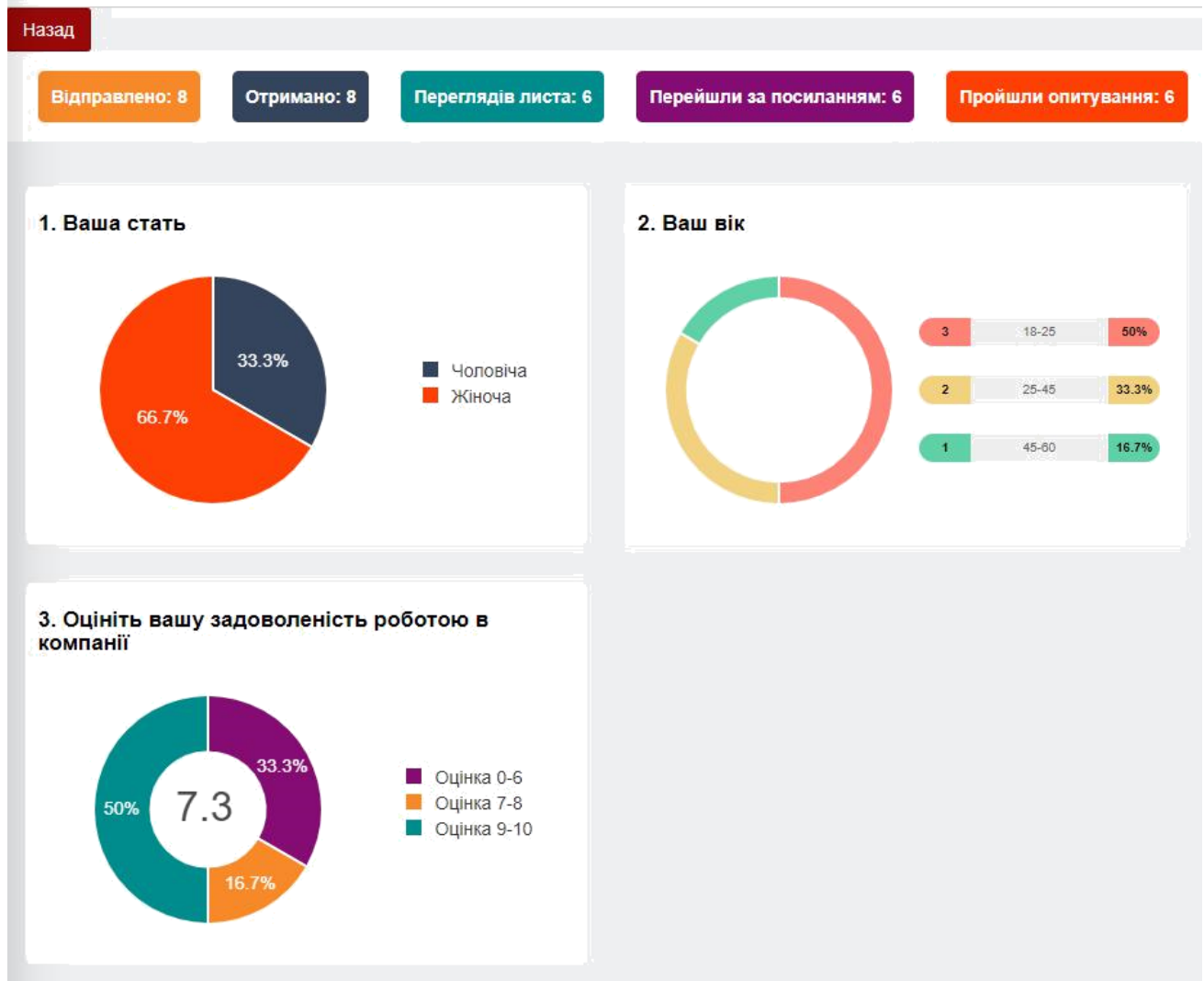


Рис. 3.29 Надійшовший лист, до кожного респондента

На рисунку 3.29 ми можемо спостерігати лист, який надійшов до кожного респондента. Нажаль картинка в шапці опитування не відображається, оскільки розробка ведеться на локальному сервері, і з мережі Інтернет до неї немає з'єднання, тому замість картинки ми можемо спостерігати надпис «header» що є описом функціональним описом цієї картини. Перейшовши за посиланням, респондент потрапить на зовнішній вид опитування та зможе його пройти.

Відправка, отримання, читання листа відстежується. Також відстежується перехід по посиланню опитування і звісно проходження опитування. Під час

## Статистика опитування "Опитування 4"



тесту усі листи відправляються на мою особисту пошту, завдяки чому можна відтворити процес проходження опитування для прикладу.

