

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

Приходько С. Б.

« ____ » _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня вищої освіти «магістр»

на тему: Математична модель для оцінювання розміру WordPress тем на PHP та розробка програмного забезпечення для її реалізації

Виконав: студент 6151м групи

 Дорогань І.Ю.
(підпис, ПІБ)

Керівник роботи:

Доцент, к.т.н.

(посада, науковий ступень вчене звання)

Пухалевич А.В.
(підпис, ПІБ)

(прізвище та ініціали)

Миколаїв – 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**Національний університет кораблебудування імені адмірала Макарова**

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

Гарант освітньої програми

_____ проф. С.Б.Приходько

(підпис)

«10» жовтня 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «магістр»

Студенту Дороганю Іллі Юрійовичу

(Прізвище, ім'я, по батькові)

1. Тема роботи: Математична модель для оцінювання розміру WordPress тем на PHP та розробка програмного забезпечення для її реалізації

Керівник роботи: доцент каф. ПЗАС, к.т.н. Пухалевич А.В.

Затверджені наказом ректора № 798 уч від «28» _____ 09 _____ 2022 р.

2. Термін подання роботи: 14.11.2022 р.

3. Вихідні дані по роботі:

4. Перелік питань, що належать до розробки (найменування розділів)

Вступ (Актуальність теми. Зв'язок роботи з науковими програмами, планами, темами. Мета і завдання дослідження. Об'єкт дослідження. Предмет дослідження. Методи дослідження. Наукова новизна одержаних результатів. Практичне значення одержаних результатів. Особистий внесок здобувача. Апробація результатів досліджень. Публікації.); Огляд літератури та обґрунтування необхідності проведення досліджень за обраною темою; Викладення результатів власних досліджень з висвітленням того нового, що пропонується; Проект програмного забезпечення; Організаційно-економічний розділ; Розділи з охорони праці та охорони навколишнього середовища; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма і методика випробувань програмного забезпечення)

5. Перелік презентаційних матеріалів

1. Титульний слайд. 2. Актуальність теми. 3. Мета, об'єкт, предмет, методи дослідження. 4. Наукова новизна та практичне значення; 5. Апробація та публікації. 6. Метод побудови нелінійних регресійних моделей з використанням нормалізуючих перетворень. 7. Лінійна регресійна модель для нормалізованих значень. 8. Нелінійна регресійна модель для оцінювання розміру WordPress тем на PHP. 9. Постановка задачі на розробку програмного забезпечення. 10. Діаграма варіантів використання. 11. Логічна модель даних. 12. Вибір мови програмування та СКБД. 13. Фізична модель даних. 14. Випробування програмного забезпечення. 15. Висновки.

6. Консультанти розділів роботи

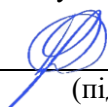
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 10.10.2022 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу Вступ	11.10.2022	НС*
2.	Підготовка розділу з огляду літератури та обґрунтування необхідності проведення досліджень за обраною темою	12.10.2022	НС*
3.	Підготовка розділу (ів) з результатів власних досліджень	13.10.2022	НС*
4.	Підготовка розділу з проєкту програмного забезпечення	03.11.2022	
5.	Підготовка організаційно-економічного розділу	06.11.2022	
6.	Підготовка розділу з охорони праці	09.11.2022	
7.	Підготовка розділу з охорони навколишнього середовища	11.11.2022	
8.	Підготовка розділу Висновки	12.11.2022	
9.	Оформлення списку використаних джерел та додатків	13.11.2022	
10.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версії роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	14.11.2022	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку)
11.	Підготовка презентації та доповіді	19.11.2022	
12.	Попередній захист роботи на засіданні кафедри ПЗАС	20.11.2022	
13.	Подання на кафедру ПЗАС електронних версії наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді	28.11.2022	

* - за результатами наукового стажування (НС), яке було з 01.09.2022 по 09.10.2022 р.

Студент		Дорогань І.Ю.
	(підпис)	(ПІБ)
Керівник роботи		Пухалевич А.В.
	(підпис)	(ПІБ)

РЕФЕРАТ

Дорогань Ілля Юрійович

Математична модель для оцінювання розміру WordPress тем на PHP та розробка програмного забезпечення для її реалізації

Кваліфікаційна робота на здобуття освітнього рівня магістра зі спеціальності 121 – «Інженерія програмного забезпечення». Національний університет кораблебудування імені адмірала Макарова. Миколаїв, 2022 р.

Обсяг роботи: 83 сторінки, 9 таблиць, 14 рисунків, 14 використаних джерел, 5 додатків.

Актуальність теми роботи: полягає в тому, що оцінювання розміру WordPress тем на PHP є актуальною задачею, виходячи з поширеності системи керування вмістом WordPress та тем до неї розроблених з використанням мови програмування PHP.

Мета роботи: підвищення достовірності оцінювання розміру WordPress тем на PHP.

Завдання дослідження: проаналізувати існуючі математичні моделі для оцінювання розміру WordPress тем на PHP; аргументувати необхідність побудови математичної моделі для оцінювання розміру WordPress тем на PHP; побудувати математичну модель для оцінювання розміру WordPress тем на PHP; розробити програмне забезпечення, що реалізує побудовану математичну модель для оцінювання розміру WordPress тем на PHP.

Об'єкт дослідження: процес оцінювання розміру WordPress тем на PHP.

Предмет дослідження: нелінійна регресійна модель для оцінювання розміру WordPress тем на PHP.

Методи дослідження: методи теорії ймовірності, математичної статистики, регресійного аналізу.

Наукова новизна одержаних результатів полягає в удосконаленні трьохфакторної нелінійної регресійної моделі для оцінювання розміру PHP застосунків із відкритим кодом за рахунок знаходження параметрів моделі для WordPress тем на PHP, що дозволяє підвищити достовірність оцінювання їх розміру в порівнянні з існуючими трифакторними нелінійними регресійними моделями.

Практичне значення одержаних результатів. Розроблено програмне забезпечення для оцінювання розміру WordPress тем на PHP.

Ключові слова: нелінійна регресійна модель, нормалізуюче перетворення, оцінювання розміру WordPress тем на PHP.

ABSTRACT

Dorohan Ilya Yuriyovych

A mathematical model for estimating the size of WordPress themes in PHP and software development for its implementation

Qualification work for obtaining a master's degree in specialty 121 - "Software engineering". Admiral Makarov National Shipbuilding University. Mykolaiv, 2022

Volume of work: 83 pages, 9 tables, 14 pictures, 14 used sources, 5 appendices.

The relevance of the topic of the work: is that estimating the size of WordPress themes on PHP is an actual task, based on the prevalence of the WordPress content management system and themes developed for it using the PHP programming language.

The purpose of the work: to improve the reliability of the estimation of the size of WordPress themes on PHP.

Research task: to analyze the existing mathematical models for estimating the size of WordPress themes in PHP; to argue the need to build a mathematical model for estimating the size of WordPress themes in PHP; build a mathematical model for estimating the size of WordPress themes in PHP; develop a software that implements a built-in mathematical model to estimate the size of WordPress themes in PHP.

Research object: the process of evaluating the size of WordPress themes in PHP.

Research subject: Non-linear regression model for estimating the size of WordPress themes in PHP.

Research methods: methods of probability theory, mathematical statistics, regression analysis.

The scientific novelty of the obtained results lies in the improvement of the three-factor nonlinear regression model for estimating the size of PHP open-source applications by finding model parameters for WordPress themes in PHP, which allows to increase the reliability of their size estimation in comparison with existing three-factor nonlinear regression models.

Practical significance of the obtained results. Developed software for estimating the size of WordPress themes in PHP.

Keywords: non-linear regression model, normalizing transformation, size estimation of WordPress themes in PHP.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ	7
ВСТУП.....	8
1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА МОДЕЛЕЙ ОЦІНЮВАННЯ РОЗМІРУ WORDPRESS ТЕМ НА PHP	11
1.1 WordPress як найрозповсюдженіша система керування вмістом вебсайтів, та її особливості	11
1.2 Аналіз існуючих методів та моделей для оцінювання розміру програмного забезпечення	13
1.3 Обґрунтування необхідності проведення досліджень за обраною темою.....	14
2 ПОБУДОВА НЕЛІНІЙНОЇ РЕГРЕСІЙНОЇ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WORDPRESS ТЕМ НА PHP	15
2.1 Аналіз підходу до побудови нелінійної регресійної моделі та вибір факторів для оцінювання розміру WordPress тем на PHP.....	15
2.2 Перевірка емпіричних даних для оцінювання розміру WordPress тем на PHP, на наявність викидів.....	16
2.3 Побудова нелінійної регресійної моделі для оцінювання розміру WordPress тем на PHP	20
3 РОЗРОБКА ПРОЄКТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WORDPRESS ТЕМ НА PHP	25
3.1 Ескізний проєкт програмного забезпечення для оцінювання розміру WordPress тем на PHP	25
3.2 Технічний проєкт програмного забезпечення для оцінювання розміру WordPress тем на PHP	29
3.3 Робочий проєкт програмного забезпечення для оцінювання розміру WordPress тем на PHP	31
3.4 Результати розробки програмного забезпечення для оцінювання розміру WordPress тем на PHP	41
4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ І ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WORDPRESS ТЕМ НА PHP	43

4.1	Розрахунок витрат на створення та впровадження програмного забезпечення	43
4.2	Розрахунок економічної ефективності розробки та впровадження програмного забезпечення	46
5	ОХОРОНА ПРАЦІ	47
5.1	Аналіз небезпечних і шкідливих факторів у відділі розробки програмного забезпечення	48
5.2	Розрахунок освітленості у відділі розробки програмного забезпечення	50
5.3	Розробка заходів щодо зменшення впливу небезпечних і шкідливих факторів	52
6	ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	54
6.1	Вступ	54
6.2	Забруднення навколишнього середовища комп'ютерною компанією	56
6.3	Розробка заходів щодо зменшення забруднення.....	58
	ВИСНОВКИ.....	62
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	64
	ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ.....	66
	ДОДАТОК Б – ТЕКСТ ПРОГРАМИ.....	69
	ДОДАТОК В – ОПИС ПРОГРАМИ	79
	ДОДАТОК Г – ІНСТРУКЦІЯ КОРИСТУВАЧА	80
	ДОДАТОК Д – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ.....	82

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

ПЗ – програмне забезпечення;

СКВ – система керування вмістом;

KLOC – the thousand lines of code;

LOC – the lines of code;

CL – the number of classes;

MBC – the number of methods by class;

DIT – the depth of inheritance tree;

PRED – percentage of prediction;

ВСТУП

Актуальність теми. З розвитком мережі Інтернет стала дуже затребуваною розробка програмного забезпечення вебсайтів. Найрозповсюдженішими в наш час є вебсайти розроблені з використанням системи керування вмістом WordPress [1], однією із складових якої є тема. WordPress теми – це готові макети, що використовуються для ведення блогів, вебсайтів новин і проєктів для електронної комерції.

Оцінювання розміру програмного забезпечення – це важлива інформація на перших етапах розробки, тому що оцінювання розміру раніше розроблених проєктів використовується для оцінювання трудомісткості розроблення програмного забезпечення, та дозволяє правильно розподілити витрати й зусилля розробників [2]. Отже, оцінювання розміру тем для системи керування вмістом WordPress, яка є найпопулярнішою такою системою в світі [3], є важливою задачею програмної інженерії, вирішення якої потребує побудови відповідних моделей.

Для оцінювання розміру коду використовуються лінійні та нелінійні регресійні моделі [4]. Існуючі моделі, такі як трифакторна нелінійна регресійна модель для оцінювання розміру PHP застосунків із відкритим кодом [5], не призначені для оцінювання розміру WordPress тем, що призводить до низької достовірності відповідного оцінювання. Тому, розробка математичної моделі для оцінювання розміру WordPress тем на PHP є актуальною задачею, а розробка програмного забезпечення, що реалізує таку модель, має практичну цінність.

Мета і завдання дослідження. Метою роботи є підвищення достовірності оцінювання розміру WordPress тем на PHP.

Для досягнення поставленої мети потрібно вирішити наступні завдання:

- Проаналізувати існуючі моделі для оцінювання розміру програмного забезпечення на PHP;
- Аргументувати необхідність побудови моделі для оцінювання розміру WordPress тем;

- Побудувати нелінійну регресійну модель для оцінювання розміру WordPress тем;

- Розробити проєкт програмного забезпечення, що реалізує побудовану математичну модель для оцінювання розміру WordPress тем;

- Провести розрахунок економічної ефективності;

- Розглянути питання охорони праці;

- Розглянути питання охорони навколишнього середовища;

Об'єктом дослідження є процес оцінювання розміру WordPress тем на PHP.

Предметом дослідження є нелінійна регресійна модель для оцінювання розміру WordPress тем на PHP.

Методи дослідження. Для вирішення поставленого завдання застосовано методи теорії ймовірності, математичної статистики, регресійного аналізу.

Наукова новизна одержаних результатів полягає в удосконаленні трьохфакторної нелінійної регресійної моделі для оцінювання розміру PHP застосунків із відкритим кодом за рахунок знаходження параметрів моделі для WordPress тем на PHP, що дозволяє підвищити достовірність оцінювання їх розміру в порівнянні з існуючими трифакторними нелінійними регресійними моделями.

Практичне значення одержаних результатів. Розроблене програмне забезпечення для оцінювання розміру WordPress тем на PHP, що реалізує побудовану математичну модель.

Особистий внесок здобувача. Ця кваліфікаційна (магістерська) робота є самостійно виконаною працею. Всі результати що представлені в роботі, отримані автором особисто. В публікації [6], опублікованій у співавторстві з керівником кваліфікаційної роботи, здобувачеві належать: збір емпіричних даних про метрики тем для системи керування вмістом WordPress та побудова нелінійної моделі для оцінювання розміру WordPress тем на PHP.

Апробація результатів досліджень. Основні положення та результати досліджень, викладені у роботі, були оприлюднені на III Всеукраїнській науково–практичній інтернет – конференції «Інформаційні технології: моделі, алгоритми, системи – 2022» (Миколаїв, 26–28 жовтня 2022 року).

Публікації. Основні результати кваліфікаційної роботи викладені у 1 науковій праці – матеріалах інтернет – конференції «Інформаційні технології: моделі, алгоритми, системи – 2022».

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА МОДЕЛЕЙ ОЦІНЮВАННЯ РОЗМІРУ WORDPRESS ТЕМ НА PHP

1.1 WordPress як найрозповсюдженіша система керування вмістом вебсайтів, та її особливості

Система керування вмістом (Content management system, CMS, СКВ) вебсайтів надає користувачу можливість використовувати існуючий функціонал комп'ютерної програми, що забезпечує спрощений процес розробки, створення, редагування й керування контентом вебсайту. Різні системи керування вмістом надають можливість швидко й просто розгорнути вебсайт що має базовий функціонал, це дозволяє не витрачати багато часу на реалізацію стандартних функцій. Більшість сучасних систем керування вмістом вебсайтів мають свій API, плагіни та теми що дозволяють запросто розширити функціональність системи керування вмістом вебсайтів.

На даний момент близько 43% [3] з вебсайтів розміщених в мережі Інтернет працюють з використанням системи керування вмістом WordPress, а через простоту використання їх загальна кількість збільшується з кожною годиною. Багато відомих всесвітніх компаній використовують систему керування вмістом WordPress, наприклад Sony, The Wall Street Journal, Forbes, PlayStation, The New Yorker, Disney, Vogue. WordPress використовують навіть NASA й Білий Дім.

Саме тому можна стверджувати що WordPress – це найпопулярніша система керування вмістом вебсайтів, використовуючи яку функціонують різноманітні вебсайти, платформи, сервіси в мережі Інтернет.

WordPress користується великим попитом також серед професіональних розробників програмного забезпечення, бо має відкритий вихідний код та якісну, обширну документацію. Все це дає можливість розробникам програмного забезпечення без зайвих зусиль розширити й удосконалити функціонал розроблюваних проєктів. Система керування вмістом вебсайтів WordPress у своїй основі використовує HTML, CSS, PHP й JavaScript.

Легкість у використанні системи керування вмістом WordPress й наявність детальної документації робить саме цю систему вдалим вибором для користувачів будь якого рівня професіоналізму. Базовий функціонал WordPress дозволяє швидко створювати, редагувати чи видаляти вебсторінки; створювати, редагувати чи видаляти контент за допомогою текстового редактора що має два режими роботи: візуальний режим та HTML режим (для просунутих користувачів); додавати, редагувати чи видаляти зображення; налаштування відповідної URL адреси для вебсторінок; додавати, редагувати чи видаляти різноманітні медіафайли; додавати, редагувати чи видаляти категорії та теги, RSS стрічки та форми для введення даних чи пошукових запитів.

