

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»

«Допущений до захисту»
Завідувач кафедри

«___» _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «магістр»

на тему: «Нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue, та розробка програми для її реалізації»

Виконав: студент групи 6151м

 Конишев О. В.
(підпис, ПІБ)

Керівник роботи:

доцент кафедри ПЗАС, к.т.н., доцент
(посада, науковий ступень, вчене звання)

 Макарова Л.М.
(підпис, ПІБ)

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 _____ Конишеву Олександру Володимировичу
(Прізвище, ім'я, по батькові)

1. Тема роботи: Нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue, та розробка програми для її реалізації

Керівник роботи _____ Макарова Лідія Миколаївна

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 Титульний аркуш, Мета, об'єкт та предмет дослідження, Наукова новизна та практичне значення одержаних результатів, Існуючі моделі для оцінювання розміру веб-застосунків, Способи удосконалення моделі для оцінювання розміру веб-застосунків, Ескізний проект програми, Технічний проект програми, Робочий проект програми, Висновки, Заключний аркуш

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 проекту 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.

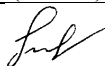
Студент


(підпис)

Конишев О.В.

(ПІБ)

Керівник роботи


(підпис)

Макарова Л.М.

(ПІБ)

РЕФЕРАТ

Конишев Олександр Володимирович

«Нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue, та розробка програми 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 108 стор., 7 табл., 15 рис., 38 використаних джерел, 5 додатків.

Актуальність теми роботи: полягає в тому, що в епоху інформаційних технологій, що швидко розвиваються, постійно зростаючої кількості високобюджетних проектів в області розробки ПЗ, дуже важливим стає вміння оцінити на ранніх етапах можливі вигоди і збитки від проекту, проаналізувати можливі сценарії розвитку подій.

Метою дослідження є підвищення достовірності оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue.

Об'єктом дослідження є процес оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue.

Предметом дослідження є нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue.

Наукова новизна одержаних результатів: удосконалено нелінійну регресійну модель для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue, за рахунок використання нормалізуючого перетворення на основі десяткового логарифму і викидів регресії. Ця нелінійна регресійна модель у порівнянні з однофакторною нелінійною регресійною моделлю має кращі показники якості R^2 , $MMRE$ та $PRED(0,25)$.

Практичне значення одержаних результатів полягає в розробленій програмі для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue, яка дозволить автоматизувати та скоротити час необхідних розрахунків.

Апробація результатів роботи. Основні положення і результати досліджень, викладені у магістерській роботі, пройшли апробацію на V Всеукраїнській науково-практичній інтернет-конференції студентів, аспірантів та молодих вчених «Сучасні комп'ютерні системи та мережі в управлінні» (м. Хмельницький, 30 листопада 2022 р.).

Публікації. Основні результати магістерської роботи викладено у 1 науковій праці – матеріалах конференції.

Ключові слова: нелінійна регресійна модель, веб-застосунок, оцінювання розміру, логарифмічне перетворення, фреймворк Vue.

ABSTRACT

Konyshev Oleksandr Volodymyrovich

«A nonlinear regression model to estimate the size of web applications created using the Vue framework and developing the software for its implementation»

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

The volume of the work is 108 pages, 7 tables, 15 figures, 38 used sources, 5 appendices.

The relevance of the theme is that in the era of rapidly developing information technologies, an ever-increasing number of high-budget projects in the field of software development, it is very important to be able to assess possible benefits and losses from the project at the early stages, to analyze possible scenarios of the development of events.

The purpose of the study is to increase the reliability of estimating the size of web applications created using the Vue framework.

The object of the study is the process of estimating the size of web applications created using the Vue framework.

The subject of the study is a nonlinear regression model for estimating the size of web applications created using the Vue framework.

The scientific novelty of the obtained results: a nonlinear regression model was improved for estimating the size of web applications created using the Vue framework, due to the use of a normalizing transformation based on the decimal logarithm and regression outliers. This non-linear regression model has better R², MMRE and PRED(0.25) quality indicators compared to the univariate non-linear regression model.

The practical value of the obtained results lies in the developed program for estimating the size of web applications created using the Vue framework, which will allow to automate and reduce the time of necessary calculations.

Approbation of work results. The main provisions and results of the research presented in the master's thesis were approved at the 5th All-Ukrainian Scientific and Practical Internet Conference of Students, Postgraduate Students and Young Scientists "Modern Computer Systems and Networks in Management" (Khmelnyskyi, November 30, 2022) .

Publications. The main results of the master's thesis are presented in 1 scientific work – conference materials.

Keywords: nonlinear regression model, web application, size estimation, logarithmic transformation, Vue framework.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП	8
1 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ	11
1.1 Фреймворк Vue та його особливості	11
1.2 Метрики програмного забезпечення для оцінювання його розміру .	14
1.3 Існуючі моделі для оцінювання розміру веб-застосунків	16
2 НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ VUE.....	22
2.1 Способи удосконалення регресійної моделі для оцінювання розміру веб-застосунків.....	22
2.2 Перевірка якості регресійної моделі для оцінювання розміру веб-застосунків.....	26
2.3 Побудова удосконаленої нелінійної регресійної моделі для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue	28
3 ПРОЕКТ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ VUE	36
3.1 Ескізний проект програмного забезпечення.....	36
3.1.1 Постановка задачі на розробку проекту програмного забезпечення для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку VUE.	36
3.1.2 Побудова діаграми варіантів використання	39
3.1.3 Побудова діаграми діяльності.....	42
3.1.4 Розробка прототипу користувацького інтерфейсу	43
3.2 Технічний проект програмного забезпечення	45
3.2.1 Побудова логічної моделі даних	45
3.2.2 Побудова динамічної моделі програми.....	46
3.3 Робочий проект програмного забезпечення.....	48

3.3.1	Визначення набору інструментальних засобів для розробки проекту.....	48
3.3.2	Реалізація та тестування програми	50
4	РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ VUE.....	54
4.1	Розрахунок витрат на створення й використання програмного забезпечення	54
4.2	Економічна ефективність розробки і впровадження програмного забезпечення	57
5	ОХОРОНА ПРАЦІ	59
5.1	Визначення, цілі, завдання та актуальність питань, пов'язаних з охороною праці	59
5.2	Аналіз небезпечних і шкідливих факторів, що впливають на людину при роботі з персональним комп'ютером.....	60
5.3	Розрахунок загального рівномірного освітлення люмінісцентними лампами у виробничому приміщенні для роботи за комп'ютером	64
5.4	Розрахунок загального рівномірного освітлення люмінісцентними лампами у виробничому приміщенні	71
6	ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	75
6.1	Вплив людини на навколишнє середовище	75
	ВИСНОВКИ.....	81
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	82
	ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ.....	87
	ДОДАТОК Б – ОПИС ПРОГРАМИ	90
	ДОДАТОК В – ІНСТРУКЦІЯ КОРИСТУВАЧА.....	91
	ДОДАТОК Г – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ.....	95
	ДОДАТОК Д – ТЕКСТ ПРОГРАМИ	97

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД	База даних
ПЗ	програмне забезпечення
API	Application Programming Interface
COCOMO	Constructive Cost Model
CSS	Cascading Style Sheets
FP	Function points
HTML	HyperText Transfer Protocol
IDE	Integrated development environment
LOC	Lines Of Code
MMRE	Mean of Magnitude of Relative Errors
UI	User Interface

ВСТУП

Актуальність теми. JavaScript – це мультипарадигменна мова програмування, яку використовують для написання як frontend- та backend-частин сайтів, так і ряду мобільних додатків, яку підтримують усі популярні браузери. У frontend-частині сайтів вона використовують для створення інтерактивного контенту на веб-сторінках, у backend-частині за її допомогою, наприклад, розробляють серверні веб-додатки та підключають бібліотеки. У 2022 році JavaScript знову стала найпопулярнішою мовою програмування серед українських розробників з часткою ринку у 18,8% [1].

Як і для багатьох інших мов програмування, для неї було створено кілька фреймворків. JavaScript фреймворки – це інструмент для побудови динамічних веб, мобільних або настільних застосунків на мові JavaScript. Як і з будь-яким іншим інструментом, розробники вдаються до використання фреймворків там, де неможливо або, принаймні, дуже складно і дуже довго виконувати завдання звичайними засобами. У більшості випадків фреймворки використовуються для написання Single-Page Applications, тобто все, що відбувається на сайті, відбувається на одній сторінці без прямого переходу з неї.

На сьогоднішній день існує декілька JavaScript фреймворків. Відповідно до низки опитувань [2, 3], до трійки лідерів для фронтенд розробки незмінно входять React, Angular та Vue [4]. Отже, існує необхідність достовірного оцінювання розміру веб-застосунків, що створюються саме з використанням фреймворку Vue.

Актуальність задачі також полягає в тому, що в епоху інформаційних технологій, що швидко розвиваються, постійно зростаючої кількості високобюджетних проектів в області розробки ПЗ, дуже важливим стає вміння

оцінити на ранніх етапах можливі вигоди і збитки від проекту, проаналізувати можливі сценарії розвитку подій [5].

Метою дослідження є підвищення достовірності оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue.

Для досягнення поставленої мети необхідно вирішити наступні **завдання**:

- проаналізувати існуючі моделі для оцінювання розміру веб-застосунків;
- обґрунтувати необхідність удосконалення регресійної моделі для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue;
- побудувати удосконалену нелінійну регресійну модель для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue, використовуючи нормалізуюче перетворення;
- перевірити якість побудованої нелінійної регресійної моделі для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue;
- розробити програму для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue, використовуючи побудовану удосконалену нелінійну регресійну модель.

Об'єктом дослідження є процес оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue.

Предметом дослідження є нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue.