Пройшовши певну кількість опитувань запустимо консольну команду, яка оновлює статуси листів. Далі закінчимо опитування, натиснувши на кнопку «закінчити опитування» на сторінці керування розсилкою. Після чого стане доступна зручна статистика в графіках, де наочно можна побачити результати опитування (див. рис. 3.30)

Рис. 3.30 Статистика опитування

Як видно на ілюстрації вище, статистика відображає статистику розсилки, та

статистику відповідей опитування у графіках. Кожен тип питання відображається унікальним типом графіка, так, наприклад, графік для шкали відображає середнє арифметичне число відповідей по центру.

### **3.4. ВИСНОВОК ДО РОЗДІЛУ 3**

Цей розділ демонструє як можна реалізувати просту систему контролю задоволеності співробітниками у вигляді Web-додатку. Використовуючи якісне програмне забезпечення та правильні і зручні технології, побудова додатку не стала проблемою. Було використано велику кількість технологій. В першу чергу це мова розмітки HTML, таблиці стилей CSS та об'єтно-орієнтована мова програмування PHP, фреймворк Yii2, серверна платформа Open Server, середовище програмування phpStorm.

Web-додаток орієнтований в першу чергу на керівників компаній та менеджерів, але цю систему також можна інтегрувати у різні середовища та використовувати за різних потреб, наприклад для особистого розсилання запрошення друзям. В проекті створено простий і зручний дизайн для конструктора опитувань, але нажаль за браком часу та досвіду не вдалося стилізувати зовнішній вид опитування.

## ВИСНОВКИ

У ході виконання дипломного проекту були досліджені та випробувані методи створення Web-додатку системи контролю задоволеності співробітників.

Проведено дослідження ефективності використання різних технологій для реалізації функціонального і готового до експлуатації Web-додатку. Для цього було проаналізовано переваги програмного забезпечення, редакторів вихідного коду, фреймворків та засобів, які допомагають полегшити розробку.

Також були розглянуті питання необхідності такої системи в наші часи, описано важливість покращення середовища для робітників та своєчасного виявлення проблем в компанії, що в свою чергу дозволить покращити виробництво підприємства.

Розглянуті та використані технології показали, що розумний підбір технологій для розробки проекту може полегшити процес розробки та навіть зробити його цікавим.

В результаті дипломної роботи було отримано працездатний web-додаток, який можна використовувати як систему контролю задоволеності співробітників так і за особистими чи корпоративними цілями. Наприклад можна робити розсилку для запрошення друзів, чи колег на захід, та запитати в них про їх вподобання і інтереси. Чи також після заходу провести опитування про його організацію, отримати побажання та рекомендації.

Незаперечно цей проект буде і надалі розвиватись, в планах розвивати готовий функціонал, додавати новий, робити можливість інтегрування проекту у інші, покращувати зовнішній вид та зручність використання.



**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. hrliga.com [Електронний ресурс]. – Режим доступу: <https://hrliga.com/index.php?module=profession&op=view&id=1676>
2. lifewire.com [Електронний ресурс]. – Режим доступу: <https://www.lifewire.com/what-is-a-web-application-3486637>
3. habr.com [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/450282/>
4. en.yeeply.com [Електронний ресурс]. – Режим доступу: <https://en.yeeply.com/blog/advantages-and-disadvantages-of-web-app-development/>
5. svitla.com [Електронний ресурс]. – Режим доступу: <https://svitla.com/blog/web-application-architecture>
6. slideshare.net [Електронний ресурс]. – Режим доступу: <https://www.slideshare.net/ssusere5f319/web-66422826>
7. wikipedia.org [Електронний ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/Доменна\\_система\\_імен](https://uk.wikipedia.org/wiki/Доменна_система_імен)
8. wikipedia.org [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/HTML>
9. tutorialspoint.com [Електронний ресурс]. – Режим доступу: [https://www.tutorialspoint.com/css/what\\_is\\_css.htm](https://www.tutorialspoint.com/css/what_is_css.htm)
10. wikipedia.org [Електронний ресурс]. – Режим доступу: <https://en.wikipedia.org/wiki/PHP>
11. habr.com [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/464417/>
12. web-creator.ru [Електронний ресурс]. – Режим доступу: <https://web-creator.ru/articles/yii>

13. jetbrains.com [Электронный ресурс]. – Режим доступа:  
<https://www.jetbrains.com/phpstorm/>
14. habr.com [Электронный ресурс]. – Режим доступа:  
<https://habr.com/ru/post/137388/>
15. computerhope.com [Электронный ресурс]. – Режим доступа:  
<https://www.computerhope.com/jargon/c/css.htm>
16. yii2-framework.readthedocs.io [Электронный ресурс]. – Режим доступа:  
<https://yii2-framework.readthedocs.io/en/latest/guide-ru/tutorial-advanced-app/>