Велика кількість розроблених користувачами тем, віджетів та плагінів для системи керування вмістом WordPress дозволяють розширити наявний функціонал чи додати нові можливості, змінити зовнішній вигляд сторінок. Все це значно розширює базовий функціонал системи керування вмістом WordPress та спрощують процес розробки програмного забезпечення. Ідея користувацьких додатків до WordPress – залишити основу (ядро) незмінним, але змінити поведінку системи з допомогою розроблених користувачами скриптів-плагінів, що легко підключаються і відключаються.

Теми для системи керування вмістом WordPress – це набір файлів, сукупність яких відповідає за зовнішній вигляд сайту, готовий макет для системи управління контентом. Головною задачею теми є виведення вмісту на вебсайт у зручному для користувача вигляді. За допомогою тем розробник програмного забезпечення може вибрати відповідну та зручну структуру вебсайту та розмістити на вебсайті потрібну інформацію. Ці макети можна використовувати для початку розробки вебпроєкту чи модернізації вже існуючого проєкту. Теми легко поєднуються із системою управління вмістом вебсайтів WordPress. Теми дозволяють з легкістю розроблювати вебсайти з привабливим та зручним для користувачів дизайном, інтерфейсом. В офіційному каталозі тем для системи керування вмістом WordPress більше ніж 10200 безплатних тем, ще більш широким є вибір платних тем. Знаючи такі технології як HTML, CSS, SCSS та PHP розробник має можливість

модифікувати готову тему під свої потреби змінивши зовнішній вигляд та функціонал.

1.2 Аналіз існуючих методів та моделей для оцінювання розміру програмного забезпечення

Метрика програмного забезпечення — це міра, що дозволяє отримати числове значення деяких властивостей програмного забезпечення та його специфікацій [7]. Для кожної з метрик зазвичай існують еталонні показники, що вказують граничні значення, при виході за які варто звернути увагу. На основі метрик, проєкт програмного забезпечення можна порівняти з іншими проєктами, щоб провести аналіз та зробити висновки щодо ходу процесу розробки, вартості, складності тощо.

Важливе значення на початковому етапі розробки програмного забезпечення має оцінювання такої метрики, як розмір програмного забезпечення – це покращує планування та управління процесом розробки.

На сьогодні для розробників існує велика кількість алгоритмічних моделей за допомогою яких можна оцінювати розмір програмного забезпечення для подальшої побудови графіку розробки. Ці моделі відрізняються між собою тим, що використовують різні параметри для оцінювання. Загалом моделі для оцінювання розміру програмного забезпечення поділяють на такі види: регресійні моделі, моделі на основі експертних оцінок, аналогові моделі, моделі базовані на функціональних точках та параметричні моделі.

Регресійні моделі оцінювання розміру ПЗ поділяють на лінійні та нелінійні. Лінійні моделі – це моделі, в яких використовуючи емпіричні дані, методом найменших квадратів знаходять коефіцієнти рівняння регресії. За допомогою отриманого рівняння оцінюється значення розміру програмного забезпечення.

При побудові нелінійних регресійних моделей використовують дві методики. Перша – методом перебору знаходиться нелінійна функція регресії, а потім на її основі будується нелінійна регресійна модель. Друга – знаходяться перетворення

випадкових змінних регресії (нормалізуючі перетворення), що потім дозволять отримати близький до лінійного зв'язок між вже нормалізованими змінними. При побудові нелінійних регресійних моделей, у якості нормалізуючого перетворення найчастіше використовується десятковий логарифм. Саме тому побудова нелінійної регресійної моделі для оцінювання розміру WordPress тем на PHP, буде проводитись на основі нормалізуючого перетворення у вигляді десятичного логарифма.

1.3 Обґрунтування необхідності проведення досліджень за обраною темою

Розмір коду є значущим показником під час розробки програмного забезпечення. Ця інформація використовується на перших стадіях розробки для оцінювання трудомісткості та визначення ресурсів проєкту. Розвиток інтернет-технологій, поява нових вебзастосунків та додатків до них, підвищення вимог до якості проєктів, нові мови програмування, конкуренція – все це підвищує вимоги до точності та швидкості оцінювання програмного забезпечення.

Аналіз існуючих моделей для оцінювання розміру програмного забезпечення дає можливість зробити висновки, що вони не дозволяють з достатньою достовірністю оцінити розмір WordPress тем на PHP. Існуючі моделі, такі як трифакторна нелінійна регресійна модель для оцінювання розміру PHP застосунків із відкритим кодом [5], не призначені для оцінювання розміру WordPress тем, що призводить до низької достовірності відповідного оцінювання. Тому виникає необхідність побудови удосконаленої математичної моделі. Побудова математичної моделі для оцінювання розміру WordPress тем на PHP є актуальною задачею, а розробка програмного забезпечення, що реалізує таку модель, має практичну цінність.

2 ПОБУДОВА НЕЛІНІЙНОЇ РЕГРЕСІЙНОЇ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WORDPRESS ТЕМ НА PHP

2.1 Аналіз підходу до побудови нелінійної регресійної моделі та вибір факторів для оцінювання розміру WordPress тем на PHP

В попередньому розділі розглянуто існуючі моделі для оцінювання розміру програмного забезпечення на PHP, в результаті виявлено необхідність удосконалення нелінійної регресійної моделі [5] для оцінювання розміру WordPress тем на PHP. Для побудови нелінійної регресійної моделі використовується наступний алгоритм:

- дослідити WordPress теми з відкритим вихідним кодом, провести збір емпіричних даних;
- виконати нормалізацію отриманих емпіричних даних, за допомогою використання перетворення у вигляді десяткового логарифма;
- провести перевірку даних на викиди за допомогою квадрата відстані Махаланобіса;
- побудувати лінійну регресійну модель та провести перевірку її адекватності;
- побудувати нелінійну регресійну модель.

Побудова нелінійної регресійної моделі для оцінювання розміру WordPress тем є актуальною задачею для покращення достовірності оцінювання програмного забезпечення. Щоб оцінити розмір WordPress тем були обрані такі емпіричні дані: розмір WordPress теми у тисячах рядків коду, кількість класів, середня кількість методів на клас та глибина дерева успадкування.

2.2 Перевірка емпіричних даних для оцінювання розміру WordPress тем на PHP, на наявність викидів

Для побудови нелінійної регресійної моделі для оцінювання розміру WordPress тем на PHP, були зібрані емпіричні значення метрик 51 WordPress теми. Емпіричні дані включають такі метрики:

- кількість тисяч рядків коду (LOC);
- кількість класів (CL);
- середня кількість методів на клас (MBC);
- глибина дерева успадкування (DIT).

Значення вказаних метрик були отримані за допомогою утиліти для аналізу програмного коду на PHP – PhpMetrics [8] та представлені у таблиці 2.1.

Таблиця 2.1 – Значення метрик WordPress тем

№ з/п	Назва теми	LOC	CL	MBC	DIT
1	Agnar	18,633	137	4,44	1,31
2	Dumbo	7,282	26	7,65	1,47
3	MagazineX	6,474	56	3,88	1,25
4	Multi Store	20,330	152	4,88	1,28
5	News Hub	6,061	18	10,06	1,46
6	OceanWP	24,204	73	8,29	1,23
7	RuRu	8,526	28	7,32	1,5
8	Tattoized	7,973	23	13,52	1,28
9	Blocksy	4,658	29	8,03	1,27
10	GeneratePress	2,584	23	5	1,42
11	Hestia	18,300	107	6,58	1,39
12	Inspiro	6,613	52	4,4	1,14
13	Kadence	21,739	82	8,16	1,05
14	Neve	23,858	166	6,33	1,23
15	Storefront	3,751	15	7,8	1,17
16	Mesmerize	13,937	134	3,8	1,4
17	Phlox	10,747	67	5,58	1,52

18	Rife Free	5,213	22	7,64	1,53
19	Chic Lite	5,248	21	8,43	1,43
20	Bam	4,061	25	4,2	1,2
21	TutorStarter	5,161	37	5	1,12
22	Virtue	16,789	97	5,21	1,18
23	Ascendoor Magazine	6,480	18	9,94	1,5
24	Best Shop	4,010	16	7,5	1,5
25	Elsie	4,608	27	5,59	1,13
26	HitMag	13,970	139	3,32	1,29
27	Burgerlo	6,872	59	3,9	1,22
28	Maglist	9,625	33	10,21	1,58
29	MasterStroke	8,801	30	6,97	1,5
30	Namaha	3,836	30	4,27	1,42
31	NewsCard	4,992	15	10,13	1,5
32	One Page Express	13,592	132	3,72	1,39
33	Orchid Store	7,667	23	8,52	1,46
34	Responsive	6,425	72	3,19	1,13
35	ScapeShot	4,105	16	8,69	1,46
36	Superb eCommerce	5,312	23	7,74	1,22
37	Aasta	2,535	29	3,24	1,67
38	Cream Magazine	9,458	53	5,72	1,17
39	WPKites	4,598	48	3,38	1,62
40	MoreNews	6,730	27	7,37	1,5
41	NewsCut	7,972	28	7,32	1,5
42	OliveWP	4,648	47	3,74	1,54
43	Personal CV Resume	4,779	16	10,25	1,31
44	Saya	2,747	12	7,25	1,43
45	ShoppingCart	4,274	13	10,38	1,5
46	Sparkling	3,352	19	6,63	1,33
47	Veritable	8,525	28	7,32	1,5
48	Vilva	5,002	19	8,68	1,46
49	WP Newspaper	3,671	15	7,93	1,64
50	Construction Light	7,490	31	5,55	1,57
51	Colibri WP	9,110	101	5,76	1,45

Метрики CL, MBC, DIT не мають лінійної залежності між декількома незалежними змінними у регресійній моделі, тому ці дані не мають проблем із явищем мультиколінеарності.

Щоб побудувати нелінійну регресійну модель для оцінювання розміру WordPress тем на PHP, потрібно провести нормалізацію даних з таблиці 2.1 за допомогою одновимірного перетворення у вигляді десяткового логарифма:

$$z = \lg x. \quad (2.1)$$

Нормалізовані дані перевіряються на наявність викидів. Якщо викиди виявлено, вони видаляються та виконується повторна перевірка доки викидів не залишиться. Для визначення викидів використовується квадрат відстані Махаланобіса для нормалізованих даних.

Значення квадрата відстані Махаланобіса d_i^2 для кожної багатовимірної точки даних $i = 1, 2, \dots, n$, визначається як:

$$d_i^2 = (Z_i - \bar{Z})^T S_N^{-1} (Z_i - \bar{Z}), \quad (2.2)$$

де $\bar{Z}$ – вектор вибірових середніх;

S_N – вибірова коваріаційна матриця, $S_N = \frac{1}{N} \sum_{i=1}^N (X_i - \bar{Z})(X_i - \bar{Z})^T$;

Z_i – випадковий вектор $Z = \{Z_1, Z_2, \dots, Z_m\}^T$ для кожної багатовимірної точки даних $i = 1, 2, \dots, m$;

m – кількість компонентів вектора Z .

В нашому випадку m дорівнює 4.

У таблиці 2.2 представлені нормалізовані дані та значення квадрата відстані Махаланобіса d_i^2 .

Таблиця 2.2 – Нормалізовані значення метрик WordPress тем та значення квадрата відстані Махаланобіса d_i^2

№ з/П	Назва теми	LOC	CL	MBC	DIT	d_i^2
1	Agnar	1,27	2,14	0,65	0,12	3,62
2	Dumbo	0,86	1,41	0,88	0,17	1,35
3	MagazineX	0,81	1,75	0,59	0,10	2,23
4	Multi Store	1,31	2,18	0,69	0,11	4,42
5	News Hub	0,78	1,26	1,00	0,16	1,80
6	OceanWP	1,38	1,86	0,92	0,09	7,31
7	RuRu	0,93	1,45	0,86	0,18	3,65
8	Tattoized	0,90	1,36	1,13	0,11	7,73
9	Blocksy	0,67	1,46	0,90	0,10	9,09
10	GeneratePress	0,41	1,36	0,70	0,15	6,25
11	Hestia	1,26	2,03	0,82	0,14	5,45
12	Inspiro	0,82	1,72	0,64	0,06	3,35
13	Kadence	1,34	1,91	0,91	0,02	8,44
14	Neve	1,38	2,22	0,80	0,09	9,41
15	Storefront	0,57	1,18	0,89	0,07	5,29
16	Mesmerize	1,14	2,13	0,58	0,15	4,38
17	Phlox	1,03	1,83	0,75	0,18	2,78
18	Rife Free	0,72	1,34	0,88	0,18	1,16
19	Chic Lite	0,72	1,32	0,93	0,16	1,03
20	Bam	0,61	1,40	0,62	0,08	8,41
21	TutorStarter	0,71	1,57	0,70	0,05	4,27
22	Virtue	1,23	1,99	0,72	0,07	3,43
23	Ascendoor Magazine	0,81	1,26	1,00	0,18	2,31
24	Best Shop	0,60	1,20	0,88	0,18	1,45
25	Elsie	0,66	1,43	0,75	0,05	4,59
26	HitMag	1,15	2,14	0,52	0,11	4,22
27	Burgerlo	0,84	1,77	0,59	0,09	2,48
28	Maglist	0,98	1,52	1,01	0,20	5,18
29	MasterStroke	0,94	1,48	0,84	0,18	3,97
30	Namaha	0,58	1,48	0,63	0,15	2,33
31	NewsCard	0,70	1,18	1,01	0,18	2,14
32	One Page Express	1,13	2,12	0,57	0,14	4,17
33	Orchid Store	0,88	1,36	0,93	0,16	2,46
34	Responsive	0,81	1,86	0,50	0,05	5,70
35	ScapeShot	0,61	1,20	0,94	0,16	1,71
36	Superb eCommerce	0,73	1,36	0,89	0,09	2,19

37	Aasta	0,40	1,46	0,51	0,22	9,56
38	Cream Magazine	0,98	1,72	0,76	0,07	1,96
39	WPKites	0,66	1,68	0,53	0,21	6,34
40	MoreNews	0,83	1,43	0,87	0,18	0,87
41	NewsCut	0,90	1,45	0,86	0,18	2,36
42	OliveWP	0,67	1,67	0,57	0,19	4,55
43	Personal CV Resume	0,68	1,20	1,01	0,12	2,98
44	Saya	0,44	1,08	0,86	0,16	2,68
45	ShoppingCart	0,63	1,11	1,02	0,18	2,58
46	Sparkling	0,53	1,28	0,82	0,12	2,49
47	Veritable	0,93	1,45	0,86	0,18	3,65
48	Vilva	0,70	1,28	0,94	0,16	1,21
49	WP Newspaper	0,56	1,18	0,90	0,21	3,08
50	Construction Light	0,87	1,49	0,74	0,20	5,97
51	Colibri WP	0,96	2,00	0,76	0,16	14,97

Згідно таблиці критичне значення $d^2 = 14,86$ (4 ступені свободи, рівень значущості $\alpha = 0,005$). Лише одна тема (Colibri WP) має значення d^2 що перевищує критичне значення. Виключивши цю тему з набору метрик можна переходити до побудови нелінійної регресійної моделі.

2.3 Побудова нелінійної регресійної моделі для оцінювання розміру WordPress тем на PHP

Щоб побудувати нелінійну регресійну модель для оцінювання розміру WordPress тем на PHP, було проведено нормалізацію даних за допомогою одновимірного перетворення у вигляді десяткового логарифма (2.1). Після видалення викидів для нормалізованих даних будується лінійна регресійна модель:

$$Z_Y = \hat{Z}_Y + \varepsilon = \hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \hat{b}_3 Z_3 + \varepsilon \quad (2.3)$$

де $Z_Y = \lg Y$;

$Z_j = \lg X_j$;

$j = 1, 2, 3$.

Параметри моделі (2.3) оцінюються методом найменших квадратів. Значення отриманих оцінок параметрів: $\hat{b}_0 = -1.54$; $\hat{b}_1 = 0.97$; $\hat{b}_2 = 1.05$; $\hat{b}_3 = 0.19$.