Методи дослідження. Для вирішення поставлених завдань були застосовані методи теорії ймовірностей та математичної статистики, математичного моделювання, регресійного аналізу, об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів: удосконалено нелінійну регресійну модель для оцінювання розміру веб-застосунків, що створюються з

використанням фреймворку Vue, за рахунок використання нормалізуючого перетворення на основі десяткового логарифму і викидів регресії. Ця нелінійна регресійна модель у порівнянні з однофакторною нелінійною регресійною моделлю має кращі показники якості R^2 , $MMRE$ та $PRED(0,25)$.

Практичне значення одержаних результатів полягає в розробленій програмі для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue, яка дозволить автоматизувати та скоротити час необхідних розрахунків.

Особистий внесок здобувача. Магістерська робота є самостійно виконаною працею. Усі результати, викладені у роботі, отримані автором особисто. У праці, яка опублікована у співавторстві з керівником магістерської роботи [6], здобувачеві належить: розробка нелінійної регресійної моделі для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue.

Апробація результатів роботи. Основні положення і результати досліджень, викладені у магістерській роботі, пройшли апробацію на V Всеукраїнській науково-практичній інтернет-конференції студентів, аспірантів та молодих вчених «Сучасні комп'ютерні системи та мережі в управлінні» (м. Хмельницький, 30 листопада 2022 р.).

Публікації. Основні результати магістерської роботи викладено у 1 науковій праці – матеріалах конференції.

1 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ

1.1 Фреймворк Vue та його особливості

Як вже було сказано у вступі, на сьогодні JavaScript є популярною мовою програмування, для якої доступно декілька фреймворків, які дозволяють користувачам створювати масштабовані веб-застосунки за допомогою стандартних інструментів [3]. На відміну від JavaScript-коду бібліотек JavaScript, JavaScript-фреймворки визначають дизайн веб-застосунків, викликають JS-бібліотеку та використовують код у ній для завершення застосунку.

JavaScript-фреймворки дуже популярні. Вони є ефективними, безпечними, вільно розповсюдженими (безкоштовними), забезпечують підтримку громади, підтримують можливість повторного використання та швидкого розвитку.

Існує багато JS-фреймворків, доступних для розробникам для створення застосунків, один з найпопулярніших для фронтенд-розробки – це Vue [4].

Фреймворк Vue був розроблений у 2014 році Еваном Ю та використовував ліцензію MIT. Цей фреймворк з відкритим вихідним кодом, який допомагає створювати інтуїтивно зрозумілі користувацькі інтерфейси. Він поєднує в собі популярні функції Angular та React: використовує шаблони Angular та прив'язку даних, а також реквізити React.

У 2020 році фреймворк Vue став одним з найбільш зоряних фреймворків на github [7], випередивши Angular та React. Крім того, в останні роки інтерес до фреймворку зріс на 18-20%. Його використовують StackOverflow, Gitlab, Adobe,

Xiaomi, Alibaba та Reuters, а також багато інших компаній. Цей фреймворк має власну сторінку на GitHub [8]

Фреймворк Vue не приніс чогось нового, він взяв уже здобутий багаж знань від React/AngularJS/Angular та інших інструментів, видалив роздутий і складний API і привів все до мінімалістичного вигляду. Користувачам сподобалось в цьому фреймворку багато чого: від низького порогу входу до швидкості розробки застосунків.

Із особливостей фреймворку Vue можна виділити наступне:

- two-way data binding (подобно Angular);
- SSR, як коробочне рішення є Nuxt.js;
- компонентний підхід (компоненти мають власний життєвий цикл, що схоже з концепцією React);
- State container – Vuex (аналог Redux для React);
- CLI – command line interface, або набір утіліт для розробки застосунків;
- можливість використовувати JSX для написання компонентів;
- малий розмір при зборці у порівнянні з React та Angular;
- забезпечення інтерактивності;
- зв'язування DOM з об'єктними даними за допомогою HTML-шаблонів;
- вбудовані CSS-переходи та анімації;
- прив'язка даних за допомогою v-bind;
- шаблони на основі HTML;
- простий синтаксис та інтеграція;
- Vue-маршрутизатор (виконує навігацію між сторінками).

Можна відмітити наступні переваги фреймворку Vue.

Простота – це величезна перевага фреймворку, що дозволяє розробникам досягти гарних результатів при необхідності кодування лише кілька рядків.

Однофайлові компоненти можуть зберігати всі коди в одному файлі і вимагають відносно невеликих витрат.

Vue.js можна легко інтегрувати в інші фреймворки, такі як React.

Зручний і простий у освоєнні, оскільки програмісти повинні знати лише основи CSS, HTML та JavaScript.

Фреймворк Vue можна використовувати зі звичайними редакторами. Усі його функції легко доступні, і додатки можуть бути налаштовані відповідно до конкретних потреб. Забезпечує більшу гнучкість та меншу кількість обмежень. Хороша документація.

Велика спільнота, яка пропонує підтримку у навчанні та часто публікує оновлену інформацію [9].

Він добре підходить для створення повноцінних Single-Page Application (SPA), адже має повний набір функціоналу.

Слід також відмітити недоліки фреймворку Vue.

Під час формування та розвитку цього фреймворку розробники скаржилися на невелику спільноту та відсутність інструкцій для написання масштабних застосунків. Досягнувши певної LOC та складної бізнес-логіки, вони не знали, куди і як рухатися далі. У той же час додаток починав дублювати код і втрачати уніфікацію.

Ще один недолік – компонентний підхід не досягає до конкурентів, таких, як React по гнучкості. Але це не недолік у Vue.js, оскільки це фреймворк, який забезпечує інструменти для повного циклу розробки застосунків, тоді як React – це бібліотека для рендерингу, яка в основному зосереджується на компонентному підході [10].

1.2 Метрики програмного забезпечення для оцінювання його розміру

В стандарті IEEE 1061 термін «метрика програмного забезпечення» визначається як показник якості ПЗ, і тлумачиться таким чином: метрика якості програмного забезпечення – це функція, яка відображає програмний блок в числовому значенні [11].

Різні групи метрик ПЗ враховують різні його характеристики, наприклад, статичні метрики ПЗ дають уявлення про дизайн, саму програму або документацію і враховують такі фактори, як складність та простота обслуговування. У свою чергу динамічні метрики ПЗ дають уявлення про оцінку продуктивності та надійності ПЗ, зокрема необхідний час виконання або кількість помилок. Відмінності між парадигмою розробки ПЗ, а саме: процедурною або об'єктно-орієнтованою, враховуються через подальшу диференціацію метрик ПЗ у звичайні та об'єктно-орієнтовані.

Тобто метрика ПЗ – це кількісна міра, яка дозволяє отримати чітко виражене чисельне значення відповідної специфікації ПЗ. Кількісні міри дають змогу порівнювати різні проєкти між собою та також використовуються при розробці ПЗ, в тому числі для оцінювання його розміру.

Оцінювання розміру ПЗ дуже важливо на всіх етапах його розробки, особливо на стадії ескізного проектування. LOC – кількість рядків коду ПЗ – одна із основних метрик, що використовуються для оцінювання. Ця метрика також використовується і для оцінювання трудомісткості при розробці ПЗ, зокрема, у відомій моделі COSOMO II.

Ця метрика дуже залежна від синтаксису конкретної мови програмування, адже або декілька команд можуть бути записані в один рядок, або одну команду можна записати у декілька рядків. Тому в якості модифікації метрики LOC додатково розділяють фізичні та логічні рядки коду. Окремо ще виділяють

порожні рядки та рядки коментарів. Додатково до перелічених можна відмітити похідні метрики, які можна розрахувати безпосередньо з перелічених, а саме: відсоток коментарів у тексті програми, середню кількість рядків для функцій, класів, модулів або файлів у проєкті та інші.

Додатково у метрики LOC є ще ряд переваг, зокрема, можливість отримати необхідні дані за допомогою інтегрованих середовищ розробки (IDE) або спеціалізованих програм.

Іншою метрикою, яка використовується для оцінювання розміру ПЗ є функціональні точки FP. Аналіз за допомогою цієї метрики ґрунтується на вимірюванні розміру ПЗ з точки зору користувача ПЗ, при чому використовується функціональна модель ПЗ. Базовою одиницею виміру, на якій ґрунтується даний метод, є функціональна точка [12]. Однак використання цієї метрики має суттєвий недолік. Крім необхідності детального вивчення специфікації вимог до ПЗ та підрахунку всіх вхідних та вихідних елементів, а саме: файлів, транзакцій і таке інше, що само по собі може бути дуже трудомістким, ця метрика не залежить від конкретної мови програмування та не враховує її особливості та специфіку розробки ПЗ. Крім того, у початковому варіанті цього методу не було передбачено його використання для застосування з ПЗ, яке розробляється на основі об'єктно-орієнтованого підходу. Подальша модифікація метрики FP для застосування з об'єктно-орієнтованим підходом розробки ПЗ показала її більшу зручність для оцінювання проєктів з великою кількістю компонентів з візуальною складовою. У проєктах з високою алгоритмічною складністю підрахунок FP може бути ускладнений.

1.3 Існуючі моделі для оцінювання розміру веб-застосунків

Усі моделі оцінювання розміру веб-застосунків можна розділити на декілька основних категорій: моделі на основі експертних оцінок, моделі на основі функціональних точок, параметричні моделі. Але більш розповсюдженими на теперішній час є регресійні моделі для оцінювання розміру веб-застосунків [13].

В загальному вигляді регресійна модель для оцінювання розміру веб-застосунків може бути описана таким рівнянням:

$$y = \hat{y} + \varepsilon = f(\mathbf{x}) + \varepsilon, \quad (1.1)$$

де y – результативна ознака, або залежна змінна;

f – функція, що визначає вид рівняння регресії;

$\mathbf{x}$ – вектор незалежних змінних або факторів, $\mathbf{x} = \{x_1, x_2, \dots, x_k\}$;

ε – випадкова похибка, розподіл якої підпорядковується гаусівському розподілу.