Для перевірки адекватності лінійної регресійної моделі потрібно провести перевірку нормальності закону розподілу випадкової величини ε . У разі великої вибірки значень випадкової величини (коли $n > 30$) перевірку нормальності закону розподілу результатів спостережень, у тому числі, виконують за критерієм Пірсона (критерієм χ^2) [9]. Відповідно за цим критерієм спочатку обчислюють значення χ^2 :

$$\chi^2 = \sum_{j=1}^m \frac{(n_j - np_j)^2}{np_j},$$

де m – кількість підінтервалів, на які розбивається інтервал $[\chi_{min}, \chi_{max}]$; χ_{min} і χ_{max} – відповідно мінімальне і максимальне значення випадкової величини X ;

n_j – абсолютна частота в j -му підінтервалі (кількість значень випадкової величини, які попадають у j -ий підінтервал);

p_j – ймовірність того, що значення випадкової величини X попадають у j -ий підінтервал.

Кількість підінтервалів, на які розбивається інтервал $[\chi_{min}, \chi_{max}]$ може бути визначена за наступними формулами: $m = \log_2 n + 1 = 3,3 \lg n + 1$ (формула Старджеса) або $m = 5 \lg n$ (формула Брукса і Карузера) [9].

Ймовірність того, що значення випадкової величини X попадають у j -ий підінтервал, може бути визначена як:

$$p_j = \int_{x_{j-1}}^{x_j} f(x) dx,$$

де x_{j-1} і x_j – ліва і права границі j -го підінтервалу.

Після цього в залежності від рівня значимості α та кількості ступенів вільності ν , за таблицею верхніх $100\alpha\%$ -вих точок розподілу χ^2 знаходять значення $\chi_{кр}$.

Якщо, $\chi^2 \leq \chi_{кр}^2$ то з ймовірністю $1 - \alpha$ можна прийняти гіпотезу про те, що закон розподілу результатів спостережень є нормальним, якщо $\chi^2 > \chi_{кр}^2$ – цю гіпотезу потрібно відкинути.

Кількість ступенів вільності визначається як

$$v = m - k - 1,$$

де k – кількість параметрів, від яких залежить закон розподілу. Для нормального розподілу $k = 2$.

У нашому випадку $\chi_{кр} = 9,49$, $\chi^2 = 6,07$. Так як $\chi^2 \leq \chi_{кр}^2$, то закон розподілу випадкової величини ε є нормальним.

Побудована гістограма розподілу випадкової величини ε зображена на рисунку 2.1.

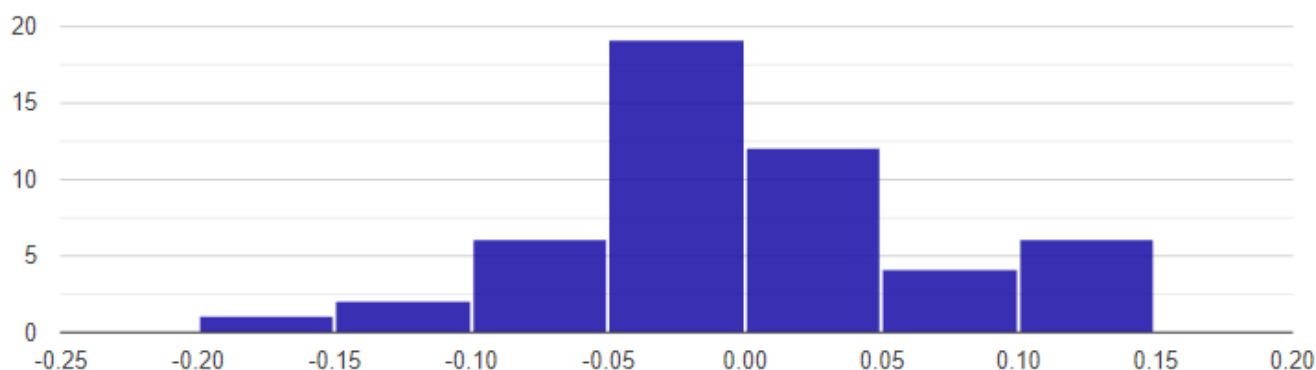


Рисунок 2.1 – Гістограма розподілу випадкової величини ε

В результаті будується нелінійна регресійна модель за лінійним рівнянням та перетворенням у вигляді десяткового логарифма.

$$Y = 10^{\varepsilon + \hat{b}_0} X_1^{\hat{b}_1} X_2^{\hat{b}_2} X_3^{\hat{b}_3} \quad (2.4)$$

Рівняння (2.4) – отримана нелінійна регресійна модель для оцінювання розміру WordPress тем на PHP.

Визначаємо границі інтервалу передбачення нелінійної регресії для рівня значущості, що дорівнює 0,05. Границі інтервалу передбачення нелінійної регресії визначаються як:

$$\Psi_Y^{-1} \left(\hat{Z}_Y \pm t_{\alpha/2, v} S_{Z_Y} \left\{ 1 + \frac{1}{N} + (Z_X^+)^T [(Z_X^+)^T Z_X^+]^{-1} (Z_X^+) \right\}^{1/2} \right),$$

де Ψ_Y^{-1} – зворотне до нормалізуючого перетворення Ψ_Y для залежної змінної Y ;

$\widehat{Z_Y}$ – результат передбачення лінійного регресійного рівняння для значень компонента вектора $z_X = \{Z_1, Z_2, \dots, Z_k\}$, що отриманий за нормалізованими даними;

k – кількість незалежних змінних (у нашому випадку $k = 3$);

Z_X^+ – матриця центрованих нормалізованих незалежних змінних, що містить значення $Z_{1i} - \bar{Z}_1, Z_{2i} - \bar{Z}_2, \dots, Z_{ki} - \bar{Z}_k$;

$t_{\alpha/2, v}$ – квантиль t -розподілу Стюдента в залежності від рівня значимості $\alpha/2$ та кількості ступенів свободи v ;

$S_{Z_Y}^2 = \frac{1}{v} \sum_{i=1}^N (Z_{Y_i} - \hat{Z}_{Y_i})^2, v = N - k - 1$; $(Z_X^+)^T Z_X^+$ – $k \times k$ матриця

$$(Z_X^+)^T Z_X^+ = \begin{pmatrix} S_{Z_1 Z_1} & S_{Z_1 Z_2} & \dots & S_{Z_1 Z_k} \\ S_{Z_1 Z_2} & S_{Z_2 Z_2} & \dots & S_{Z_2 Z_k} \\ \vdots & \vdots & \ddots & \vdots \\ S_{Z_1 Z_k} & S_{Z_2 Z_k} & \dots & S_{Z_k Z_k} \end{pmatrix}, \quad (2.4)$$

де $S_{Z_q Z_r} = \sum_{i=1}^N [Z_{qi} - \bar{Z}_q][Z_{ri} - \bar{Z}_r]$,

$q, r = 1, 2, \dots, k$.

У формулі (2.4) знаки мінус та плюс відповідають нижній та верхній границям інтервалу передбачення нелінійної регресії, яка побудована на основі нормалізуючого перетворення.

Для емпіричних даних метрик WordPress тем на PHP матриця $(Z_X^+)^T Z_X^+$ є наступною:

$$(Z_X^+)^T Z_X^+ = \begin{pmatrix} 5,082 & -1,518 & -0,301 \\ -1,518 & 1,266 & 0,072 \\ -0,301 & 0,072 & 0,126 \end{pmatrix}.$$

Ця нелінійна регресійна модель перевірена за множинним коефіцієнтом детермінації R^2 , середньою величиною відносної помилки MMRE і відсотком прогнозованих результатів, для яких величини відносної помилки MRE менші за 0,25, $PRED(0,25)$.

Середня величина відносної похибки MMRE визначається як:

$$MMRE = \frac{1}{N} \sum_{i=1}^N MRE_i,$$

Значення величини відносної похибки MRE обчислюється як:

$$MRE_i = \left| \frac{(Y_i - \hat{Y}_i)}{Y_i} \right|,$$

Зазвичай $MMRE \leq 0,25$ вважається прийнятною точністю моделі. Значення $PRED(0,25)$ обчислюється як:

$$PRED(0,25) = \frac{1}{N} \sum_{i=1}^N \begin{cases} 1 & \text{if } MRE_i < 0,25 \\ 0 & \text{otherwise.} \end{cases},$$

Зазвичай $PRED(0,25) \geq 0,75$ вважається прийнятною точністю оцінювання за допомогою регресійних моделей

Ці показники переважно використовуються для оцінювання якості прогнозування за допомогою регресійних моделей. Допустимі значення R^2 , $MMRE$ і $PRED(0,25)$ складають не менше 0,7, не більше 0,25 і не менше 0,75 відповідно.

Для моделі, що була побудована за даними метрик 50 WordPress тем на PHP з таблиці 2.2 значення R^2 , $MMRE$ і $PRED(0,25)$ дорівнюють 0,911; 0,117; 0,9 відповідно та вказують на високу достовірність побудованої нелінійної регресійної моделі.

У порівнянні з трифакторною нелінійною регресійною моделлю для оцінювання розміру PHP застосунків із відкритим кодом [5], у якої значення R^2 , $MMRE$ і $PRED(0,25)$ для емпіричних даних метрик WordPress тем на PHP дорівнюють 0,439, 0,419, 0,06 відповідно, побудована модель має більшу достовірність. Значення R^2 покращилось на 107,52%, $MMRE$ покращилось на 72,08%, $PRED(0,25)$ покращилось на 93,33%.

Результати досліджень проведених в цьому розділі опубліковано автором в публікації [6].

3 РОЗРОБКА ПРОЄКТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WORDPRESS ТЕМ НА PHP

3.1 Ескізний проєкт програмного забезпечення для оцінювання розміру WordPress тем на PHP

В ескізному проєкті, з ціллю визначити необхідні функції програмного забезпечення, за допомогою мови моделювання UML було побудовано діаграму варіантів використання та сценарії використання, розроблено зовнішню специфікацію програми для оцінювання розміру WordPress тем на PHP.

Щоб детально представити варіанти використання побудуємо діаграму діяльності для варіантів використання, проєкт інтерфейсу користувача.

Контекстну діаграму програмного забезпечення для оцінювання розміру WordPress тем на PHP представлено на рисунку 3.1.



Рисунок 3.1 – Контекстна діаграма розробленого програмного забезпечення

Перелік та опис потоків даних, що відповідають побудованій діаграмі представлено у таблиці 3.1

Таблиця 3.1 – Опис потоків даних програмного забезпечення

Назва потоку	Опис потоку
Дані метрик ПЗ	Назва теми; Класи; Методи на клас; Глибина дерева успадкування
Результати розрахунків	Кількість строк коду; Довірчий інтервал нижній; Довірчий інтервал верхній; Інтервал передбачення нижній; Інтервал передбачення верхній

Цей проєкт передбачає наявність лише одного актора, а саме – користувача, який буде використовувати всі функції розробленого програмного забезпечення. Актори розробленого програмного забезпечення представлені в таблиці 3.2

Таблиця 3.2 – Актори програмного забезпечення

Актор	Опис актора
Користувач	Людина, яка використовує програмне забезпечення, а саме вводить дані, щоб виконати обчислення розміру WordPress тем на PHP, вирахувати довірчий інтервал та інтервал передбачення нелінійної регресії

В таблиці 3.3 представлений опис виявлених варіантів використання програмного забезпечення.

Таблиця 3.3 – Опис виявлених варіантів використання програмного забезпечення

Актор	Назва варіанту використання	Опис варіанту використання
Користувач	Введення назви теми та її метрик	Користувач вводить назву тем та її метрики у відповідні поля програми
Користувач	Отримання розміру WordPress теми на PHP	Користувач отримує розмір WordPress теми на PHP у тисячах рядків коду

Користувач	Отримання довірчого інтервалу нелінійної регресії	Користувач отримує довірчий інтервал нелінійної регресії для інтервального оцінювання вибіркового середнього розміру WordPress тем на PHP
Користувач	Отримання інтервалу передбачення нелінійної регресії	Користувач отримує інтервал передбачення нелінійної регресії для інтервального оцінювання вибіркового середнього розміру WordPress тем на PHP
Користувач	Збереження отриманих результатів в базу даних	Користувач зберігає отримані дані в базу даних
Користувач	Перегляд даних збережених в базу даних	Користувач переглядає сторінку де відображаються дані що збережені в базу даних

На рисунку 3.2 зображено діаграму варіантів використання розробленого програмного забезпечення.

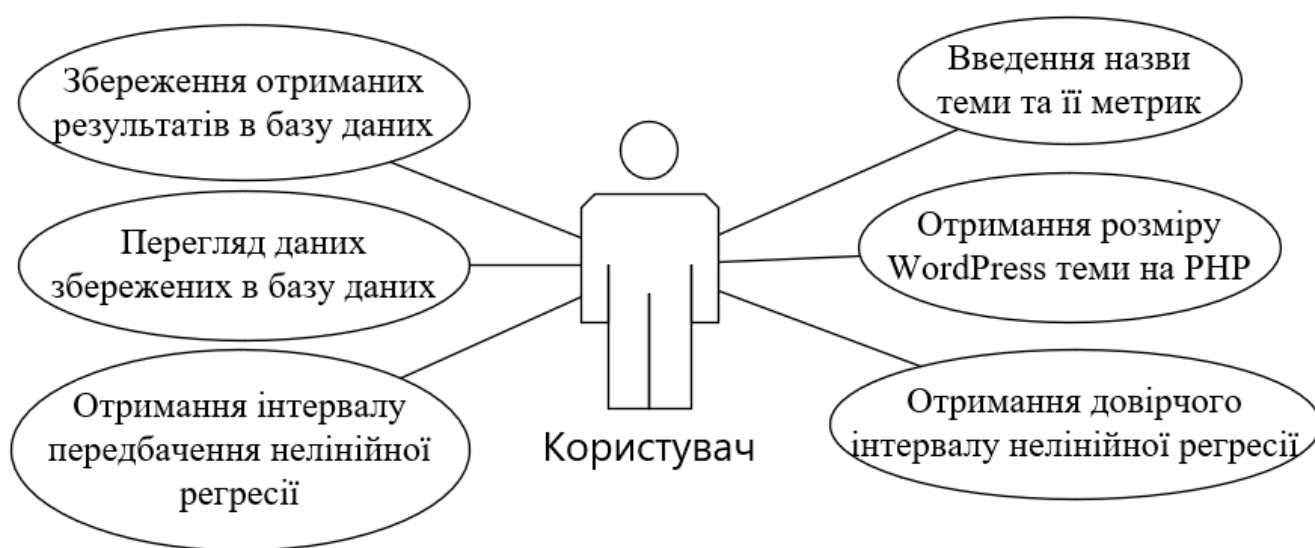


Рисунок 3.2 – Діаграма варіантів використання розробленого програмного забезпечення

Далі доцільно розробити діаграму діяльності для розробленого програмного забезпечення. На рисунку 3.3 зображено діаграму діяльності розробленого програмного забезпечення.

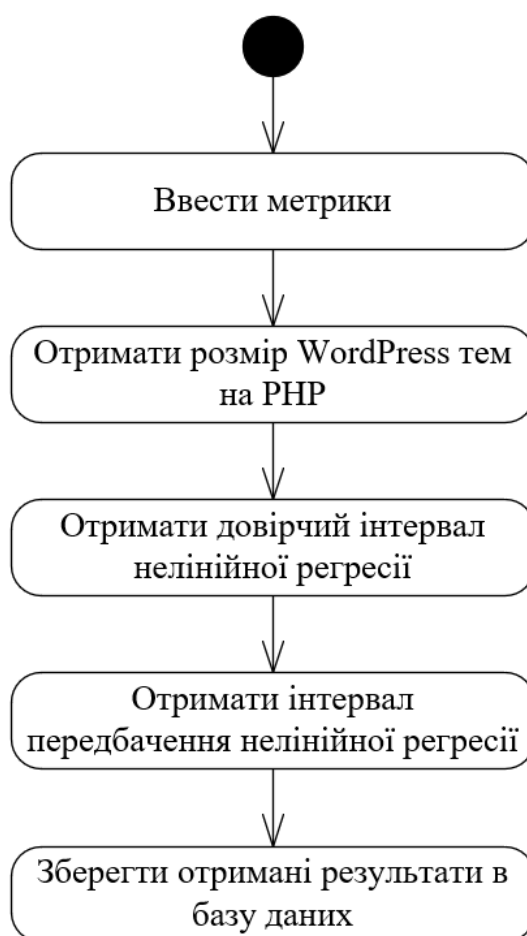


Рисунок 3.3 – Діаграма діяльності розробленого програмного забезпечення

Інтерфейс – це зовнішня оболонка програмного забезпечення, що являє собою систему засобів і правил та забезпечує зручну взаємодію користувача з програмним забезпеченням. У програмному забезпеченні для оцінювання розміру WordPress тем на PHP були використані такі інтерфейсні компоненти як: рядки редагування, рядки для виводу тексту. Проєкт інтерфейсу програмного забезпечення зображено на рисунку 3.4.