У випадку однієї змінної x однофакторна регресійна модель може бути описана таким рівнянням:

$$y = \hat{y} + \varepsilon = f(x) + \varepsilon. \quad (1.2)$$

У випадку двох змінних x_1 та x_2 двохфакторна регресійна модель може бути описана таким рівнянням:

$$y = \hat{y} + \varepsilon = f(x_1, x_2) + \varepsilon. \quad (1.3)$$

У випадку побудови двохфакторної регресійної моделі (1.3) спершу потрібно перевірити майбутні фактори на наявність мультиколінеарності між ними. Зробити цю перевірку можна за допомогою коефіцієнтів впливу дисперсії *VIFs*. У разі лінійної моделі множинної регресії з k факторами $X_i, i = 1, \dots, k$, *VIFs* – це діагональні елементи оберненої кореляційної матриці $k \times k$ [14]. Якщо значення *VIFs* > 10 – такі дані мають проблеми з мультиколінеарністю. Якщо значення *VIFs* знаходяться у діапазоні [1, 5] – такі дані не мають проблем з мультиколінеарністю.

У разі, коли можливо використовувати лінійну регресійну модель, рівняння (1.1) може бути замінено таким рівнянням:

$$Z_y = \hat{Z}_y + \varepsilon = \bar{Z}_y + (\mathbf{Z}_x^+)^T \hat{\mathbf{b}} + \varepsilon, \quad (1.4)$$

де Z_y – результативна ознака, або залежна змінна;

ε – випадкова похибка, розподіл якої підпорядковується гаусівському розподілу;

$\mathbf{Z}_x^+$ – матриця центрованих регресорів, $\mathbf{Z}_x^+ = \{Z_1 - \bar{Z}_1, Z_2 - \bar{Z}_2, \dots, Z_k - \bar{Z}_k\}$;

$\hat{\mathbf{b}}$ – оцінка вектору параметрів рівняння лінійної регресії, $\mathbf{b} = \{b_1, b_2, \dots, b_k\}^T$.

У випадку однієї змінної Z_1 однофакторна лінійна регресійна модель може бути описана таким рівнянням:

$$Z_y = \hat{Z}_y + \varepsilon = b_0 + b_1 Z_1 + \varepsilon. \quad (1.5)$$

У випадку двох змінних Z_1 та Z_2 двохфакторна лінійна регресійна модель може бути описана таким рівнянням:

$$Z_y = \hat{Z}_y + \varepsilon = b_0 + b_1 Z_1 + b_2 Z_2 + \varepsilon. \quad (1.6)$$

Однак у випадку використання саме лінійної регресійної моделі потрібно перевірити випадкову похибку ε на відповідність гаусівському закону розподілу. Зробити це можна, зокрема, за допомогою критерію згоди χ^2 Пірсона [15].

Критерій χ^2 можна використовувати для перевірки гіпотез різних розподілів (рівномірного, нормального або іншого), отже, він дозволяє порівнювати розподіл частот незалежно від того, розподілені вони нормально чи ні. У цьому полягає його перевага [16].

Розрахувати критерій згоди χ^2 Пірсона можна наступним чином:

$$\chi^2 = \sum_{i=1}^m \frac{(n_i - np_i)^2}{np_i}, \quad (1.7)$$

де m – кількість підінтервалів, що залежить на від обсягу вибірки n та може бути розрахована, зокрема, за формулою Старджеса:

$$m = 3,31 \cdot \lg n + 1,$$

n_i – кількість значень випадкової величини, що потрапили в i -й підінтервал;

p_i – ймовірність потрапляння випадкової величини в i -й підінтервал відповідно до гіпотетичного закону розподілу випадкової величини.

Задавшись потрібним рівнем значимості α та розрахувавши кількості ступенів вільності ν :

$$\nu = m - k - 1,$$

де k – кількість параметрів, від яких залежить закон розподілу (для гаусівського закону розподілу $k = 2$),

за таблицею верхніх $100\alpha\%$ точок розподілу χ^2 знаходять значення $\chi^2_{кр}$. Якщо $\chi^2 \leq \chi^2_{кр}$, можна стверджувати з ймовірністю $1-\alpha$, що закон розподілу випадкової похибки ε відповідає гаусівському закону розподілу.

Однак, при побудові регресійної моделі для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue, стикнулися з ситуацією, коли вихідні емпіричні дані, на яких будується регресійна модель, не відповідають гаусівському закону розподілу, що призвело до необхідності нормалізації вихідних емпіричних даних, для чого можуть використовуватися різні нормалізуючі перетворення, зокрема, перетворення Бокса-Кокса, перетворення Джонсона або інші. У даній роботі використовується десятковий

логарифм як простіше нормалізуюче перетворення, яке може бути описане таким рівнянням:

$$z = \lg(x) . \quad (1.8)$$

Це нормалізуюче перетворення має зворотне до нього, яке може бути описане таким рівнянням:

$$x = 10^z . \quad (1.9)$$

Використання нормалізуючого перетворення (1.8) та зворотного до нього перетворення (1.9) дозволять побудувати відповідні нелінійні регресійні моделі.

У випадку однієї змінної x однофакторна нелінійна регресійна модель з урахуванням (1.5), може бути описана таким рівнянням:

$$Y = 10^{\varepsilon + b_0} X^{b_1} . \quad (1.10)$$

У випадку двох змінних x_1 та x_2 двохфакторна нелінійна регресійна модель з урахуванням (1.6), може бути описана таким рівнянням:

$$Y = 10^{\varepsilon + b_0} X_1^{b_1} X_2^{b_2} . \quad (1.11)$$

Таким чином, рівняння (1.10) є однофакторною нелінійною регресійною моделлю, а рівняння (1.11) – двохфакторною нелінійною регресійною моделлю, які можна застосувати для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue.

При використанні адекватної та ефективної моделі для оцінювання розміру веб-застосунків можна стверджувати про якісну розробку відповідного ПЗ.

2 НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ VUE

2.1 Способи удосконалення регресійної моделі для оцінювання розміру веб-застосунків

Одним із способів удосконалення регресійної моделі для оцінювання розміру веб-застосунків, а також покращення якості регресійної моделі є перевірка вихідних емпіричних даних на викиди. Наявність викидів у вихідному наборі даних може свідчити про невідповідність даних гаусівському закону розподілу та суттєво знизити якість побудованої на цих даних регресійної моделі. Усі знайдені викиди потрібно видалити із вихідного набору даних і після цього можна переходити до наступного кроку – побудови регресійної моделі [17, 18].

Для цього можна використовувати квадрат відстані Махаланобіса (Mahalanobis squared distance), який може бути описаний таким рівнянням:

$$d_i^2 = (Z_i - \bar{Z})^T S_N^{-1} (Z_i - \bar{Z}), \quad (2.1)$$

де $\bar{Z}$ – середній вектор вибірки,

S_N – матриця кореляції вибірки, яка може бути представлена як:

$$S_n = \frac{1}{N} \sum_{i=1}^N (Z_i - \bar{Z}) (Z_i - \bar{Z})^T.$$

Після розрахунку d_i^2 за допомогою (2.1), для знаходження викидів можна використовувати критерій Пірсона χ^2 або критерій Фишера F . В останньому випадку додатково потрібно розрахувати значення T_S (Test statistic). T_S для d_i^2 може бути представлена як:

$$T_S = (N - N_m)Nd_i^2/(N^2 - 1)m, \quad (2.2)$$

яка має розподіл F з m та $N-m$ ступенями свободи.

Значення d_i^2 порівнюється з квантилем розподілу χ^2 , а значення T_S порівнюється з квантилем розподілу F . Точки, для яких ці значення перевищать відповідний квантиль розподілу, є викидами. Процедура ітераційно повторюється до того, поки всі значення не будуть менше або дорівнювати відповідному квантилю розподілу.

Додатково викиди також можна вилучити, використовуючи інтервал передбачення. Усі значення, які потрапляють за межі інтервала передбачення для нелінійної регресії, також є викидами і повинні бути видалені із подальшого аналізу.

У загальному вигляді інтервал передбачення лінійного рівняння регресії може бути представлений як:

$$\hat{Y}_i \pm t_{\alpha/2, v} S_Y \left\{ 1 + \frac{1}{N} + (x^+)^T [(X^+)^T X^+]^{-1} (x^+) \right\}^{1/2}, \quad (2.3)$$

де $\hat{Y}_i$ – розрахункове значення за лінійним рівнянням регресії;

$\mathbf{X}^+$ – матриця центрованих регресорів, яка містить значення $\mathbf{X}^+ = \{X_{1i} - \bar{X}_1, X_{2i} - \bar{X}_2, \dots, X_{ki} - \bar{X}_k\}$;

$$S_Y^2 = \frac{1}{v} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2;$$

$$v = N - k - 1;$$

$(\mathbf{X}^+)^T \mathbf{X}^+$ – матриця розміру $k \times k$, яка може бути представлена як:

$$(\mathbf{X}^+)^T \mathbf{X}^+ = \begin{pmatrix} S_{X_1 X_1} & S_{X_1 X_2} & \dots & S_{X_1 X_k} \\ S_{X_1 X_2} & S_{X_2 X_2} & \dots & S_{X_2 X_k} \\ \dots & \dots & \dots & \dots \\ S_{X_1 X_k} & S_{X_2 X_k} & \dots & S_{X_k X_k} \end{pmatrix},$$

$$\text{де } S_{X_q X_r} = \sum_{i=1}^N [X_{qi} - \bar{X}_q] [X_{ri} - \bar{X}_r];$$

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

У випадку однієї змінної x інтервал передбачення лінійного рівняння регресії може бути представлений як:

$$\hat{Y} \pm t_{\alpha/2, v} S_Y \left(1 + \frac{1}{N} + \frac{(x - \bar{x})^2}{\sum_{i=1}^N (x_i - \bar{x})^2} \right)^{1/2}. \quad (2.4)$$

Також можна додатково побудувати, відповідно до наведеного вище, у загальному вигляді довірчий інтервал лінійного рівняння регресії, який може бути представлений як:

$$\hat{Y}_l \pm t_{\alpha/2, \nu} S_Y \left\{ \frac{1}{N} + (x^+)^T [(X^+)^T X^+]^{-1} (x^+) \right\}^{1/2}. \quad (2.5)$$

У випадку однієї змінної x довірчий інтервал лінійного рівняння регресії може бути представлений як:

$$\hat{Y} \pm t_{\alpha/2, \nu} S_Y \left(\frac{1}{N} + \frac{(x - \bar{x})^2}{\sum_{i=1}^N (x_i - \bar{x})^2} \right)^{1/2}. \quad (2.6)$$

Використовуючи побудований інтервал передбачення лінійного рівняння регресії згідно з (2.3) або (2.4) та зворотнє нормалізуюче перетворення згідно з (1.9), отримаємо інтервал передбачення нелінійного рівняння регресії, який і будемо додатково використовувати для визначення викидів у вихідному наборі даних.

Аналогічно отримаємо і довірчий інтервал нелінійного рівняння регресії: використовуючи побудований довірчий інтервал лінійного рівняння регресії згідно з (2.5) або (2.6) та зворотнє нормалізуюче перетворення згідно з (1.9).

Видаливши із вихідного набору даних викиди за допомогою квадрату відстані Махаланобіса та інтервалу передбачення, тим самим отримаємо більш точну нелінійну регресійну модель для оцінювання розміру веб-застосунків, розроблених з використанням фреймворку Vue.

2.2 Перевірка якості регресійної моделі для оцінювання розміру веб-застосунків

Для перевірки якості нелінійної регресійної моделі використаємо такі параметри, як коефіцієнт детермінації R^2 , середню величину відносної похибки $MMRE$ (Mean of Magnitude of Relative Errors) та рівень прогнозування $PRED(0,25)$.

Коефіцієнт детермінації R^2 може бути представлений як [19]:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (2.7)$$

де y_i – емпіричне значення y ;

$\hat{y}_i$ – розрахункове значення y ;

$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ – середнє значення y .

При значенні $R^2 \geq 0,5$ можна вважати, що дана модель є прийнятною. Достатньо ефективною та результативною можна вважати модель з показником детермінації $R^2 \geq 0,8$. Якщо ж $R^2 = 1$, то лінія регресії точно відповідає усім спостереженням та вимогам, а модель можна вважати адекватною та достовірною.

Середня величина відносної похибки $MMRE$ може бути представлена як:

$$MMRE = \frac{1}{n} \sum_{i=1}^n MRE_i, \quad (2.8)$$

де n – кількість значень у вибірці;

MRE_i – величина відносної похибки, $MRE_i = \left| \frac{y_i - \hat{y}_i}{y_i} \right|$

Прийнятний рівень $MMRE$ має бути не більше 0,25.

Рівень прогнозування $PRED(0,25)$ може бути представлений як:

$$PRED(0,25) = \frac{k}{n}, \quad (2.9)$$

де k – кількість значень з $MRE \leq 0,25$.

Прийнятний рівень $PRED(0,25)$ має бути не менше 0,75.

За допомогою цих параметрів також можна порівнювати побудовані регресійні моделі.

2.3 Побудова удосконаленої нелінійної регресійної моделі для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue

Для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue, однією з основних задач є побудова відповідної математичної моделі, а саме: двохфакторної нелінійної регресійної моделі [20 – 22].

На інформаційному ресурсі GitHub [23] було знайдено 31 веб-застосунок, розроблений з використанням фреймворку Vue, з відкритим вихідним кодом. Було виконано роботу зі збору набору метрик з цих веб-застосунків, у результаті чого отримали такий набір даних:

Y – число рядків коду проєкту у тисячах;

X_1 – кількість компонентів проєкту;

X_2 – кількість сторінок проєкту.

Двохфакторну нелінійну регресійну модель будемо будувати на основі незалежних факторів X_1 та X_2 . Перед побудовою моделі перевіримо майбутні незалежні фактори на наявність мультиколінеарності, використовуючи коефіцієнти впливу дисперсії $VIFs$. У цьому випадку отримаємо таку обернену кореляційну матрицю розміром 2 x 2:

$$\begin{pmatrix} 1,02 & -0,15 \\ -0,15 & 1,02 \end{pmatrix}.$$

Шукані значення знаходяться у діапазоні [1, 5] та дорівнюють 1,02, тобто проблем з мультиколінеарністю незалежних факторів X_1 та X_2 не має.

Вихідні емпіричні дані Y , X_1 та X_2 наведено у табл. 2.1. Оскільки ці дані не підпорядковуються гаусівському закону розподілу, виконаємо нормалізацію

цих даних згідно з (1.8) за допомогою десяткового логарифму. Нормалізовані дані Z_Y , Z_1 та Z_2 також наведено у табл. 2.1.

Перевірку на виявлення викидів серед вихідних емпіричних даних спершу проведемо за допомогою квадрату відстані Махаланобіса. Розраховані значення згідно з (2.1) для d_i^2 та згідно з (2.2) для T_S також наведено у табл. 2.1.

Таблиця 2.1 – Дані веб-застосунків, розроблених з використанням фреймворку Vue

№ п/п	Y	X_1	X_2	Z_Y	Z_1	Z_2	d_i^2	T_S
1	350	173	87	2,5441	2,2380	1,9395	12,22	3,68
2	78	111	91	1,8921	2,0453	1,9590	5,01	1,51
3	135	115	49	2,1303	2,0607	1,6902	6,79	2,05
4	4	40	113	0,6021	1,6021	2,0531	5,07	1,53
5	10,6	148	134	1,0253	2,1703	2,1271	4,55	1,37
6	8	28	66	0,9031	1,4472	1,8195	4,02	1,21
7	11,4	147	92	1,0569	2,1673	1,9638	3,24	0,98
8	5,9	66	73	0,7709	1,8195	1,8633	2,48	0,75
9	12,2	176	48	1,0864	2,2455	1,6812	2,82	0,85
10	8,8	85	42	0,9445	1,9294	1,6232	0,87	0,26
11	6,6	40	35	0,8195	1,6021	1,5441	0,99	0,30
12	6,3	50	51	0,7993	1,6990	1,7076	1,44	0,44
13	3,1	25	16	0,4914	1,3979	1,2041	1,72	0,52
14	7	29	11	0,8451	1,4624	1,0414	1,31	0,40
15	12,1	32	11	1,0828	1,5051	1,0414	1,94	0,58
16	1,9	15	11	0,2788	1,1761	1,0414	3,54	1,07
17	29,4	139	18	1,4683	2,1430	1,2553	1,16	0,35
18	21,3	110	13	1,3284	2,0414	1,1139	1,03	0,31
19	34,4	133	10	1,5366	2,1239	1,0000	2,39	0,72
20	24	114	4	1,3802	2,0569	0,6021	4,83	1,45
21	10,4	81	11	1,0170	1,9085	1,0414	0,87	0,26
22	15,3	89	20	1,1847	1,9494	1,3010	0,16	0,05
23	7,3	60	23	0,8633	1,7782	1,3617	0,27	0,08
24	11	67	10	1,0414	1,8261	1,0000	0,71	0,21
25	12,1	61	7	1,0828	1,7853	0,8451	1,42	0,43
26	6,7	46	11	0,8261	1,6628	1,0414	0,57	0,17
27	26,8	507	25	1,4281	2,7050	1,3979	8,76	2,64
28	14,6	105	13	1,1644	2,0212	1,1139	0,96	0,29
29	7	54	7	0,8451	1,7324	0,8451	1,43	0,43
30	8	61	19	0,9031	1,7853	1,2788	0,17	0,05
31	1,2	6	2	0,0792	0,7782	0,3010	10,24	3,09

Для рівня значимості 0,05 величина квантіля χ^2 дорівнює 7,81, а величина квантіля $F_{m,N-m,\alpha}$ дорівнює 2,95. Як видно із таблиці 2.1, викидами є 1, 27 та 31 набори даних, якщо використовувати квантіль χ^2 , і 1 та 31 набори даних, якщо використовувати квантіль $F_{m,N-m,\alpha}$. Будемо використовувати більш жорсткий критерій – квантіль χ^2 . Видаляємо 1, 27 та 31 набори даних з вибірки та повторюємо перевірку на викиди. У результаті такої ітераційної перевірки за 5 ітерацій було видалено 8 викидів. Таким чином, скоригований набір даних містить 23 проєкти.

Вихідні емпіричні дані Y , X_1 та X_2 , нормалізовані дані Z_Y , Z_1 та Z_2 , а також розраховані значення d_i^2 та T_S для 23 проєктів наведено у табл. 2.2.