Enter theme name

Theme Name

Enter the number of Classes

0.00

Enter the number of Methods by Class

0.00

Enter the number of Depth of Inheritance Tree

0.00

CALCULATE PREVIOUS RESULTS

Lines of Code: Please enter a value

Confidence Interval: Please enter a value

Prediction Interval: Please enter a value

Рисунок 3.4 – Проєкт інтерфейсу програмного забезпечення

3.2 Технічний проєкт програмного забезпечення для оцінювання розміру WordPress тем на PHP

Логічна модель даних — модель даних предметної області, яка виражена незалежно від конкретного продукту управління базами даних чи технологій зберігання.

Логічна модель даних програмного забезпечення розроблялася згідно з технічним завданням та зображена на рисунку 3.5.

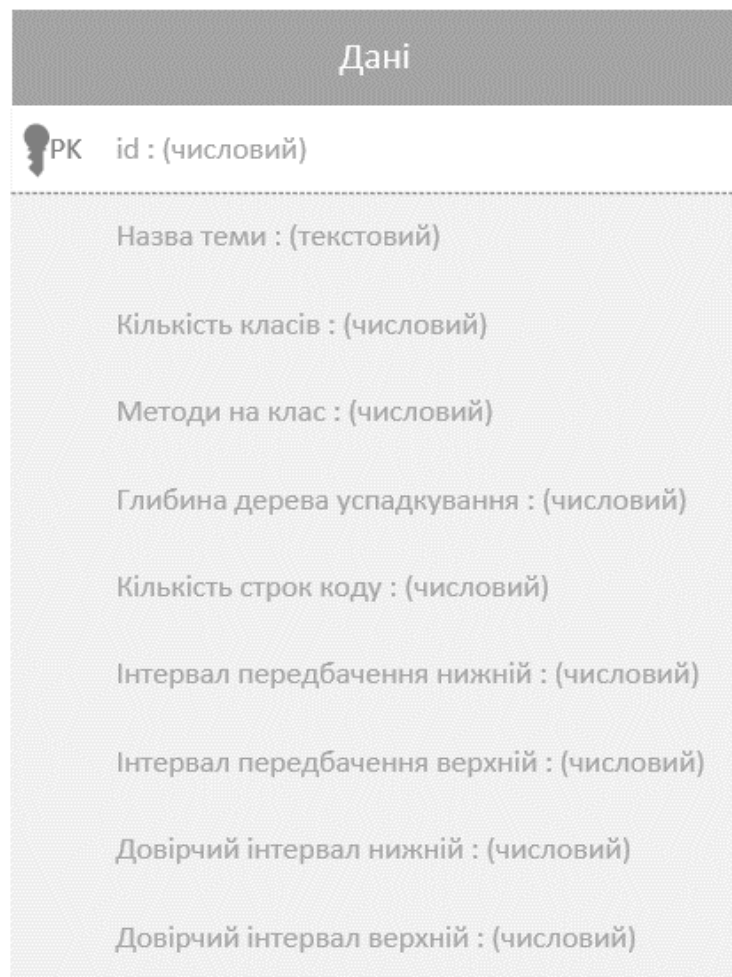


Рисунок 3.5 – Логічна модель даних програмного забезпечення

3.3 Робочий проєкт програмного забезпечення для оцінювання розміру WordPress тем на PHP

3.3.1 Обґрунтування вибору мови програмування, СКБД та середовища розробки

Для розробки програмного забезпечення обрано мову програмування PHP та СКБД MySQL, інтерфейс буде створено за допомогою мови розмітки HTML, мови стилю сторінок CSS. Для розробки програмного забезпечення обрано редактор коду Visual Studio Code — він безкоштовний та має увесь потрібний для вирішення задачі функціонал. Далі стисло описано кожен з використаних мов, СКБД та редактор коду.

PHP — скриптова мова програмування, яка була створена для генерації HTML-сторінок на стороні вебсервера. PHP зараз це одна з найрозповсюдженіших мов програмування, що використовуються для веброзробки. Більшістю хостинг-провайдерів мають підтримку PHP. PHP інтерпретується вебсервером в HTML-код, що в свою чергу передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач безпосередньо не бачить PHP-код, браузер отримує вже готовий HTML-код. Це велика перевага для безпеки вебсайту, але в свою чергу погіршує інтерактивність сторінок. Для обходу цього обмеження можна використовувати PHP для генерування JavaScript-коду, що буде виконуватися вже на стороні клієнта [10].

HTML — це мова тегів, засобами якої здійснюється розмічання вебсторінок для мережі Інтернет. Браузер отримує HTML-документ з вебсервера чи безпосередньо з локальної пам'яті та інтерпретує цей документ в мультимедійну вебсторінку. HTML описує структуру вебсторінки семантично, а елементи HTML це блоки для побудови вебсторінок. З використанням конструкцій HTML, зображення чи інші об'єкти можуть бути вбудовані у візуалізовану вебсторінку. HTML має засоби для створення структурованих документів, позначивши структурну семантику тексту, таку як заголовки, посилання, списки, абзаци, цитати й інші елементи HTML. HTML дає можливість вбудовувати в код програми

скрипти, написані мовою скриптів, наприклад JavaScript, які змінюють чи впливають на поведінку й вміст вебсторінок. Включення CSS в свою чергу задає чи змінює зовнішній вигляд й компоновання вмісту вебсторінки [11].

CSS — це мова стилю сторінок, яка використовується для опису зовнішнього вигляду цих сторінок. Сторінки в свою чергу написані з використанням мови розмітки HTML. CSS це одна з основних технологій всесвітньої павутини, поряд із HTML та JavaScript й PHP. Зазвичай CSS використовується для візуальної презентації сторінок, код яких написаний з використанням HTML/XHTML, також CSS можна використовувати й разом з іншими видами XML-документів. В сучасному CSS каскадна/блочна верстка прийшла на заміну табличній верстці вебсторінок. Головною перевагою блочної верстки є розподілення змісту вебсторінки (даних) від їх візуальної презентації [12].

MySQL — вільна система керування реляційними базами даних, що була розроблена компанією «ТсХ», щоб покращити швидкість обробки великих баз даних. Ця система керування базами даних (СКБД) має відкритий код та була створена як альтернатива комерційним системам. Зараз MySQL — одна з найпоширеніших систем керування базами даних. Використовується, в першу чергу, для створення динамічних вебсторінок, через те що має підтримку зі сторони різних мов програмування [13].

Visual Studio Code — інструмент для створення, редагування й зневадження сучасних вебзастосунків та програмного забезпечення для хмарних систем. Visual Studio Code є безкоштовною та доступна у версіях для таких платформ, як: Windows, Mac OS, Linux. Компанія Microsoft презентувала Visual Studio Code у квітні 2015 року. Це середовище розробки перший кросплатформовий продукт у лінійці продуктів Visual Studio. Редактор коду має вбудований зневаджувач, автодоповнення типових конструкцій й контекстні підказки, набір інструментів для роботи з Git та засоби рефакторингу, зручної навігації по коду. Продукт підтримує розробку для платформ ASP.NET й Node.js, позиціонується як спрощене рішення, яке дозволяє обійтися без повного інтегрованого середовища розробки. Підтримувані мови та технології: HTML, CSS, LESS і SASS, JavaScript, TypeScript,

JSON, C++, C#, Objective-C, jade, PowerShell, PHP, Java, Python, XML, Batch, F#, Visual Basic, DockerFile, Coffee Script, HandleBars, R, Luna, Markdown, Haxe [14].

3.3.2 Фізична модель бази даних програмного забезпечення для оцінювання розміру WordPress тем на PHP

Фізична модель бази даних програмного забезпечення зображена на рисунку 3.6. Структура таблиці бази даних описана в таблиці 3.4.

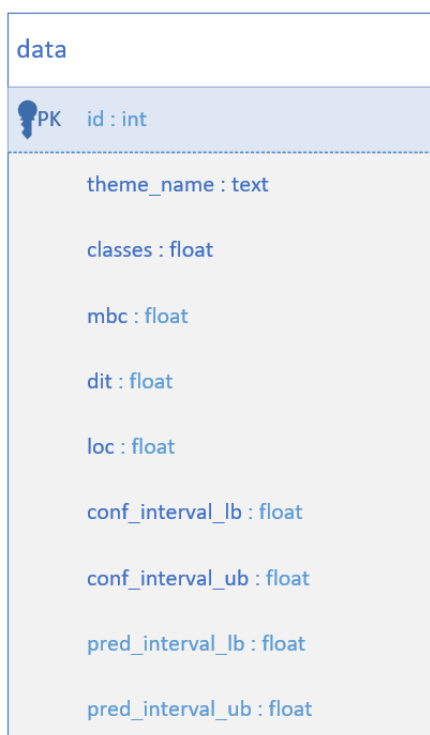


Рисунок 3.6 – Фізична модель даних програмного забезпечення

Таблиця 3.4 – Опис полів таблиці бази даних програмного забезпечення

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Ідентифікатор	id	text	РК
Назва теми	theme_name	float	
Класи	classes	float	
Методи на клас	mbc	float	
Глибина дерева успадкування	dit	float	
Кількість строк коду	loc	float	
Довірчий інтервал нижній	conf_interval_lb	float	

Довірчий інтервал верхній	conf_interval_ub	float	
Інтервал передбачення нижній	pred_interval_lb	float	
Інтервал передбачення верхній	pred_interval_ub	float	

3.3.3 Тестування програмного забезпечення для оцінювання розміру WordPress тем на РНР

Потрібно провести тестування основних варіантів використання. Тестування будемо проводити з варіантами використання: «Отримання розміру WordPress теми на РНР», «Збереження отриманих результатів в базу даних», «Перегляд даних збережених в базу даних».

Тестування варіанта використання «Отримання розміру WordPress теми на РНР»

Вхідні дані: слова та числа.

Слово – набір символів латиницею або кирилицею.

Перше поле «Назва теми» приймає слова, числа та спецсимволи. Інші поля приймають лише числа.

При введенні дробових чисел у відповідні поля форми ціла частина повинна відділятися комою від дробової.

При недопустимих значеннях даних на екран виводиться повідомлення в про помилку:

– поля форми залишилось пустим – повідомлення про обов’язковість заповнення кожного поля форми з текстом «Enter numeric data»;

– поля форми заповнені не числовими символами – повідомлення про обов’язковість заповнення кожного поля форми з текстом «Enter numeric data».

Якщо поля форми заповнені вірно, результат розрахунків виводиться на екран та зберігається в базу даних.

Тестові вимоги:

– перевірити, чи буде відбуватися перевірка, якщо поля форми залишилось пустим;

– перевірити, чи буде відбуватися перевірка, якщо поля форми заповнені не числовими символами;

Розробка випадків для тестових вимог

Перевірити, чи буде відбуватися перевірка введених даних у відповідні поля форми

– клас еквівалентності: поле введення пароля залишилось пустим

Вхідні дані: «».

Результат: повідомлення «Enter numeric data».

– клас еквівалентності: поля форми заповнені латиницею

Вхідні дані: «qwerty»

Результат: повідомлення «Enter numeric data».

– клас еквівалентності: поля форми заповнені кирилицею.

Вхідні дані: «йцукен»

Результат: повідомлення «Enter numeric data».

– клас еквівалентності: поля форми заповнені спецсимволами.

Вхідні дані: «*!№;%:?)_»

Результат: повідомлення «Enter numeric data».

– клас еквівалентності: поля форми заповнені латиницею та кирилицею

Вхідні дані: «йцукasdf».

Результат: повідомлення «Enter numeric data».

– клас еквівалентності: поля форми заповнені латиницею та кирилицею, спецсимволами та числами.

Вхідні дані: «й1342цукas*():№df».

Результат: повідомлення «Enter numeric data».

– клас еквівалентності: поля форми заповнені правильно.

Вхідні дані: поле «Назва теми» – «qwerty»; інші поля «10».

Результат: повідомлення про помилку відсутнє, результат розрахунків виведено на екран та збережено в базу даних.

Тестування варіанта використання «Збереження отриманих результатів в базу даних»

Вхідні дані: слова та числа.

Слово – набір символів латиницею або кирилицею.

Перше поле «Назва теми» приймає слова, числа та спецсимволи. Інші поля приймають лише числа.

При введенні дробових чисел у відповідні поля форми ціла частина повинна відділятися комою від дробової.

При недопустимих значеннях даних на екран виводиться повідомлення в про помилку:

– поля форми залишилось пустим – повідомлення про обов’язковість заповнення кожного поля форми з текстом «Enter numeric data»;

– поля форми заповнені не числовими символами – повідомлення про обов’язковість заповнення кожного поля форми з текстом «Enter numeric data».

Якщо поля форми заповнені вірно, результат розрахунків виводиться на екран та зберігається в базу даних.

Тестові вимоги:

– перевірити, чи буде відбуватися перевірка, якщо поля форми залишилось пустим;

– перевірити, чи буде відбуватися перевірка, якщо поля форми заповнені не числовими символами;

Розробка випадків для тестових вимог

Перевірити, чи буде відбуватися перевірка введених даних у відповідні поля форми

– клас еквівалентності: поле введення пароля залишилось пустим

Вхідні дані: «».

Результат: дані не збережено в базу даних, повідомлення «Enter numeric data».

– клас еквівалентності: поля форми заповнені латиницею

Вхідні дані: «qwerty»

Результат: дані не збережено в базу даних, повідомлення «Enter numeric data».

– клас еквівалентності: поля форми заповнені кирилицею.

Вхідні дані: «йцукен»

Результат: дані не збережено в базу даних, повідомлення «Enter numeric data».

– клас еквівалентності: поля форми заповнені спецсимволами.

Вхідні дані: «*!№;%:?)_»

Результат: дані не збережено в базу даних, повідомлення «Enter numeric data».

– клас еквівалентності: поля форми заповнені латиницею та кирилицею

Вхідні дані: «йцукасdf».

Результат: дані не збережено в базу даних, повідомлення «Enter numeric data».

– клас еквівалентності: поля форми заповнені латиницею та кирилицею, спецсимволами та числами.

Вхідні дані: «й1342цукас*():№df».

Результат: дані не збережено в базу даних, повідомлення «Enter numeric data».

– клас еквівалентності: поля форми заповнені правильно.

Вхідні дані: поле «Назва теми» – «qwerty»; інші поля «10».

Результат: повідомлення про помилку відсутнє, результат розрахунків виведено на екран та збережено в базу даних.

Тестування варіанта використання «Перегляд даних збережених в базу даних»

При натисканні на кнопку «Previous results» користувача перенаправляє на вебсторінці де відображаються дані збережені в базу даних, у разі помилки на вебсторінці виводиться відповідне повідомлення:

- якщо база даних пуста – «Empty database»;
- доступ до бази даних відсутній – «Error».

Тестові вимоги:

– перевірити який результат буде відображатися на вебсторінці перегляду минулих результатів розрахунків;

Розробка випадків для тестових вимог

Перевірити який результат буде відображатися на вебсторінці перегляду минулих результатів розрахунків:

– клас еквівалентності: база даних пуста

Вхідні дані: пуста база даних.

Результат: вебсторінка з повідомленням «Empty database».

– клас еквівалентності: база даних відсутня

Вхідні дані: відсутні.

Результат: вебсторінка з повідомленням «Error».

– клас еквівалентності: база даних

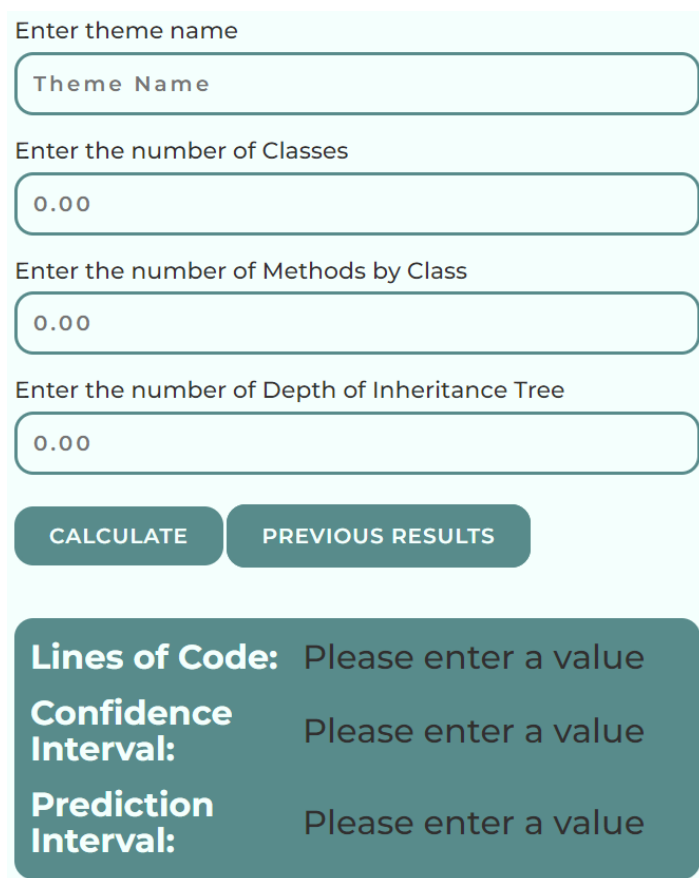
Вхідні дані: база даних.

Результат: вебсторінка з повідомленням результатами розрахунків.

3.3.4 Випробування програмного забезпечення для оцінювання розміру WordPress тем на PHP

Програму та методику випробувань розробленого програмного забезпечення описано в Додатку Д. Проведемо випробування розробленого програмного забезпечення згідно з методикою описаної у додатку Д.

Запуск програмного забезпечення відбувається у вікні браузера за допомогою файлу `index.php`, головне вікно інтерфейсу програмного забезпечення зображено на рисунку 3.7.



Enter theme name

Theme Name

Enter the number of Classes

0.00

Enter the number of Methods by Class

0.00

Enter the number of Depth of Inheritance Tree

0.00

CALCULATE PREVIOUS RESULTS

Lines of Code: Please enter a value

Confidence Interval: Please enter a value

Prediction Interval: Please enter a value

Рисунок 3.7 – Головне вікно інтерфейсу програмного забезпечення

Якщо користувач вводить неправильні дані, то система виводить повідомлення про помилку, що приведенне на рисунку 3.8.

The screenshot shows a web form with four input fields, each containing the text 'Qwerty' in red, indicating an error. The fields are labeled: 'Enter theme name', 'Enter the number of Classes', 'Enter the number of Methods by Class', and 'Enter the number of Depth of Inheritance Tree'. Below the fields are two buttons: 'CALCULATE' and 'PREVIOUS RESULTS'. At the bottom, a dark green box contains the following text:

Lines of Code:	Enter numeric data
Confidence Interval:	Enter numeric data
Prediction Interval:	Enter numeric data

Рисунок 3.8 – Головне вікно інтерфейсу програмного забезпечення з повідомленням про помилку

Якщо користувач вводить дані вірно, то система виводить результат розрахунків та зберігає його в базу даних, що приведенне на рисунку 3.9.

The screenshot shows the same web form as in Figure 3.8, but with valid input. The input fields contain: 'Agnar', '137', '4.44', and '1.31'. The 'CALCULATE' and 'PREVIOUS RESULTS' buttons are still present. The results are displayed in a dark green box at the bottom:

Lines of Code:	17.78
Confidence Interval:	12.3 - 25.5
Prediction Interval:	4.89 - 63.67

Рисунок 3.9 – Головне вікно інтерфейсу програмного забезпечення з результатами розрахунків

3.4 Результати розробки програмного забезпечення для оцінювання розміру WordPress тем на PHP

В процесі виконання роботи та розробки програмного забезпечення для оцінювання розміру WordPress тем на PHP була виконана програмна реалізація клієнтської та серверної частин програмного забезпечення та реалізовані наступні вимоги до програмного забезпечення:

- 1) Відкриття вікна графічного інтерфейсу програмного забезпечення.
- 2) Створення графічних компонентів інтерфейсу для введення назви та метрик WordPress теми.
- 3) Створення графічних компонентів інтерфейсу для виведення розрахованого розміру, довірчого інтервалу та інтервалу передбачення.
- 4) Виклик функції обчислення розміру плагіну в тисячах строк коду, довірчого інтервалу та інтервалу передбачення та збереження результатів розрахунків в базу даних.
- 5) Виклик функції для перегляду минулих результатів розрахунків.

В робочому проєкті було розроблено програмне забезпечення для оцінювання розміру WordPress тем на PHP, проведено тестування розробленого програмного забезпечення, створено документацію до програмного забезпечення.

Мовою програмування PHP було виконано кодування алгоритмів, що були розроблені на стадії технічного проєкту програмного забезпечення. Інтерфейс розроблений за допомогою HTML та CSS. Текст коду програми представлений у додатку Б.

Знімок екрана програмного забезпечення показано на рисунку 3.10.

Enter theme name

Theme Name

Enter the number of Classes

0.00

Enter the number of Methods by Class

0.00

Enter the number of Depth of Inheritance Tree

0.00

CALCULATE PREVIOUS RESULTS

Lines of Code: Please enter a value

Confidence Interval: Please enter a value

Prediction Interval: Please enter a value

Рисунок 3.10 – Знімок екрана програмного забезпечення

Програмно-апаратне забезпечення для клієнта:

- будь-яка операційна система, що забезпечує мережеву взаємодію по протоколу TCP/IP;
- веббраузер з підтримкою HTML5, наприклад, Mozilla Firefox, Опера, або Google Chrome 30.0 і вище;
- підключення до мережі Internet.

Інструкція користувача до розробленого програмного забезпечення представлена в додатку Г.

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ І ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WORDPRESS ТЕМ НА PHP

Ця кваліфікаційна робота присвячена побудові нелінійної регресійної моделі для оцінювання розміру WordPress тем на PHP та розробці програмного забезпечення для реалізації цієї моделі.

Доцільність розробки програмного забезпечення можна обґрунтувати з економічної точки зору. Якщо ми розрахуємо собівартість програмного забезпечення та прибуток. Основним критерієм економічною доцільністю витрат на розробку програмного продукту, є річний економічний ефект.

Щоб обґрунтувати доцільність розробки цього програмного забезпечення з економічної точки зору потрібно розрахувати його собівартість та прибуток, який буде отриманий в результаті продажу програмного забезпечення. Основний показник, який визначає економічну доцільність витрат на розробку програмного продукту – є річний економічний ефект.

Саме ефективний результат – оптимальний результат при мінімальних витратах на розробку. А розрахунок ефективності надає інформацію, наскільки виправдані вкладення, яка результативність роботи компанії, ці показники характеризують оптимальність використання ресурсів протягом процесу розробки.

Ефективність витрат – показник економічної ефективності програмного продукту, який розраховується на стадії створення і впровадження й характеризується терміном окупності розробленого програмного забезпечення. В цьому розділі розраховується собівартість розробки програмного продукту, виконується оцінювання окупності розроблюваного проекту.

4.1 Розрахунок витрат на створення та впровадження програмного забезпечення

Основними витратами під час розробки програмного забезпечення є витрати на виплату заробітної плати для розробників, експлуатацію сучасного ПК, за

допомогою якого виконується розробка, засоби розробки та витрати на матеріали й комплектуючі. Розробка програмного забезпечення виконується спеціалістом, заробітна плата якого складає 15000 грн на місяць. Додаткова заробітна плата складає 10% від основної. Виходячи з цього, основна й додаткова заробітна плата розробника програмного забезпечення 16500 грн на місяць, а вартість сучасного ПК складає близько 14000 грн (середня вартість машини на базі процесора Intel Core i5 чи аналогічного процесора AMD). Вартість кіловат-години електроенергії дорівнює 1.44 грн. Витрати на різні допоміжні матеріали представлені в таблиці 4.1.

Таблиця 4.1 – Витрати на допоміжні матеріали

Пункт витрат	Сума витрат, грн
Програмне забезпечення	1200
Папір	200
Непередбачені витрати	300
Матеріали і комплектуючі вироби	350

Вартість розробки програмного забезпечення розраховується за формулою:

$$C_{\text{ПР}} = (З_{\text{ЗП}} + З_{\text{СЗ}} + З_{\text{ЗГ}} + З_{\text{Е}}) * T + З_{\text{М}}, \quad (4.1)$$

де T – тривалість розробки, міс.;

$З_{\text{ЗП}}$ – основна і додаткова заробітна плата, грн;

$З_{\text{СЗ}}$ – відрахування на соціальні заходи (38% від основної й додаткової заробітної плати), грн;

$З_{\text{ЗГ}}$ – загальногосподарські витрати (10% від заробітної плати), грн;

$З_{\text{Е}}$ – витрати на електроенергію;

$З_{\text{М}}$ – витрати на різні основні й допоміжні матеріали.

Розрахуємо $З_{\text{Е}}$ при споживанні потужності 0,3 Вт, тривалості роботи на місяць рівної $21 * 8 = 168$ годин та вартості кіловат-години електроенергії 6 грн:

$$З_{\text{Е}} = 168 * 6 * 0,3 = 302,4 \text{ грн}$$

Занесемо в таблицю усі витрати пов'язані з розробкою програмного забезпечення.

Таблиця 4.2 – Витрати на розробку програмного забезпечення

Найменування	Одиниця виміру	Кількість
Тривалість розробки	міс.	2
Заробітна плата	грн	16500
Відрахування на соціальні заходи	грн	6270
Загальногосподарські витрати	грн	1650
Витрати на допоміжні матеріали	грн	2500
Витрати на електроенергію	грн	302,4

За формулою (4.1) вартість розробки програмного забезпечення становить:

$$C_{\text{пр}} = (16500 + 6270 + 1650 + 302,4) * 2 + 2500 = 51944,8 \text{ грн},$$

Амортизаційні відрахування на устаткування складають 60% балансової вартості в рік:

$$A_{\text{об}} = 14000 * 0,6 = 8400 \text{ грн}$$

У масштабах підприємства річні витрати на основні і допоміжні матеріали визначаються в розмірі 5% вартості основного устаткування:

$$B_{\text{м}} = 15000 * 0,05 = 750 \text{ грн}$$

Річний обсяг робіт ПЕОМ у годинах визначається в такий спосіб:

$$\Phi_{\text{м}} = 253,3 * T_{\text{з}}$$

де $T_{\text{з}}$ – це середнє місячне завантаження устаткування (близько 7 годин), 253,3 – середня кількість робочих днів на рік;

$$\Phi_{\text{м}} = 253,3 * 7 = 1773,1 \text{ год.}$$

Витрати на електроенергію $З_{\text{е}}$ при 1773,1 годинах роботи устаткування в рік складуть:

$$З_{\text{е}} = 1773,1 * 0,3 * 6 = 765,9 \text{ грн}$$

Експлуатаційні витрати для ПЕОМ за рік складуть:

$$З_{\text{зр}} = 8400 + 750 + 765,9 = 9915,9 \text{ грн}$$

У перший рік витрати на створення й експлуатацію програмного забезпечення складуть:

$$Z_{cp} = 51944,8 + 9915,9 = 61860,7 \text{ грн}$$

4.2 Розрахунок економічної ефективності розробки та впровадження програмного забезпечення

Визначити пряму економічну ефективність можна, основуючись на тому, що впровадження програмного забезпечення вивільняє одного працівника (за експертною оцінкою фахівців підприємства).

Зарплата одного працівника в рік складає:

$$20000 * 12 * 1 = 240000 \text{ грн}$$

Річний економічний ефект розраховується по формулі:

$$\mathcal{E}_{\text{рік}} = \Delta C_n + E_n * k,$$

де ΔC_n – вивільнені кошти після впровадження системи (240000 грн) мінус експлуатаційні витрати (9915,9 грн);

E_n – коефіцієнт ефективності (дорівнює коефіцієнта амортизації (0,6));

k – одноразові витрати на впровадження продукту (61860,7).

$$\mathcal{E}_{\text{рік}} = 240000 - 9915,9 - 61860,7 * 0,6 = 192967,68 \text{ грн}$$

Строк окупності системи розраховується по формулі:

$$T = \frac{k}{\Delta C} = \frac{61860,7}{240000 - 9915,9} \approx 0,26 \text{ року}$$

Отже, строк окупності розробленого програмного забезпечення складає приблизно 3 місяці.

В результаті за проведеними розрахунками, після впровадження програмного забезпечення строк його окупності складає приблизно 3 місяці протягом першого року експлуатації. Тому можна сказати що розроблений програмний продукт – економічно обґрунтований та вигідний.

5 ОХОРОНА ПРАЦІ

Охорона праці – система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів і засобів, направлених на збереження і працездатності людини в процесі трудової діяльності.

Дана система включає в себе:

- розробка заходів щодо усунення шкідливого впливу на людину та створення нормальних умов праці;
- правила з техніки безпеки та виробничої санітарії;
- аналіз шкідливих та небезпечних факторів, які впливають на людину;
- спеціальні норми охорони праці осіб, які працюють в тяжких, шкідливих та небезпечних виробничих умовах. Безпека людини на виробництві залежить від багатьох факторів;
- норми з охорони праці жінок, неповнолітніх та осіб з низькою працездатністю.

Здебільшого ця система визначається кваліфікацією та відповідальністю робітників управління.

Велике значення для підвищення безпеки праці у виробництві має система стандартів безпеки праці, яка включає значну кількість взаємопов'язаних стандартів, розроблених для запобігання ушкодженням і захворювань працюючих.

Закон України “Про охорону праці” визначає положення щодо реалізації права на охорону їх життя і здоров'я в процесі трудової діяльності, регулює за участю відповідних органів відносини між власником підприємства, установи і організації і робітником з питань безпеки, гігієни праці і виробничої санітарії, встановлює єдиний порядок організації охорони праці на Україні.

Поліпшення умов праці, підвищення їх безпеки нешкідливості має велике економічне значення. Воно впливає на економічні результати виробництва, на продуктивність праці, якість і собівартість продукції, що виготовляється.

Підвищення умов праці призводить до таких соціальних результатів, як покращення здоров'я працюючих, ріст ступеня задоволення працею, зміцнення

трудової дисципліни, підвищення престижу ряду професій, ріст виробничої активності і поліпшення ряду інших показників, які характеризують більш високий ступінь розвитку працюючих.

Охорона природи, оскільки очищення і знешкодження стічних вод і викидів в атмосферу, боротьба із шумом і вібрацією, захист від електромагнітних полів та іонізуючих випромінювань служать не тільки цілям охорони праці, але й одночасно сприяють охороні середовища мешкання людини.

5.1 Аналіз небезпечних і шкідливих факторів у відділі розробки програмного забезпечення

При роботі співробітники відділу зазнають впливу небезпечних і шкідливих факторів, як-то:

- недостатня освітленість робочого місця;
- підвищений рівень шуму і вібрації від роботи вентиляторів, принтерів;
- висока ймовірність виникнення пожеж у зв'язку з наявністю великої кількості паперу;
- підвищена напруга в електричній мережі, замикання якої може відбутися через тіло людини;
- вплив неіонізуючих електромагнітних випромінювань, електростатичних і магнітних полів, які виникають при роботі комп'ютерів;
- недостатній рівень параметрів мікроклімату і температури, відносної вологості повітря, швидкості руху повітря.

Робота співробітника вимагає від нього значної зорової уваги, і як наслідок, у приміщенні повинно бути правильно спроектоване і розташоване освітлення.

В залежності від часу доби та джерела світла на підприємстві використовується освітлення трьох видів: природне, штучне і комбіноване.

У денний час і ясну погоду використовується природне освітлення через світлові віконні пройми. Для нього характерна висока розсіяність і це є сприятливим фактором для здорової діяльності.

У передвечірній час і за несприятливих погодних умовах використовується комбіноване освітлення. Оскільки денне світло не забезпечує достатню освітленість робочого місця, то використовуються настільні лампи місцевого освітлення.

У вечірній час взимку використовується штучне освітлення. В якості джерел використовуються лампи накаливання.

Рекомендована і дійсна освітленість у приміщенні відділу товарно-матеріальних цінностей складає 750 лк.

Одним із найшкідливіших факторів, що чинить негативний вплив на працездатність робітників, є шум. Джерелами шуму у відділі є вентилятори, ЕОМ.