Таблиця 2.2 – Дані 23 проєктів веб-застосунків після 5 ітерації

№ п/п	Y	X_1	X_2	Z_Y	Z_1	Z_2	d_i^2	T_S
4	4	40	113	0,6021	1,6021	2,0531	5,42	1,58
5	10,6	148	134	1,0253	2,1703	2,1271	5,09	1,48
7	11,4	147	92	1,0569	2,1673	1,9638	3,87	1,12
8	5,9	66	73	0,7709	1,8195	1,8633	1,94	0,56
9	12,2	176	48	1,0864	2,2455	1,6812	6,10	1,77
10	8,8	85	42	0,9445	1,9294	1,6232	0,65	0,19
11	6,6	40	35	0,8195	1,6021	1,5441	3,40	0,99
12	6,3	50	51	0,7993	1,6990	1,7076	1,99	0,58
13	3,1	25	16	0,4914	1,3979	1,2041	5,33	1,55
14	7	29	11	0,8451	1,4624	1,0414	5,17	1,50
17	29,4	139	18	1,4683	2,1430	1,2553	4,81	1,40
18	21,3	110	13	1,3284	2,0414	1,1139	1,88	0,54
19	34,4	133	10	1,5366	2,1239	1,0000	5,81	1,69
20	24	114	4	1,3802	2,0569	0,6021	4,37	1,27
21	10,4	81	11	1,0170	1,9085	1,0414	1,84	0,53
22	15,3	89	20	1,1847	1,9494	1,3010	1,01	0,29
23	7,3	60	23	0,8633	1,7782	1,3617	0,39	0,11
24	11	67	10	1,0414	1,8261	1,0000	0,71	0,21
25	12,1	61	7	1,0828	1,7853	0,8451	1,47	0,43
26	6,7	46	11	0,8261	1,6628	1,0414	1,66	0,48
28	14,6	105	13	1,1644	2,0212	1,1139	1,10	0,32
29	7	54	7	0,8451	1,7324	0,8451	4,70	1,36
30	8	61	19	0,9031	1,7853	1,2788	0,29	0,08

Для рівня значимості 0,05 величина квантіля χ^2 дорівнює 7,81, а величина квантіля $F_{m,N-m,\alpha}$ дорівнює 3,10. Як видно із таблиці 2.2, викидів немає.

Далі будуємо згідно з (1.6) двохфакторну лінійну регресійну модель для даних, нормалізованих згідно з (1.8) за допомогою десятичного логарифму. Коефіцієнти рівняння лінійної регресії b_0 , b_1 та b_2 знаходимо методом найменших квадратів: $b_0=-0,3625$, $b_1=0,9437$, $b_2=-0,2965$.

Двохфакторна лінійна регресійна модель для нормалізованих даних має такий вигляд:

$$Z_Y = -0,3625 + 0,9437 Z_1 - 0,2965 Z_2 + \varepsilon \quad (2.10)$$

Далі згідно з (1.7) з використанням критерію згоди χ^2 Пірсона перевіряємо випадкову похибку ε на відповідність гаусівському закону розподілу. При заданому рівні значимості $\alpha=0,05$ та кількості ступенів вільності $\nu=2$ за таблицею верхніх $100\alpha\%$ точок розподілу χ^2 знаходимо критичне значення $\chi^2_{кр}$, яке дорівнює 5,99. Розрахункове значення χ^2 дорівнює 1,59. Побудова лінійної регресійної моделі виконана обгрунтовано, адже $\chi^2 < \chi^2_{кр}$.

Також будуємо згідно з (2.3) інтервал передбачення лінійного рівняння регресії, а згідно з (2.5) – довірчий інтервал лінійного рівняння регресії.

Тепер переходимо до вихідних емпіричних даних та будуємо згідно з (1.11) двохфакторну нелінійну регресійну модель. Робимо це на основі вже побудованої двохфакторної лінійної регресійної моделі та зворотного нормалізуючого перетворення (1.9). Двохфакторна нелінійна регресійна модель має вигляд:

$$Y = 10^{\varepsilon - 0,3625} X_1^{0,9437} X_2^{-0,2965} \quad (2.11)$$

Далі будуємо інтервал передбачення нелінійного рівняння регресії та довірчий інтервал нелінійного рівняння регресії, використовуючи вже побудовані аналогічні інтервали для лінійного рівняння регресії.

Аналіз значень нижньої та верхньої границь інтервалу передбачення показав, що усі значення Y не виходять за границі інтервалу передбачення, тобто немає викидів за цим показником.

Перевіримо якість побудованої двохфакторної нелінійної регресійної моделі за такими параметрами: коефіцієнт детермінації R^2 , середня величина відносної похибки $MMRE$, рівень прогнозування $PRED(0,25)$, які розрахуємо згідно з (2.7), (2.8) та (2.9) відповідно. Для остаточної двохфакторної нелінійної регресійної моделі (2.11) R^2 дорівнює 0,9170, $MMRE$ дорівнює 0,1922, $PRED(0,25)$ дорівнює 0,6522.

Хоча існує багато моделей для оцінювання розміру веб-застосунків, що створюються як з використанням різних фреймворків [24, 25], так і без їх допомоги [26 – 29], однак підібрати відповідну модель для порівняння з побудованою двохфакторною нелінійною регресійною моделлю для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue, не вдалося з такої причини. Усі згадані моделі побудовані для об'єктно-орієнтованої методології розробки ПЗ та будуються за допомогою таких метрик, як кількість класів, кількість методів, глибина дерева успадкування та деяких інших об'єктно-орієнтованих метрик. У разі використання фреймворку Vue беруться до уваги інші метрики, такі, як кількість компонентів, кількість сторінок, кількість плагінів та інші, які не відносяться до об'єктно-орієнтованих метриках.

Тому для порівняння були обрані та побудовані такі моделі: двохфакторна лінійна регресійна модель без нормалізації даних та однофакторна нелінійна регресійна модель, де в незалежній змінній X була взята кількість компонентів проєкту.

Згідно з (1.6) була побудована двохфакторна лінійна регресійна модель без нормалізації даних, коефіцієнти рівняння лінійної регресії b_0 , b_1 та b_2 були знайдені методом найменших квадратів: $b_0=-3,1842$, $b_1=0,1510$, $b_2=0,4891$, модель має наступний вигляд:

$$Z_Y = -3,1842 + 0,1510Z_1 + 0,4891Z_2 + \varepsilon$$

Випадкова похибка ε була перевірена згідно з (1.7) за допомогою критерію згоди χ^2 Пірсона на відповідність гаусівському закону розподілу. Перевірка показала, що при рівні значимості $\alpha=0,05$ та кількості ступенів вільності $v=2$ закон розподілу випадкової похибки ε не відповідає гаусівському закону розподілу: при критичному значення $\chi^2_{кр}$, яке дорівнює 5,99, розрахункове значення χ^2 дорівнює 149,82. Отже, цю модель не можна використовувати для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue.

Згідно з (1.10) була побудована однофакторна нелінійна регресійна модель, коефіцієнти рівняння лінійної регресії b_0 та b_1 були знайдені методом найменших квадратів: $b_0=-0,3184$, $b_1=0,7201$, модель має наступний вигляд:

$$Y = 10^{\varepsilon - 0,3184} X_1^{0,7201} \quad (2.12)$$

Випадкова похибка ε була перевірена згідно з (1.7) за допомогою критерію згоди χ^2 Пірсона на відповідність гаусівському закону розподілу. Перевірка показала, що при рівні значимості $\alpha=0,05$ та кількості ступенів вільності $\nu=2$ закон розподілу випадкової похибки ε відповідає гаусівському закону розподілу: при критичному значення $\chi^2_{\text{кр}}$, яке дорівнює 5,99, розрахункове значення χ^2 дорівнює 5,45. Отже, цю модель можна використовувати для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue.

Перевіримо якість побудованої однофакторної нелінійної регресійної моделі за такими параметрами: коефіцієнт детермінації R^2 , середня величина відносної похибки $MMRE$, рівень прогнозування $PRED(0,25)$, які розрахуємо згідно з (2.7), (2.8) та (2.9) відповідно. Для моделі (2.12) R^2 дорівнює 0,7985, $MMRE$ дорівнює 0,3454, $PRED(0,25)$ дорівнює 0,4000.

Представимо параметри якості регресійних моделей (2.11) та (2.12) у таблиці 2.3. Причому, для моделі (2.11) наведемо параметри якості для 1 та 5 ітерацій.

Таблиця 2.3 – Параметри якості регресійних моделей

Параметр	Двохфакторна нелінійна регресійна модель (2.11), 1 ітерація	Двохфакторна нелінійна регресійна модель (2.11), 5 ітерація	Однофакторна нелінійна регресійна модель (2.12)
R^2	0,1791	0,9170	0,7985
$MMRE$	0,6174	0,1922	0,3454
$PRED(0,25)$	0,2258	0,6522	0,4000

Аналіз даних, представлених у табл. 2.3, дозволяє зробити наступні висновки:

1. Двохфакторна нелінійна регресійна модель (2.11) за всіма показниками якості краща, ніж однофакторна нелінійна регресійна модель (2.12): R^2 на 14,84%, $MMRE$ на 44,35%, $PRED(0,25)$ на 63,05%.

2. Включення в модель другого фактору суттєво покращує якість регресійної моделі для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue.

3. Врахування викидів регресії також суттєво покращує якість регресійної моделі. Параметри якості після 5 ітерації, тобто після видалення всіх викидів, кращі за параметри якості після 1 ітерації: R^2 на 412%, $MMRE$ на 68,87%, $PRED(0,25)$ на 189%.

4. Незважаючи на кращі показники якості двухфакторної нелінійної регресійної моделі (2.11), значення $PRED(0,25)$ не відповідає мінімально допустимому значенню не менше 0,75.

Покращити якість двухфакторної нелінійної регресійної моделі для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue, можливо в такий спосіб:

1. Збільшити кількість факторів при побудові нелінійної регресійної моделі.

2. Використати інші нормалізуючі перетворення для побудови нелінійної регресійної моделі.

3. Збільшити кількості веб-застосунків, що створюються з використанням фреймворку Vue для побудови нелінійної регресійної моделі.

3 ПРОЕКТ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ VUE

3.1 Ескізний проект програмного забезпечення

3.1.1 Постановка задачі на розробку проекту програмного забезпечення для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку VUE.

Сучасні програмні продукти обробки інформації є орієнтованими на кінцевого користувача, тому важливою частиною розробки є зручний інтерфейс та отримання очного результату обробки даних. Для створення програми для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue необхідно розглянути існуючі проектні рішення з вибору мови програмування. Застосунок має бути із можливостями вводу вхідних даних, а також містити алгоритми щодо обробки цих даних за допомогою обраних методів.