Для забезпечення нормативного рівня шуму у виробничому приміщенні відділу і на робочих місцях використовуються шумопоглинаючі засоби: важкогорючі спеціальні перфоровані плити, мінеральна вата з максимальним коефіцієнтом звукопоглинання в межах частот 31,5–8000Гц.

Рівень вібрації при виконанні робіт з використанням ЕОМ не повинен перевищувати допустимих значень. Оскільки в бюро використовуються 2 ПК з одним струйним принтером, то рівень вібрації, який створюють один принтер і два процесори, знаходиться в межах допустимих норм.

У відділі є меблі, які являються легкозаймистими предметами. Існує ризик у випадку виникнення пожежі, швидке її поширення по приміщенню.

Електричний струм, проходячи через тіло людини справляє тепловий, хімічний, біологічний вплив, порушуючи нормальну життєдіяльність організму. Будь-який з цих впливів може привести до електротравм.

У відділі є холодильник, електрочайник, кондиціонер, касовий апарат. Електрошкідливість здійснюється через трьохштирьові штепсельні вилки, які заземлені на штатну систему заземлення.

Робота комп'ютера супроводжується електромагнітним і електростатичним випромінюванням. Навколо електростатично-зарядженого монітора підвищується концентрація пилу. Такий електризований пил може викликати запалення шкіри, а

електромагнітне випромінювання може призвести до зниження загальної продуктивності співробітника.

Потужність дози рентгенівського випромінювання у відділі на відстані 0.05 м від екрана не перевищує $7,74 \cdot 10^{-12}$ А/кг, а вміст озону в повітрі робочої зони відповідає еквівалентній дозі 0.1 мг/м³, окислів азоту – 5 мг/м³, вміст пилу – 4 мг/м³.

Мікрокліматичні параметри (температура, відносна вологість повітря, швидкість руху повітря) впливають у першу чергу на продуктивність праці та самопочуття людини, а також на надійність роботи обчислювальної техніки. Забезпечення достатньої роботи системи вентиляції та опалення також підтримують працездатність співробітника.

Приміщення відділу обладнане системою опалення, кондиціонування повітря. Параметри мікроклімату, які діють у відділу, відповідають встановленим нормам.

5.2 Розрахунок освітленості у відділі розробки програмного забезпечення

До сучасного виробничого освітлення висуваються наступні вимоги:

- відповідність рівня освітленості робочих місць характеру роботи, що здійснюється;
- досить рівномірний розподіл яскравості на робочих місцях поверхнях і у навколишньому просторі;
- неперервність освітленості у часі;
- оптимальна направленість світлового потоку, який випромінюють світлові прилади;
- довговічність, економічність, електро і вогненебезпека, естетичність, зручність і простота експлуатації;

Штучне освітлення за конструктивним виконанням може бути загальним і місцевим.

При загальному освітленні всі робочі місця отримують освітлення від загальної освітлюваної установки. Комбіноване освітлення поряд із загальним включає місцеве освітлення, що зосереджує світловий потік безпосередньо на робочих місцях. Використання лише місцевого освітлення не допустимо, оскільки виникає необхідність постійної адаптації зору, створюються тіні та інші негативні фактори.

Для штучного освітлення приміщень використовують люмінесцентні лампи. В них висока світлова віддача і тривалий термін служби. Разом із тим необхідно врахувати недоліки: висока пульсація світлового потоку, необхідність використання спеціальної апаратури, складність їх утилізації із-за наявності у лампах парів ртуті.

Розрахунок освітлення здійснюється для приміщень завдовжки 5-м і завширшки 6-м методом коефіцієнта використання світлового потоку наступними етапами:

1. Визначення світлового потоку (F), що падає на поверхню, здійснимо за формулою (5.1):

$$F = \frac{E_{min} * k_z * S * z}{\eta} \quad (5.1)$$

де E_{min} – нормована мінімальна освітленість, Лк (визначається за таблицею).

В нашому випадку $E_{min} = 300$ Лк;

k_z – коефіцієнт запасу, який враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації.

В нашому випадку $k_z = 1,5$;

S – площа приміщення, яке освітлюється ($S = 30 \text{ m}^2$);

z – коефіцієнт використання світлового потоку з урахуванням коефіцієнта відображення стелі = 70% і стін = 50%; виражається відношенням світлового потоку, який падає на розрахункову поверхню, до сумарного потоку всіх ламп і виражається у долях одиниці.

Значення η визначається за таблицею коефіцієнтів використання. Для цього розрахуємо індекс приміщення (i) за формулою (5.2):

$$i = \frac{S}{h \cdot (A+B)} \quad (5.2)$$

де h – розрахункова висота підвісу, $h = 2,7$ -м;

A – ширина приміщення, $A = 5$ -м;

B – довжина приміщення, $B = 6$ -м.

За таблицею знаходимо коефіцієнт використання люмінесцентних світильників $\eta = 0,32$.

Підставимо у формулу й отримаємо:

$$F = \frac{300 \cdot 1,5 \cdot 30 \cdot 1,1}{0,3} = 49500 \text{ Лм}$$

Для освітлення приміщення даного відділу обираємо люмінесцентні лампи типу ЛДУ 80, світловий потік яких дорівнює 4070 Лм.

2. Розрахуємо необхідну кількість світильників для приміщення з комп'ютерами при загальному рівномірному освітленні.

Кількість необхідних ламп N розраховується за формулою (5.3):

$$N = \frac{F}{F_{\text{л}}} \quad (5.3)$$

де $F_{\text{л}}$ – світловий потік, 4070 Лм.

$$N = \frac{49500}{4070} = 13,6 \approx 14 \text{ шт}$$

З розрахунку освітленості виробничого приміщення виходить, що для правильної подачі світла і правильної освітленості робочих місць відділу необхідно використовувати світильники типу УПМ. Кожний світильник комплектувати 2 лампами. Розмістити світильники двома паралельними рядами по три в кожному ряді й один світильник перпендикулярно двом паралельним у середині приміщення.

5.3 Розробка заходів щодо зменшення впливу небезпечних і шкідливих факторів

Для зменшення впливу небезпечних і шкідливих факторів у приміщенні необхідно здійснити ряд заходів.

1. Для правильної подачі світла і правильної освітленості робочих місць слід використовувати світильники типу УПМ. Кожний світильник комплектувати 2 лампами. Розташувати світильники двома паралельними рядами по три в кожному ряді й один світильник перпендикулярно двом паралельним усередині приміщення.

2. Зниження впливу шуму на робітників відділу можна досягти шляхом більш раціонального розташування джерел шуму у приміщенні або зменшення рівня шуму у джерелі його виникнення, використання додаткової звукоізоляції та звукопоглинання, впровадження мал шумного обладнання (більш сучасного) і правильного планування режиму праці та відпочинку робітників.

3. Зменшити ймовірність виникнення пожежі у відділі можна шляхом зменшення документообігу, наприклад, повністю автоматизувати обліковоаналітичний процес, при якому значна кількість документів буде залишатись в електронній формі.

4. Для забезпечення безпечної для робітників відділу, безаварійної роботи електроприладів, необхідно поряд з положенням, виконанням, і обладнанням так організувати їх експлуатацію, щоб повністю виключити будь-яку ймовірність помилок з боку робітників. Для організації безпечної експлуатації електроприладів необхідно підвищити технічну грамотність і свідому дисципліну робітників, які зобов'язані чітко дотримуватись організаційних і технічних заходів відповідно до Правил технічної експлуатації електроприладів споживачами та Правил техніки безпеки при експлуатації електроприладів споживачами.

5. Знизити вплив електромагнітних випромінювань на співробітників при роботі на комп'ютері можна шляхом встановлення захисних екранів. Такий спосіб захисту є дуже ефективним і може бути досягнутий шляхом раціонального вибору конструкції екрана і матеріалу для екранування.

6. Наявність пилу, озону, можна знищити шляхом обладнання робочого приміщення системою кондиціонування повітря. Всі ці заходи у сукупності сприяють злагодженій, безперебійній роботі відділу розробки програмного забезпечення, дозволяють підвищити продуктивність праці робітників.

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Вступ

Під навколишнім середовищем розуміють цілісну систему взаємопов'язаних природних і антропогенних об'єктів і явищ, під впливом і при безпосередньому використанні яких відбувається праця, побутова діяльність, відпочинок людей. Поняття «навколишнє середовище» включає соціальні, природні та штучно створені фізичні, хімічні та біологічні фактори, тобто все те, що впливає на життя і діяльність людини. Складовою частиною навколишнього середовища є природне середовище. Перед сучасним суспільством стоїть завдання не тільки зберегти природу, а й запобігти негативним наслідкам господарської діяльності людини в майбутньому.

Економічна діяльність у всіх її проявах здійснює забруднення навколишнього середовища. У процесі цієї діяльності забруднюються і стають дефіцитними ресурси повітря, води, територій, що здавалися нескінченними. Нині рівень забруднення досяг загрозливих розмірів, набувши по суті кризового характеру.

Цей процес посилюється розвитком виробничих сил і збільшенням маси речовин, що залучаються в господарський обіг. Через це в навколишнє середовище надходить все більше й більше різноманітних речовин, які йому чужі, а часом токсичні. Значна частина з них не включається в природний колообіг, накопичується в біосфері та зумовлює небажані екологічні наслідки. Відомо, що екологія – це наука взаємовідносин між живими організмами та сферою їх перебування, тому наслідки промислової та господарської діяльності людства можуть завдати непоправні збитки біосфері й велику шкоду людині.

Забруднюючі речовини, що потрапляють в природне середовище здатні переміщуватись на досить великі відстані, а закономірність цих процесів вивчена ще недостатньо. Ці речовини мігрують у великих кількостях в контурах окремих складових біосфери. Так в атмосфері вони переносяться повітряними течіями. Ступінь їх розсіювання формується швидкістю і напрямом переміщення

повітряних мас і залежить від метеорологічних умов. потрапивши у воду, забруднюючі речовини окислюються мікроорганізмами або адсорбуються частинками речовин, які є у воді.

Охорона навколишнього середовища являє собою форму відносин між суспільством і природою. Вона здійснюється різними засобами: економічними, правовими, науково-технічними, санітарно-гігієнічними, біологічними та іншими.

В загальному випадку проблема охорони навколишнього середовища зводиться до вирішення двох завдань:

- 1) організації раціонального природокористування;
- 2) забезпечення чистоти природних (екологічних) систем.

При здійсненні різних видів економічної діяльності суб'єкти господарювання використовують різноманітні природні ресурси: землю, воду, корисні копалини тощо. Проте ресурси ці обмежені. Обмеженість природних ресурсів була і залишається головною і дуже жорсткою умовою, що накладається на розвиток економіки та відповідно зростання суспільного добробуту.

Наслідком обмеженості природних ресурсів є конкуренція за їх застосування, тобто суперництво між альтернативними цілями використання ресурсів. Адже майже всі ресурси можуть використовуватися для задоволення найрізноманітніших потреб.

Раціональне природокористування означає розробку та здійснення концепції та конкретних заходів щодо раціонального використання і відтворення природних ресурсів, гармонічну взаємодію суспільства і природи, людини й навколишнього природного середовища.

Завдання організації раціонального природокористування вирішується шляхом:

- 1) оптимального розподілу ресурсів між різними господарськими цілями;
- 2) використання технологій, що зберігають ресурси;
- 3) проведення заходів щодо поповнення природних ресурсів.

Іншим, не менш важливим, завданням охорони навколишнього середовища є забезпечення чистоти природних екологічних систем, тобто водного середовища,

повітряного басейну, ґрунтових покривів тощо, з тим, щоб забезпечити населення екологічно чистими продуктами харчування, водою, повітрям і, в остаточному підсумку, зберегти високий рівень здоров'я населення та його активного довголіття.

6.2 Забруднення навколишнього середовища комп'ютерною компанією

Комп'ютерна компанія – це офісна компанія, яка займається наданням послуг з розробки та супроводження програмного забезпечення. Хоча компанія не займається промисловим виробництвом з викидом шкідливих речовин в атмосферу, воду та ґрунт, це не виключає існування джерел забруднення довкілля.

До таких джерел насамперед належать:

- електромагнітне випромінювання;
- дизельна електростанція;
- твердопаливні котли;
- відходи від застарілого обладнання та оргтехніки;
- побутове сміття;

Одночасно в офісі компанії можуть працювати до 50 комп'ютерів, стаціонарних телефонів та велика кількість серверної техніки, яка обслуговує роботу цього обладнання, аварійне чи планове відключення електроенергії призводить до значних перешкод у роботі персоналу. Для того, щоб уникнути таких ситуацій, на території компанії знаходиться дизельна електростанція. Вона працює як в аварійному, так і в резервному режимі.

Результатом роботи дизельної електростанції є продукти згоряння дизельного палива та шумове забруднення.

Твердопаливні котли використовуються в холодну пору року в системі опалення офісних приміщень. Для опалення використовуються деревні палети. Хоча таке джерело тепла має менший відсоток шкідливих викидів в атмосферу, ніж, наприклад, вугілля чи мазут, цей вид опалення не можна вважати екологічно чистим.

Побутові відходи – це залишки речовин і предметів, які утворюються в результаті побутової та господарської діяльності людини і які не можуть бути використані на місці утворення, а їх накопичення і зберігання порушують санітарний стан навколишнього середовища.

Всі побутові відходи, які утворюються в процесі функціонування компанії, можна розділити на рідкі та тверді. До рідких побутових відходів належать нечистоти з вигребів туалетів, помий (від миття посуду, підлоги тощо) і стічні води (побутові, зливові). До твердих побутових відходів можна віднести сміття (побутові відходи) та кухонні відходи. Зокрема, твердими відходами є папір, картон, скло, пластмаса, продукти харчування.

Високі темпи розвитку технологій привели до того, що утилізація оргтехніки стала особливо актуальною. Комп'ютери та інші пристрої не просто стають малопотужними, але і застарівають морально, що змушує купувати нові ноутбуки та ПК.

До складу оргтехніки входить маса деталей, які можна піддати переробці. Утилізація оргтехніки та обладнання дозволяє зберегти стан навколишнього середовища і є досить прибутковим бізнесом. Тим більше сьогодні ця послуга дуже популярна. Вже сьогодні кількість випущених комп'ютерів перевищило за один мільярд. Додайте сюди друковані та копіювальні машини – актуальність переробки відпрацьованих пристроїв зростає з їх числом.

Видалення техніки має на увазі складний ланцюг робіт. Але для початку потрібно визначитися з поняттям. За КВЕД і ОКПД комп'ютерний лом, переробка якого проводиться в спеціальних організаціях, ділиться на:

- монітори;
- флеш – накопичувачі;
- жорсткі диски;
- системні блоки;
- мережеве обладнання;
- друковано – копіювальні машини;
- шредери, плоттери;

– клавіатури, миші.

Всі вони мають свої особливості, від яких залежить подальша переробка електронної техніки. Хоча в цілому складові кожного пристрою схожі, проте, утилізація моніторів відрізняється від процесу утилізації картриджів для принтерів. Всі вони вимагають спеціальної підготовки й обробки.

Найчастіше, коли мова заходить про старе комп'ютерне обладнання, згадують про зміст дорогоцінних металів. Але мало хто говорить про що до 65 складу техніки входять сполуки ртуті, цинку, свинцю і кадмію. Поки вони знаходяться в складі оргтехніки, вони є безпечними, але потрапляючи на звалища і взаємодіючи з вологою, вони розпадаються на з'єднання, які отруюють ґрунт і воду. Тому дуже важливо правильно утилізувати комп'ютерну техніку.

Питання з дорогоцінними металами теж є не таким простим. Обладнання, що містить золото, срібло та інші елементи, знаходиться на балансі підприємства. Керівники не можуть просто взяти й викинути комп'ютерний лом на звалище, інакше підприємство отримає солідний штраф. Для цього потрібна наявність певної документації та сертифікатів.

Генератором електромагнітного забруднення є, в першу чергу, велика кількість комп'ютерної техніки на території офісу. Більшість із них працює 8 годин на добу, а деякі не вимикаються цілодобово. Випромінювачами в цьому випадку є і процесор, і монітор. Випромінювання останнього значно вищі, особливо його бічні та задні стінки, адже вони не мають спеціального захисного покриття, як у лицьовій частині монітора. Крім того, в офісі є дві кухні, кожна з яких обладнана 5 мікрохвильовими печами.

Електромагнітне випромінювання має несприятливий вплив на організм людини. Його наслідками можуть бути головний біль, порушення сну, перевтома і, навіть, у деяких випадках стенокардія.

6.3 Розробка заходів щодо зменшення забруднення

Охорона навколишнього природного середовища, раціональне використання природних ресурсів, забезпечення екологічної безпеки життєдіяльності людини –

невід'ємна умова сталого економічного та соціального розвитку України. Відносини у галузі охорони навколишнього природного середовища в Україні регулюються законом України «Про охорону навколишнього природного середовища», а також розробленими відповідно до нього земельним, водним, лісовим законодавством, законодавством про надра, про охорону атмосферного повітря, про охорону і використання рослинного і тваринного світу та іншим спеціальним законодавством.

Для того, щоб зменшити рівень забруднення навколишнього середовища, варто вжити певних заходів, які мінімізують цей вплив, якщо його неможливо усунути цілком. Серед таких заходів:

1) Очищення димових газів у мокрих золоуловлювачах.

Електрофільтри на електростанціях застосовуються для досягнення найбільш глибокого очищення димових газів в основному на великих енергоблоках потужністю 300 МВт та більше. Мокрі золоуловлювачі, які працюють при маленьких питомих витратах води та невеликих перепадах тиску, встановлюються за котлоагрегатами середньої паропродуктивності. У мокрих золоуловлювачах уловлювання часток золи на плівці води, яка стікає по його стінках, здійснюється за рахунок відцентрової сили, яка діє на частки. Ефективність апарата не перевищує 90%.

2) Використання природного газу для опалення офісних приміщень замість деревних палетів.

Згідно з показниками, наведеними в табл. 7.1, рівень шкідливих викидів від згоряння природного газу є найнижчим серед перерахованих видів палива. Проте зміну джерела енергії слід проводити, враховуючи економічну ефективність, оскільки вартість цих видів палива не є однаковою.

3) Раціональне позбавлення від побутових відходів.

Деякі побутові відходи можна безпечно спалювати для отримання енергії. Вторинну сировину (макулатуру, скло, пластмаси) можна відправляти на переробку, а не вивозити на сміттєзвалища. Крім того, зменшити кількість побутових відходів допоможе повторне їх використання (наприклад, скляних чи

пластикових пляшок та контейнерів для їжі), скорочення споживання та використання меншої кількості пакування.

4) Зменшити рівень електромагнітного забруднення та захиститися від його надмірного впливу.

Зрозуміло, що неможливо повністю обійтися без електроприладів та комп'ютерів, проте можна дещо зменшити їх негативний вплив. Для цього слід вимикати з електромережі комп'ютери, телефони та обладнання, з якими ніхто не працює. Також варто час від часу виходити на прогулянки та робити перерви в роботі з комп'ютером. Що стосується мікрохвильових печей, то їх потужність може змінюватись, тому час від часу треба звертатися до майстра, щоб контролювати рівень випромінювання.

5) Утилізація застарілої оргтехніки та комп'ютерного обладнання.

У принципі, будь-який комп'ютер чи оргтехніку можна переробити й пустити у вторинне використання. При грамотній утилізації близько 95% відходів техніки здатні повернутися до нас в тому чи іншому вигляді, і приблизно 5% відправляються на звалища або заводи з переробки твердих побутових відходів.

Співвідношення ручної та автоматизованої праці на заводі по переробці комп'ютерної техніки залежить від її типу. Для монітора це співвідношення приблизно 50 на 50 – розбирання старих кінескопів є досить трудомістким заняттям. Для системних блоків та оргтехніки частка автоматичних операцій вища.

НР вперше запропонувала переробку відслужив свій термін продукції ще в 1981 році. Сьогодні НР володіє інфраструктурою зі збору та переробки використаних ПК і оргтехніки в 50 країнах світу. У рік утилізації піддається близько 2,5 млн одиниць продукції. В одному тільки 2007 НР переробив близько 100 тис тонн списаного обладнання та витратних матеріалів, – майже в півтора раза більше, ніж роком раніше.

Будь-яка якісна переробка комп'ютерної техніки по КВЕД вимагає попередньої підготовки:

– Жорсткі диски проходять низькорівневу форматизацію.

– Утилізація картриджів вимагає попереднього очищення від тонера, до складу якого входять шкідливі хімічні елементи.

В цілому весь електро лом обробляється, розбирається і сортується. Пластикові, металеві та скляні деталі оргтехніки переробляються окремо.

Перший етап завжди проводиться вручну. Це – видалення всіх небезпечних компонентів. У сучасних настільних ПК і принтерах таких компонентів практично немає. Але переробці піддаються, як правило, комп'ютери та техніка, випущені в кінці 90-х – самому початку 2000-х років, коли плоских рідкокристалічних моніторів просто не існувало. А в кінескопних моніторах міститься чимало сполук свинцю. Інша категорія продукції, містить небезпечні елементи, – ноутбуки. В акумуляторах і екранах застарілих моделей є певна кількість ртуті, яка також дуже небезпечна для організму. Важливо відзначити, що в нових моделях ноутбуків від цих шкідливих компонентів позбулися.

Потім видаляються всі великі пластикові частини. У більшості випадків ця операція також здійснюється вручну. Пластик сортується в залежності від типу і подрібнюється для того, щоб надалі його можна було використовувати повторно. Решту відправляють у великій подрібнювач-шредер, і всі подальші операції є автоматизованими. Багато в чому технології переробки запозичені з гірської справи – приблизно таким же способом витягують цінні метали з породи.

Подрібнені в гранули залишки комп'ютерів піддаються сортуванню. Спочатку за допомогою магнітів витягуються всі залізні частини. Потім приступають до виділення кольорових металів, яких в ПК значно більше. Алюміній добувають з брухту за допомогою електролізу. У сухому залишку виходить суміш пластику і міді. Мідь виділяють способом флотації – гранули поміщають в спеціальну рідину, пластик спливає, а мідь залишається на дні. Сама ця рідина не отруйна, проте, робітники на заводі використовують захист органів дихання – щоб не вдихати пил.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було досягнуто поставленої мети – підвищення достовірності оцінювання розміру WordPress тем на PHP, для чого усі відповідні завдання було виконано у повному обсязі:

- проаналізовано існуючі моделі для оцінювання розміру програмного забезпечення, в тому числі Wordpress тем на PHP;
- аргументовано необхідність удосконалення трьохфакторної нелінійної регресійної моделі для оцінювання розміру PHP застосунків із відкритим кодом;
- зібрано емпіричні дані метрик Wordpress тем;
- побудована лінійна регресійна модель (лінійне рівняння регресії, довірчий інтервал та інтервал передбачення) для даних, нормалізованих через перетворення у вигляді десяткового логарифма;
- побудована нелінійна регресійна модель (нелінійне рівняння регресії, довірчий інтервал та інтервал передбачення) для оцінювання розміру WordPress тем на PHP;
- розроблено проєкт програмного забезпечення, що реалізує побудовану математичну модель для оцінювання розміру WordPress тем;
- розроблено програмне забезпечення для оцінювання розміру WordPress тем на PHP;
- проведено розрахунок економічної ефективності;
- розглянуто питання охорони праці;
- розглянуто питання охорони навколишнього середовища.

Наукова новизна полягає в удосконаленні трьохфакторної нелінійної регресійної моделі для оцінювання розміру PHP застосунків із відкритим кодом за рахунок знаходження параметрів моделі для WordPress тем на PHP, що дозволяє підвищити достовірність оцінювання їх розміру в порівнянні з існуючими трифакторними нелінійними регресійними моделями. У порівнянні з трифакторною нелінійною регресійною моделлю для оцінювання розміру PHP застосунків із відкритим кодом [5], побудована модель має більшу достовірність:

значення R^2 покращилось на 107,52%, MMRE покращилось на 72,08%, PRED (0,25) покращилось на 93,33%.

Технічне завдання представлено в додатку А. Текст коду програми представлений в додатку Б. Опис програми представлений в додатку В. Інструкція користувача – в додатку Г. Програма і методика випробувань програмного забезпечення у додатку Д.

Програмне забезпечення для оцінювання розміру WordPress тем на PHP, призначене для використання у веббраузері з підтримкою HTML5. Програмне забезпечення повністю пройшло тестування та випробування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Вікіпедія – WordPress. URL: <https://uk.wikipedia.org/wiki/WordPress> (дата звернення 28.09.2022).
2. Вікіпедія – Оцінка складності розробки ПЗ. URL: https://uk.wikipedia.org/wiki/Оцінка_складності_розробки_ПЗ (дата звернення 28.09.2022).
3. W3Techs – Usage statistics and market share of WordPress. URL: <https://w3techs.com/technologies/details/cm-wordpress> (retrieved 30.09.2022).
4. Tan H.B.K. Conceptual data model-based software size estimation for information systems / H.B.K Tan, Y. Zhao, H. Zhang // Transactions on Software Engineering and Methodology, 2009, vol. 19, issue 2, article No. 4.
5. Приходько С.Б. Оцінювання розміру РНР - застосунків з відкритим кодом за нелінійними регресійними моделями з різними факторами / С.Б. Приходько, М.В. Ворона // Збірник наукових праць Національного університету кораблебудування імені адмірала Макарова, 1(484)/2021. – Миколаїв: НУК, 2021. – С. 92-98. URL: [https://doi.org/10.15589/znp2021.1\(484\).13](https://doi.org/10.15589/znp2021.1(484).13)
6. Дорогань І.Ю. Математична модель для оцінювання розміру WordPress тем на РНР / І.Ю. Дорогань, А.В. Пухалевич // Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2022): Матеріали III Всеукраїнської науково-практичної інтернет конференції (26-28 жовтня 2022 р.). – Миколаїв: НУК, 2022. – С. 33-35. URL: <http://itconf.nuos.edu.ua/2022/proceedings/>
7. Вікіпедія – Метрика програмного забезпечення. URL: https://en.wikipedia.org/wiki/Software_metric (дата звернення 11.10.2022).
8. PhpMetrics is a static analysis tool for PHP. URL: <https://phpmetrics.org/#install> (дата звернення 11.10.2022).
9. Приходько С.Б. Методичні вказівки та завдання до виконання лабораторних робіт з дисципліни "Обробка експериментальних даних на комп'ютері" / С.Б. Приходько, Л.М. Макарова, К.С. Пугаченко. – Миколаїв: НУК, 2018 – 76 с.

10. Вікіпедія – PHP. URL: <https://uk.wikipedia.org/wiki/PHP> (дата звернення 13.10.2022).

11. Вікіпедія – HTML. URL: <https://uk.wikipedia.org/wiki/HTML> (дата звернення 13.10.2022).

12. Вікіпедія – CSS. URL: <https://uk.wikipedia.org/wiki/CSS> (дата звернення 13.10.2022).

13. Вікіпедія – MySQL. URL: <https://uk.wikipedia.org/wiki/MySQL> (дата звернення 13.10.2022).

14. Вікіпедія – Visual Studio Code. URL: https://uk.wikipedia.org/wiki/Visual_Studio_Code (дата звернення 13.10.2022).

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

Вступ

Назва проєкту розробки – Програмне забезпечення для оцінювання розміру WordPress тем на PHP, надалі програма.

Підстави для розробки

Завдання на кваліфікаційну роботу.

Призначення розробки

Програма призначена для оцінювання розміру WordPress тем на PHP.

Вимоги до програми

Вимоги функціональних характеристик.

Вимоги до складу функцій, що виконуються.

Програма має виконувати такі функції:

1) Інтерфейс для вводу значень метрик, за якими буде виконуватися оцінювання розміру WordPress тем у тисячах строк коду, а саме: кількість класів, середня кількість методів на клас та глибина дерева успадкування.

2) Проводити обчислення розміру WordPress тем на PHP.

3) Обчислювати довірчий інтервал нелінійної регресії розміру WordPress тем на PHP.

4) Обчислювати інтервал передбачення нелінійної регресії розміру WordPress тем на PHP.

5) Збереження результатів розрахунків в базу даних.

6) Перегляд минулих результатів розрахунків.

Вхідні дані:

– набір з трьох метрик WordPress теми, за якими потрібно провести оцінку розміру програмного забезпечення: кількість класів, середня кількість методів на клас та глибина дерева успадкування.

Вихідні дані:

– оцінка кількості строк коду програмного забезпечення, значення границь довірчого інтервалу нелінійної регресії та границь інтервалу передбачення нелінійної регресії розміру WordPress теми.

Вимоги до надійності не висуваються.

Вимоги до умов експлуатації не висуваються.

Вимоги до складу й параметрів технічних засобів.

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам'ять 2 Гб, простір на жорсткому диску 80 Мб.

Вимоги до інформаційної та програмної сумісності.

Для запуску та роботи з програмним забезпеченням необхідний веббраузер з підтримкою HTML5.

Вимоги до маркування та пакування.

Вимоги до маркування та пакування відсутні.

Вимоги до транспортування та зберігання.

Вимоги до транспортування та зберігання відсутні.

Вимоги до програмної документації

Програмна документація має включати: технічне завдання на розробку програми, текст програми, опис програми, інструкцію користувача, програму та методику випробувань.

Техніко-економічні показники не розраховуються

Стадії та етапи розробки

Стадії та етапи розробки програми представлені у таблиці А.2.

Таблиця А.2 – Стадії та етапи розробки

Стадія розробки	Етапи роботи	Термін виконання
1. Технічне завдання	Розробка та затвердження технічного завдання	22.10.2022
2. Ескізний проєкт	Розробка ескізного проєкту	23.10.2022
	Затвердження ескізного проєкту	25.10.2022
	Розробка технічного проєкту	27.10.2022

3. Технічний проект	Затвердження технічного проекту	30.10.2022
4. Робочий проект	Розробка програми	01.11.2022
	Розробка програмної документації	02.11.2022
	Випробування програми	03.11.2022

Порядок контролю і приймання

Контроль за кожним етапом розробки проводиться викладачем шляхом надсилання йому завершених розділів кваліфікаційної роботи. Кожна стадія розробки повинна бути представлена в зазначені строки та узгоджена із замовником (викладачем).

Приймання програми здійснюється представниками замовника у присутності представника розробника. Для приймання програми надається програмна документація, яка була розроблена відповідно до технічного завдання. Представники замовника установлюють відповідність програми технічному завданню. За результатами приймання програми представниками замовника складається акт приймання програми.