Оскільки сьогодні існує багато потужних мов програмування, вибір має орієнтуватися на тій, що дієво зможе виконувати складні математичні обчислення. Цій умові найкраще відповідає мова PHP. Але для більш зручної розробки буде використовуватись фреймворк Laravel, що в своїй основі має PHP.

PHP є досить широковикористовуваною, високорівневою мовою програмування. А Laravel у свою чергу є її логічним продовженням та покращеною версією, що й мають робити фреймворки. Також для розробки буде задіяна бібліотека math-php, що дозволить більш гнучко працювати із

математичними обчисленнями. Також варто відмітити одну з найкращих документаций Laravel, що дозволяє досить швидко опанувати його можливості.

Щодо реалізації інтерфейсу добре підійде фреймворк Nuxt.js, що також має дуже гарну документацію та на сьогодні швидко популяризується у сфері ІТ.

Виходячи з аналізу існуючих методів та моделей для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue, сформулюємо постановку задачі. Потрібно створити програму для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue, яка б задовольняла наступним вимогам.

Вимоги до програми.

Вимоги функціональних характеристик.

Вимоги до складу функцій, що виконуються.

Програма має виконувати наступні функції:

1. Вводити з інтерфейсу програми довірчу ймовірність, три метрики, за якими необхідно оцінити розмір ПЗ веб-застосунків, що створюються з використанням Vue в тисячах строк коду: друга X_1 , третя X_2 метрики визначають загальну кількість компонентів та кількість сторінок відповідно.
2. Виконувати обчислення розмірів веб-застосунків, що створюються з використанням фреймворку Vue.
3. Виконувати обчислення довірчого інтервалу нелінійної регресії.
4. Виконувати обчислення інтервалу передбачення нелінійної регресії.

Вимоги до складу й параметрів технічних засобів.

Застосунок має працювати на обладнанні з параметрами не нижчими за наступні: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам'ять 4Гб, простір на жорсткому диску 500 Мб.

Вимоги до інформаційної та програмної сумісності.

Для запуску та роботи з ПЗ необхідна операційна система Windows Хр\7\8\10\11, Ubuntu від версії 18.0.

На підставі цієї постановки задачі розроблено технічне завдання, яке наведено у Додатку А.

Зовнішня специфікація програми

Назва програми: VueSizer

Функції, які виконує програма:

1. Завантажувати до програми файл із метриками, за якими необхідно оцінити розмір веб-застосунка, що створюється з використанням фреймворку Vue в тисячах рядків коду: друга X_1 , третя X_2 , четверта X_3 метрики визначають загальну кількість компонентів, кількість плагінів та кількість сторінок відповідно.
2. Виконувати розрахунки розміру веб-застосунку, що створюється з використанням фреймворку Vue.
3. Виконувати розрахунки довірчого інтервалу нелінійної регресії.
4. Виконувати розрахунки інтервалу передбачення нелінійної регресії.
5. Вводити з інтерфейсу ПЗ фактичний розмір веб-застосунку, що створювався з використанням фреймворку Vue.
6. Виконувати розрахунок відносної похибки.
7. Завантажувати файл із результатами розрахунків.

Вхідні дані:

- Файл з метриками, за якими необхідно оцінити розмір веб-застосунку, в тисячах рядків коду: друга X_1 , третя X_2 метрики визначають загальну кількість компонентів та кількість сторінок відповідно; значення довірчої ймовірності.
- Значення фактичного розміру веб-застосунку, що створюється з використанням Vue.

Вихідні дані:

- Значення розміру веб-застосунка, що створюється з використанням фреймворку Vue, границі довірчого інтервалу та інтервалу передбачення.
- Значення відносної похибки.
- Файл із результатами розрахунків.

Зовнішні ефекти: графічний інтерфейс у браузері у вигляді сторінок сайту. Сторінки можуть перемикатися та маніпулювати даними за допомогою елементів інтерфейсу (завантаження, розрахунок, відображення).

3.1.2 Побудова діаграми варіантів використання

Було виявлено варіанти використання та побудовано їх сценарії, розроблено зовнішню специфікацію програми для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue.

Застосунок передбачає наявність лише одного актора, тобто користувача програми, який буде повноцінно користуватись усіма функціями програми. Виявлені варіанти використання представленні в таблиці 3.1.

Таблиця 3.1 – Варіанти використання

Актор	Варіант використання	Формулювання
1	2	3
Користувач	Завантажити файл із метриками	Завантаження файлу із метриками, за якими необхідно оцінити розмір веб-застосунку, що створюється з використанням Vue.

Продовження табл. 3.1

1	2	3
Користувач	Ввести довірчу ймовірність	Введення довірчої ймовірності, для якої потрібно оцінити довірчий інтервал та інтервал передбачення нелінійної регресії розміру веб-застосунку, що створюється з використанням фреймворку Vue.
Користувач	Отримати розмір веб-застосунка	Отримання розміру веб-застосунка, що створюється з використанням фреймворку Vue.
Користувач	Отримати довірчий інтервал	Отримання довірчого інтервалу нелінійної регресії для оцінювання вибіркового середнього розміру веб-застосунку, що створюється з використанням фреймворку Vue.
Користувач	Отримати інтервал передбачення	Отримання інтервалу передбачення нелінійної регресії для оцінювання вибіркового середнього розміру веб-застосунку, що створюється з використанням фреймворку Vue.
Користувач	Ввести фактичний розмір веб-застосунку	Введення розміру веб-застосунку, що створений з використанням фреймворку Vue, для якого буде проведено оцінювання відносної похибки.
Користувач	Отримати відносну похибку	Отримання відносної похибки для веб-застосунку, що був створений з використанням фреймворку Vue.
Користувач	Завантажити результати	Завантаження результатів обрахунків програми.

На рисунку 3.1 наведено діаграму використання ПЗ для оцінювання веб-застосунків, що створюються з використанням фреймворку Vue.



Рисунок 3.1 – Діаграма варіантів використання

Таким чином, була розроблена діаграма варіантів використання програми і далі можемо переходити до побудови діаграми діяльності.

3.1.3 Побудова діаграми діяльності

На рисунку 3.2 наведено діаграму діяльності, що відображає сценарії використання ПЗ для оцінювання застосунку, що створюється з використанням фреймворку Vue.

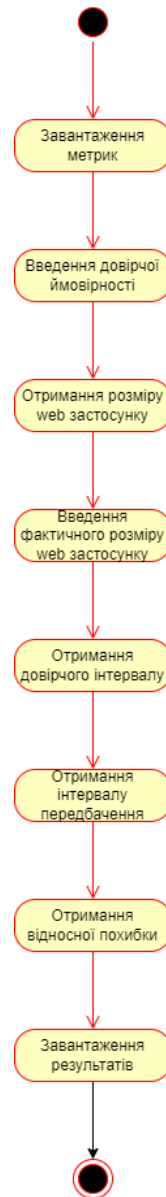


Рисунок 3.2 – Діаграма діяльності

Таким чином, була побудована діаграма діяльності для розроблюваної програми і далі можемо переходити до розробки прототипу користувацького інтерфейсу.

3.1.4 Розробка прототипу користувацького інтерфейсу

Проектування інтерфейсу є важливим кроком розробки ПЗ, оскільки воно допомагає очно оцінити результат та заздалегіть продумати зручність розташування елементів. Для даного етапу роботи було застосовано ПЗ Balsamiq Wireframes. Balsamiq Wireframes – програма для створення каркасних веб-сайтів із графічним інтерфейсом користувача. Дозволяє дизайнеру впорядковувати попередньо створені віджети за допомогою WYSIWYG-редактора з функцією перетягування. Було розроблено ескізи для проекту програмного забезпечення для оцінювання розміру веб-застосунків, що створюються с використанням фреймворку Vue, що наведено на рисунках 3.3 – 3.6.

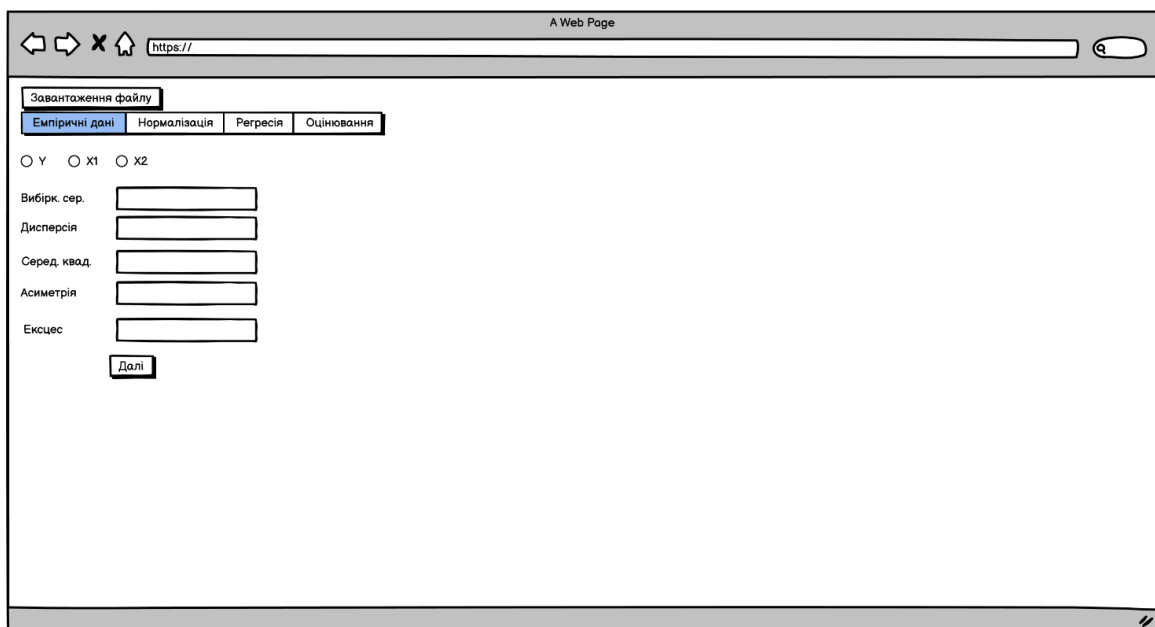


Рис 3.3 – Ескіз вкладки з емпіричними даними застосунку

А Web Page

https://

Емпіричні дані Нормалізація Регресія Оцінювання

☐ Y ☐ X1 ☐ X2

Вибірк. сер.

Дисперсія

Серед. квад.

Асиметрія

Ексцес

Далі

Рис. 3.4 – Ескіз вкладки нормалізації застосунку

А Web Page

https://

Емпіричні дані Нормалізація Регресія Оцінювання

Лінійне регресія

b0

b1

b2

R2

MMRE

PRED

Нелінійне регресія

Result text

R2

MMRE

PRED

Далі

Рис. 3.5 – Ескіз вкладки регресії застосунку

Рис. 3.6 – Ескіз вкладки оцінювання застосунку

Таким чином, розроблений прототип користувацького інтерфейсу буде зрозумілим для користувачів застосунку.

3.2 Технічний проект програмного забезпечення

3.2.1 Побудова логічної моделі даних

У технічному проекті програми для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue, було розроблено діаграму «сутність – зв’язок», що представлено на рисунку 3.7, яка демонструє модель даних.

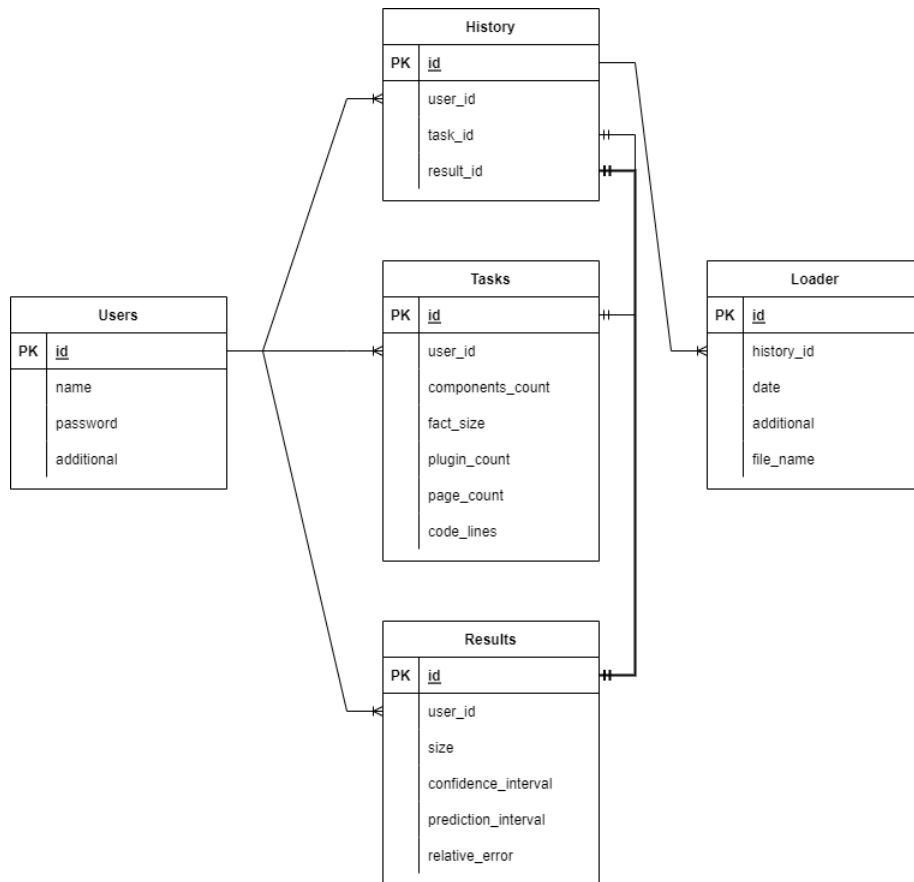


Рисунок 3.7 – Діаграма «Сутність – зв’язок»

Тож кожній таблиці буде відповідати своя логічна модель із властивостями. Також для кожної моделі буде відповідний контролер, що відповідатиме за керування логікою та керування роботи із базою.

3.2.2 Побудова динамічної моделі програми

Для відображення сутності моделі програми було розроблено діаграму послідовності, що демонструє взаємодії об’єктів впорядкованих за часом та послідовність надісланих повідомлень (рисунок 3.8).

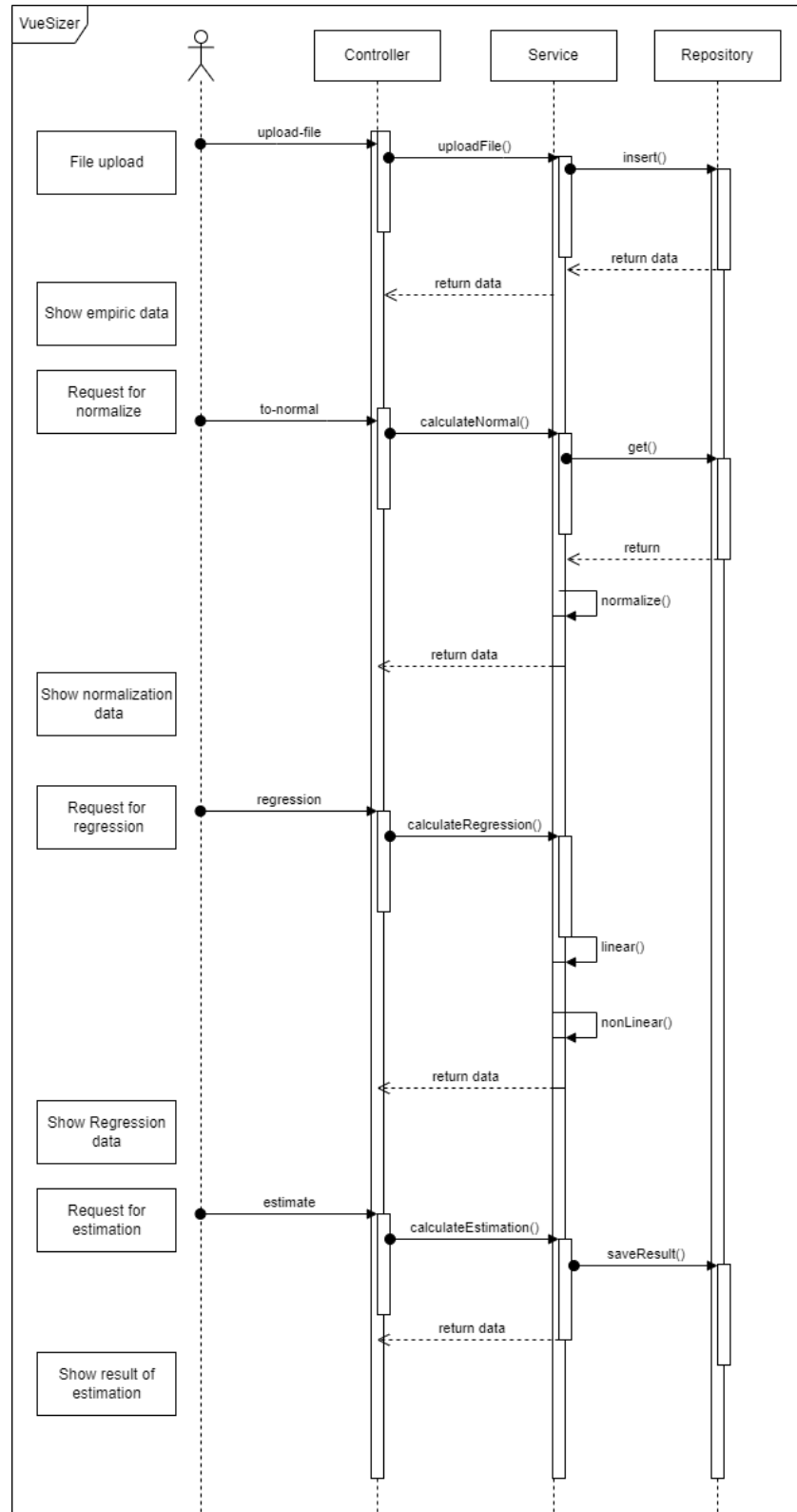


Рисунок 3.8 – Динамічна модель програми

Таким чином, можемо переходити до наступного етапу проєктування – робочого проєкту програми.

3.3 Робочий проєкт програмного забезпечення

3.3.1 Визначення набору інструментальних засобів для розробки проєкту

Laravel – безкоштовний PHP-фреймворк з відкритим кодом, для розробки веб-додатків на основі шаблону Model View Controller (MVC). Особливості Laravel: модульна система упаковки із виділеним менеджером залежностей Composer, різними методами доступу до реляційних баз даних, утилітами для розгортання та обслуговування програм, а також зосередженням на синтаксичному цукрі [30].

Nuxt.js – безкоштовний фреймворк на основі Vue.js, що дозволяє створювати графічні інтрефейси для веб-застосунків та відмальовувати їх не через браузер клієнта, а на віддаленому сервері. Також добре підходить для реалізації SPA(Single Page Application) та функцій у реальному часі за допомогою спеціальних бібліотек, наприклад Axios [31].

VS Code – середовище розробки для розробників та різних мов програмування. Застосунок створений компанією Microsoft та поширюється на безкоштовній основі. Дозволяє підтримувати проєкти на різних рівнях складності та мовах програмування.