ДОДАТОК Б – ТЕКСТ ПРОГРАМИ

reset.css

```

/*Styles reset*/
*{
    padding: 0;
    margin: 0;
    border: 0;
}
*,*:before,*:after{
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
}
:focus,:active{outline: none;}
a:active,
a:hover,
a{
    text-decoration: none;
    color:inherit;
}

nav,footer,header,aside{display: block;}

html,body{
    height: 100%;
    width: 100%;
    font-size: 100%;
    line-height: 1;
    font-size: 14px;
    -ms-text-size-adjust: 100%;
    -moz-text-size-adjust: 100%;
    -webkit-text-size-adjust: 100%;
}
input,button,textarea{font-family:inherit;}

input::-ms-clear{display: none;}
button{cursor: pointer;}
button::-moz-focus-inner {padding:0;border:0;}

ul li{list-style: none;}
img{vertical-align: top;}

h1,h2,h3,h4,h5,h6{font-size:inherit;font-weight: 400;}

```

style.css

```
@import "reset.css";
```

```

@import
url('https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,300;0,400;0,500;
0,600;0,700;1,300;1,400;1,500;1,600;1,700&display=swap');

body {
  width: 100%;
  display: flex;
  align-items: center;
  justify-content: center;
  background-color: #F4FFFD;
}

.main {
  width: 100%;
  max-width: 680px;
  padding: 0 15px;
  display: flex;
  align-items:center;
  flex-direction: column;
}

form {
  width: 100%;
}

label {
  color:#312F2F;
  font-family: 'Montserrat', sans-serif;
  font-size: 22px;
  font-weight: 500;
  letter-spacing: 0px;
  line-height: 1.3em;
  display: flex;
  flex-direction: column;
}

input{
  cursor: pointer;
}

.form_field {
  color:#EF8354;
  font-size: 20px;
  font-family: 'Montserrat', sans-serif;
  font-weight: 600;
  line-height: 1em;
  letter-spacing: 3px;
  border-radius: 18px;
  padding: 15px;
  background-color:#F4FFFD;
  border: 3px #588B8B solid;
}

```

```

    margin: 5px 0 20px 0;
    transition: all linear 0.15s;
}

.form_field:hover{
    color:#588B8B;
    border: 3px #EF8354 solid;
    transition: all linear 0.15s;
}

.form_field:focus{
    color:#588B8B;
    border-radius: 13px;
    background-color:#FAFAFA;
    border: 3px #EF8354 solid;
    box-shadow: 0px 3px 20px 4px rgba(88,139,139,0.3);
    -webkit-box-shadow: 0px 3px 20px 4px rgba(88,139,139,0.3);
    -moz-box-shadow: 0px 3px 20px 4px rgba(88,139,139,0.3);
    transition: all linear 0.15s;
}

.sbmt_btn {
    color:#F4FFFD;
    font-size: 20px;
    font-family: 'Montserrat', sans-serif;
    font-weight: 600;
    line-height: 1em;
    text-transform: uppercase;
    letter-spacing: 1px;
    border-radius: 18px;
    padding: 15px 30px;
    background-color:#588B8B;
    border: 3px #588B8B solid;
    transition: all linear 0.15s;
    margin: 10px 0 0 0;
}

.sbmt_btn:hover, .sbmt_btn:focus{
    color:#EF8354;
    border-radius: 13px;
    background-color:#F4FFFD;
    border: 3px #EF8354 solid;
    transition: all linear 0.15s;
}

.result {
    width: 100%;
    display: flex;
    flex-direction: column;
    align-items:flex-start;
    background-color:#588B8B;

```

```

    border-radius: 18px;
    margin: 50px 0 0 0;
    padding: 10px 15px;
}

.result_wrap {
    width: 100%;
    display: flex;
    align-items: center;
    margin: 10px 0;
}

.result_title {
    flex-grow: 1;
    flex-basis: 40%;
    color: #F4FFFD;
    font-family: 'Montserrat', sans-serif;
    font-size: 32px;
    font-weight: 700;
    letter-spacing: 0px;
    line-height: 1.05em;
}

.result_data {
    flex-grow: 1;
    flex-basis: 60%;
    color: #312F2F;
    font-family: 'Montserrat', sans-serif;
    font-size: 32px;
    font-weight: 500;
    letter-spacing: 0px;
    line-height: 1.05em;
    padding-left: 10px;
}

.db_container {
    display: flex;
    flex-direction: column;
    align-items: center
}

.db_data_wrap {
    display: flex;
    flex-direction: row;
}

.db_data {
    background-color: #588B8B;
    padding: 15px;
    margin: 10px;
    border: 3px solid #312F2F;

```

```

        border-radius: 5px;
    }

    .db_data span{
        color:#F4FFFD;
        font-family: 'Montserrat', sans-serif;
        font-size: 20px;
        font-weight: 500;
        letter-spacing: 0px;
        line-height: 1.4em;
    }

    .db_data .db_data_value {
        color:#312F2F;
        font-weight: 600;
    }

```

head.php

```

<?php require_once 'func/functions.php'; ?>

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>ThemeLOCCount</title>
    <link rel="stylesheet" href="css/style.css">
    <script type="text/javascript" src="https://ajax.aspnetcdn.com/ajax/jquery/jquery-
3.6.0.min.js"></script>
    <script type="text/javascript" language="javascript">
        function call(){
            var msg = $('#calculate').serialize();
            $.ajax({
                type:'POST',
                url: 'func/process_form.php',
                data: msg,
                success: function(data){
                    $('#result').html(data);
                    $("#calculate").trigger("reset");
                },
                error: function(xhr, str){
                    alert('Error: ' + xhr.responseCode);
                }
            });
        }
    </script>
</head>

```

functions.php

```

<?php

```

```

$mysqli = false;
function connectDB() {
    global $mysqli;
    $mysqli = new mysqli("localhost", "root", "", "theme_loc_count");
    $mysqli->query("SET NAMES 'utf8'");
}

function closeDB() {
    global $mysqli;
    $mysqli->close();
}

function getData($id){
    global $mysqli;
    connectDB();
    if($id)
        $where = "WHERE `id` = $id";
        $result = $mysqli->query("SELECT * FROM `data` ORDER BY `id` DESC");
        closeDB();
    if(!$id)
        return resultToArray ($result);
    else
        return $result->fetch_assoc();
}

function resultToArray($result){
    $array = array ();
    while (($row = $result->fetch_assoc()) != false)
        $array[] = $row;
    return $array;
}
?>

```

process_form.php

```

<?php

$mysqli = false;
function connectDB() {
    global $mysqli;
    $mysqli = new mysqli("localhost", "root", "", "theme_loc_count");
    $mysqli->query("SET NAMES 'utf8'");
}

function closeDB() {
    global $mysqli;
    $mysqli->close();
}

function getData($id){

```

```

global $mysqli;
connectDB();
if($id)
    $where = "WHERE `id` = $id";
    $result = $mysqli->query("SELECT * FROM `data` ORDER BY `id` DESC");
    closeDB();
if(!$id)
    return resultToArray ($result);
else
    return $result->fetch_assoc();
}

function resultToArray($result){
    $array = array ();
    while (($row = $result->fetch_assoc()) != false)
        $array[] = $row;
    return $array;
}
?>

```

index.php

```

<!DOCTYPE html>
<html lang="en">

<?php require_once 'head.php';

    $b0 = -1.54;
    $b1 = 0.97;
    $b2 = 1.05;
    $b3 = 0.19;
    $n = 50;
    $t = 0.2;
    $s2zy = 0.07;
    $cl = $mbc = $dit = '';

    $loc;
?>

<body>
    <div class="main">
        <form id="calculate" method="POST" action="javascript:void(null);"
onsubmit="call()">
            <label>Enter theme name<input class="form_field" name="theme_name"
id="theme_name" type="text" placeholder="Theme Name" required></label>
            <label>Enter the number of Classes<input class="form_field" name="classes"
id="classes" type="text" placeholder="0.00" required></label>
            <label>Enter the number of Methods by Class<input class="form_field"
name="mbc" id="mbc" type="text" placeholder="0.00" required></label>

```

```

        <label>Enter the number of Depth of Inheritance Tree<input
class="form_field" name="dit" id="dit" type="text" placeholder="0.00"
required></label>
        <input class="sbmt_btn" type="submit" value="Calculate">
        <a class="sbmt_btn" href="previous_results.php">Previous results</a>
    </form>

    <div class="result">
        <div class="result_wrap">
            <div class="result_title">Lines of Code: </div>
            <div class="result_data">
                <?php
                    if(!is_numeric($cl) || !is_numeric($mbc) || !is_numeric($dit)){
                        echo '<span>Enter numeric data!</span>';
                    } else if (empty($cl) || empty($mbc) || empty($dit)){
                        echo '<span>Enter numeric data!</span>';
                    }else {
                        $cl = floatval($cl);
                        $mbc = floatval($mbc);
                        $dit = floatval($dit);
                        $loc = $b0+$b1*log10($cl)+$b2*log10($mbc)+$b3*log10($dit);
                        $loc = pow(10, $loc);
                        echo $loc = round($loc, 2);
                    }
                ?>
            </div>
        </div>
        <div class="result_wrap">
            <div class="result_title">Confidence Interval: </div>
            <div class="result_data">
                <?php
                    if(!is_numeric($cl) || !is_numeric($mbc) || !is_numeric($dit)){
                        echo '<span>Enter numeric data!</span>';
                    } else if (empty($cl) || empty($mbc) || empty($dit)){
                        echo '<span>Enter numeric data!</span>';
                    }else {
                        $ci_lb = log10($loc)-$t*sqrt($s2zy*(1/$n));
                        $ci_ub = log10($loc)+$t*sqrt($s2zy*(1/$n));
                        $ci_lb = pow(10, $ci_lb);
                        $ci_ub = pow(10, $ci_ub);
                        echo round($ci_lb,2).'-'.round($ci_ub,2);
                    }
                ?>
            </div>
        </div>
        <div class="result_wrap">
            <div class="result_title">Prediction Interval: </div>
            <div class="result_data">
                <?php
                    if(!is_numeric($cl) || !is_numeric($mbc) || !is_numeric($dit)){
                        echo '<span>Enter numeric data!</span>';
                    }
                ?>
            </div>
        </div>
    </div>

```

```

    } else if (empty($cl) || empty($mbc) || empty($dit)){
        echo '<span>Enter numeric data!</span>';
    }else {
        $pi_lb = log10($loc)-$t*sqrt($s2zy*(1+1/$n));
        $pi_ub = log10($loc)+$t*sqrt($s2zy*(1+1/$n));
        $pi_lb = pow(10, $pi_lb);
        $pi_ub = pow(10, $pi_ub);
        echo round($pi_lb,2).'-'.round($pi_ub,2);
    }
    ?>
</div>
</div>
</div>
</div>
</body>

</html>

```

previous_results.php

```

<!DOCTYPE html>
<html lang="en">

<?php require_once 'head.php';
    $data = getData("");
?>

<body>
    <div class='db_container'>
        <div class='db_data_wrap'>
            <?php
                for($i = 0; $i < PHP_INT_MAX; $i++) {
                    echo'<div class="db_data"> <span>Theme name: <span
class="db_data_value">' . htmlspecialchars($data[$i]["theme_name"]) . '</span></br></span>
>
                    <span>Classes: <span
class="db_data_value">' . htmlspecialchars($data[$i]["classes"]) . '</span></br></span>
                    <span>MBC: <span
class="db_data_value">' . htmlspecialchars($data[$i]["mbc"]) . '</span></br></span>
                    <span>DIT: <span
class="db_data_value">' . htmlspecialchars($data[$i]["dit"]) . '</span></br></span>
                    <span>LOC: <span
class="db_data_value">' . htmlspecialchars($data[$i]["loc"]) . '</span></br></span>
                    <span>Conf Interval LB: <span
class="db_data_value">' . htmlspecialchars($data[$i]["conf_interval_lb"]) . '</span></br>
</span>
                    <span>Conf Interval UB: <span
class="db_data_value">' . htmlspecialchars($data[$i]["conf_interval_ub"]) . '</span></br>
</span>

```

```

        <span>Pred Interval LB: <span
class="db_data_value">'.htmlspecialchars($data[$i]["pred_interval_lb"]).'</span></br>
</span>
        <span>Pred Interval LB: <span
class="db_data_value">'.htmlspecialchars($data[$i]["pred_interval_ub"]).'</span></spa
n></div>';
    }
    ?>
</div>
    <a class="sbmt_btn" href="index.php">Back</a>
</div>
</body>
</html>

```

ДОДАТОК В – ОПИС ПРОГРАМИ

Загальні відомості

Програма для оцінювання розміру тем, створених для системи керування вмістом WordPress на мові програмування PHP.

Функціональне призначення

Програма ThemeLOCCount призначена для оцінювання розміру WordPress тем на PHP.

Технічні засоби, що використовуються

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам'ять 2 Гб, простір на жорсткому диску 80 Мб.

Виклик та завантаження

Завантаження програмного забезпечення здійснюється шляхом відкриття файлу index.php. Для цього потрібна будь-яка операційна система, що забезпечує мережеву взаємодію по протоколу TCP/IP та веббраузер з підтримкою HTML5 й підключення до мережі Internet.

Вхідні дані

Набір з трьох метрик WordPress теми, за якими потрібно провести оцінку розміру програмного забезпечення: кількість класів, середня кількість методів на клас та глибина дерева успадкування.

Вихідні дані

Оцінка кількості строк коду програмного забезпечення, значення границь довірчого інтервалу нелінійної регресії та границь інтервалу передбачення нелінійної регресії розміру WordPress теми.

ДОДАТОК Г – ІНСТРУКЦІЯ КОРИСТУВАЧА

Загальний опис роботи з ПЗ

Після відкриття файлу `index.php` на екрані у вікні браузера з'являється форма що відображає початковий стан програми для оцінювання розміру WordPress тем на PHP (рисунок Г.1).

Enter theme name

Theme Name

Enter the number of Classes

0.00

Enter the number of Methods by Class

0.00

Enter the number of Depth of Inheritance Tree

0.00

CALCULATE PREVIOUS RESULTS

Lines of Code: Please enter a value

Confidence Interval: Please enter a value

Prediction Interval: Please enter a value

Рисунок Г.1 – Форма для відображення початкового стану програми

У вікні браузера (рисунок Г.1) у відповідних полях для вводу необхідно ввести дані: назву теми, кількість класів, середню кількість методів на клас та глибину дерева успадкування та натиснути кнопку «Calculate». Після цього результати оцінювання розміру програмного забезпечення будуть виведені на екран (рисунок Г.2). Кнопка «Previous results» перенаправляє користувача на сторінку перегляду збережених в базі даних минулих результатів що зображена на рисунку Г.3.

Enter theme name

Agnar

Enter the number of Classes

137

Enter the number of Methods by Class

4.44

Enter the number of Depth of Inheritance Tree

1.31

CALCULATE **PREVIOUS RESULTS**

Lines of Code: 17.78

Confidence Interval: 12.3 - 25.5

Prediction Interval: 4.89 - 63.67

Рисунок Г.2 – Результат роботи програми

Theme name: Multi Store Classes: 152 MBC: 4.88 DIT: 1.28 LOC: 22.42 Conf Interval LB: 14.45 Conf Interval UB: 32.06 Pred Interval LB: 5.88 Pred Interval UB: 78.34	Theme name: MagazineX Classes: 56 MBC: 3.88 DIT: 1.25 LOC: 7.15 Conf Interval LB: 4.67 Conf Interval UB: 8.68 Pred Interval LB: 1.58 Pred Interval UB: 22.59	Theme name: Dumbo Classes: 26 MBC: 7.65 DIT: 1.47 LOC: 6.3 Conf Interval LB: 5.12 Conf Interval UB: 7.9 Pred Interval LB: 1.81 Pred Interval UB: 22.13	Theme name: Agnar Classes: 137 MBC: 4.44 DIT: 1.31 LOC: 18.63 Conf Interval LB: 12.3 Conf Interval UB: 25.5 Pred Interval LB: 4.89 Pred Interval UB: 63.67	Theme name: Test Classes: 1 MBC: 1 DIT: 1 LOC: 1 Conf Interval LB: 1 Conf Interval UB: 1 Pred Interval LB: 1 Pred Interval UB: 1
--	--	--	--	--

BACK

Рисунок Г.3 – Перегляд збережених в базі даних результатів розрахунків

Кнопка «Back» повертає користувача на головну сторінку програмного забезпечення.

ДОДАТОК Д – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ

Об’єкт випробувань: програма для оцінювання розміру WordPress тем на РНР.

Ціль випробувань: перевірка відповідності програмного забезпечення ThemeLOCCount до вимог технічного завдання та повноти його реалізації.

Склад документації програмного забезпечення:

- технічне завдання;
- текст програми;
- опис програми;
- інструкція користувача;
- програма та методика випробувань.

Технічні вимоги

Програмне забезпечення має експлуатуватись на обладнанні з параметрами не нижчими за наступні: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам’ять 2 Гб, простір на жорсткому диску 80 Мб.

Засоби та порядок випробувань

Технічні засоби, що використовуються під час випробувань програмного забезпечення:

- процесор AMD Ryzen 7 5800H 3.2GHz - 4.4GHz;
- оперативна пам’ять 32 Гб DDR4 3200MHz;
- SSD 1Тб.

Програмні засоби, що використовуються під час випробувань програмного забезпечення:

Програмне забезпечення не пред’являє ніяких вимог до апаратної платформи.

Перелік програмних засобів, які використовувалися під час випробувань програмного забезпечення:

Веббраузери:

- Google Chrome 108.0.5;
- Mozilla FireFox 107.0.1;

– Opera 93.0.45.

Порядок проведення випробувань

Порядок проведення випробувань – це перевірка вимог до функціональних характеристик розробленого програмного продукту та перевірки вимог до програмної документації.

Методи випробувань

Випробування програмного забезпечення виконується представником замовника в присутності представника розробника. Відповідно до опису програмного забезпечення виконується запуск програми ThemeLOCCount. Далі виконується перевірка роботи програмного забезпечення. Виконується перевірка всіх екранних форм програмного забезпечення, а також всіх допустимих для виконання операцій всередині екранних форм згідно з інструкцією користувача. Якщо програмне забезпечення демонструє правильне функціонування згідно з вимогами технічного завдання та представленої програмної документації, то робота з розробки програмного забезпечення вважається виконаною, і виконується впровадження програмного забезпечення до експлуатації. В протилежному випадку програмне забезпечення має бути відправлено на виправлення розробнику.