Postman – зручний HTTP-клієнт для тестування веб-сайтів, входить в число кращих розширень каталогу Chrome Web Store в категорії «інструменти для роботи» (productivity tools). Дозволяє тестувати застосунки до яких ще не було розроблено інтерфейси та випробовувати функціонал віддалених серверів.

Для більш вдалої розробки було обрано шаблон Controller-Service-Repository (рисунок 3.9) [32]. Шаблон чудово справляється з розділенням проблем: рівень контролера, відповідає виключно за надання функціональних можливостей, щоб вони могли використовуватися зовнішніми об'єктами (включаючи, можливо, компонент інтерфейсу користувача). Рівень сховища відповідає за зберігання та отримання певного набору даних. Сервісний рівень – це місце, куди повинна розташовуватися вся бізнес-логіка. Якщо бізнес-логіка вимагає отримання/збереження даних, вона підключається до сховища. Якщо хтось хоче отримати доступ до цієї бізнес-логіки, він проходить через контролер, щоб отримати її. Це досить простий розподіл проблем. Якщо код пов'язаний із зберіганням/вилученням, він має бути в репозитарії. Якщо це пов'язано з розкриттям функціональних можливостей, за це відповідає контролер. Усе, що є унікальним у бізнес-логіці, піде на рівень сервісу. Репозиторію байдуже, який компонент його викликає; воно сліпо виконує те, що його просять. Сервісному рівню байдуже, як до нього отримати доступ, він просто виконує свою роботу, використовуючи там, де потрібно, репозиторій. А контролер просто передає роботу на рівень обслуговування, тому він може залишатися гарним і ефективним.

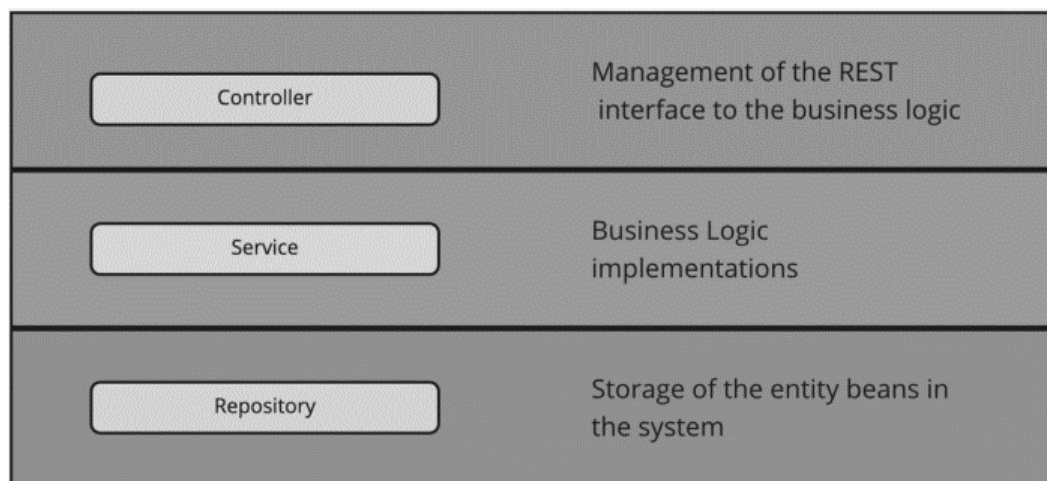


Рисунок 3.9 – Логіка шаблону Controller-Service-Repository

3.3.2 Реалізація та тестування програми

В робочому проекті було розроблено застосунок для оцінювання розміру веб-застосунків, що створюються з використанням Vue, здійснено тестування, розроблено документацію.

Спочатку за допомогою фреймворку Laravel було конструювання бази даних, використовуючи міграції, на стадії технічного проекту. Міграції – це як контроль версій вашої бази даних, що дозволяє вашій команді визначати та надавати спільний доступ до визначення схеми бази даних програми. Допомогає уникнути конфліктів при редагуванні БД та є дуже потужним інструментом при командній розробці.

Надалі було виконано кодування класів, використовуючи шаблон Controller-Service-Repository.

Після розробки логічної частини застосунку необхідно розробити інтерфейс. Для цього було використано фреймворк Nuxt.js, що дозволяє “спілкуватися” із логічною частиною за допомогою запитів. Для запитів було використано спеціальну бібліотеку Axios. Axios – це HTTP-клієнт для JavaScript. Він має можливість робити HTTP-запити з браузера та обробляти перетворення даних запиту та відповіді. Завдяки Axios інтерфейс отримує дані від логічної частини застосунку та вбудовує їх безпосередньо у структуру веб-сторінки. Текст застосунку наведено у в Додатку Д.

У застосунку були використані наступні компоненти: кнопки, рядки редагування та виведення тексту, вкладки. Графічні елементи застосунку наведено на рисунках 3.10 – 3.13.

Завантажити файл

Емпіричні дані Нормалізація Регресія Оцінювання

☐ Y ☒ X₁ ☐ X₂

Вибірк. сер. 83,73913

Дисперсія 1824,110

Серед. квадрат. 42,70960

Асиметрія 0,605812

Екссес 2,7936

Далі

Рис 3.10 – Інтерфейс вкладки “Емпіричні дані”

Емпіричні дані Нормалізація Регресія Оцінювання

☐ Y ☒ X₁ ☐ X₂

Вибірк. сер. 1,865647

Дисперсія 0,054888

Серед. квадрат. 0,234281

Асиметрія -0,214951

Екссес 2,734505

Далі

Рис. 3.11 – Інтерфейс вкладки “Нормалізація”

localhost:3000

Емпіричні дані Нормалізація **Регресія** Оцінювання

Лінійна регресія

b_0 -0,3625

b_1 0,9437

b_2 -0,2965

R^2 0,8442

MMRE 0,0878

PRED 1,0000

Нелінійна регресія

R^2 0,9170

MMRE 0,1922

PRED 0,6522

Далі

Рис. 3.12 – Інтерфейс вкладки “Регресія”

localhost:3000

Емпіричні дані Нормалізація Регресія **Оцінювання**

NC 45

CP 60

KLOC 4,683

Довірчий інтервал 3,7179 5,8975

Інтервал прогнозування 2,6191 8,3706

KLOC емпіричне

MRE

Експортувати результати

Рис. 3.13 – Інтерфейс вкладки “Оцінювання”

Було проведено функціональне тестування застосунку, а саме:

- Перевірка функцій завантаження файлу
- Перевірка функцій вводу довірчої ймовірності
- Перевірка отримання розрахунків
- Перевірка функцій експортування результатів

Всі функції були виконані.

Перевірка застосунку здійснювалась на браузерях Google Chrome, Opera, Fire Fox на операційних системах Windows та Linux. Графічний інтерфейс не змінюється в залежності від середовища запуску, оскільки всі браузери однаково інтерпретують html та css.

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ VUE

Розробка тих чи інших інформаційних технологій пов'язана з капітальними вкладеннями різного характеру. До таких можна віднести придбання техніки, саму розробку проектів, роботи із планування, підготовка фахівців. Тому має сенс економічно обґрунтовувати доцільність розробок інформаційних систем. Необхідно визначити ефективність використання автоматизованих комп'ютерних технологій та систем.

4.1 Розрахунок витрат на створення й використання програмного забезпечення

Витрату на розробку ПЗ складають витрати на заробітню плату розробника, на амортизацію комп'ютера, на якому розроблюється система, на експлуатацію цього комп'ютера, на інструменти розробки та витратні матеріали і комплектуючі частини комп'ютера.

Розробка ПЗ виконується розробником, місячний оклад якого складає 15000 грн. Додаткова заробітна плата складає 20% від основної. На основі цих даних, основна і додаткова плата становлять 18000 грн. Вартість сучасного комп'ютера складає 20000 грн. (Приведена вартість комп'ютера на базі AMD Ryzen 5 5600x).

Вартість розробки ПЗ розраховується за наступною формулою:

$$\text{Спр} = (Ззп + Зсз + Ззг + Зк) * Т + Зм$$

де Т – тривалість розробки, міс.;

Ззп – основна і додаткова заробітня плата персоналу, грн.;

Зсз – відрахування на соціальні заходи (38% від основної і додаткової заробітної плати), грн.;

Ззн – загальногосподарські витрати (10% від основної заробітної плати), грн.;

Зе – витрати на електроенергію;

Зм – витрати на основні і допоміжні матеріали.

Зе при споживаній потужності 0,5 кВт, тривалості роботи на місяць, рівної $22 \cdot 6 = 132$ години, вартості кіловат-години електроенергії 1,68 грн(за рішенням кабінету міністрів від 1 вересня 2021 року) складає:

$$Зе = 132 \cdot 1,68 \cdot 0,5 = 110,88 \text{ грн.}$$

Витрати щодо допоміжних матеріалів приведені в таблиці 4.1.

Таблиця 4.1 – Витрати на допоміжні матеріали

Витрата	Вартість, грн.
Папір	120,00
Картридж принтера	290,00
Непередбачені витрати	300,00
Загальна сума	710,00

Витрати щодо розробки програмного забезпечення наведено в таблиці 4.2.

Таблиця 4.2 – Витрати на розробку системи

Витрата	Кількість\вартість	Одиниця виміру
Тривалість розробки	1	Міс.
Основна та додаткова заробітна плата	18000,00	Грн.
Відрахування на соціальні заходи	6840,00	Грн.
Загальногосподарські витрати	1500,00	Грн.
Витрати на допоміжні матеріали	710,00	Грн.
Витрати на електроенергію	110,88	Грн.

Відповідно до формули, вартість розробки ПЗ оцінюється:

$$\text{Спр} = (18000 + 6840 + 1500 + 110,88) * 1 + 710 = 27160,88 \text{ грн.}$$

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

$$\text{Аок} = 20000 * 0,6 = 12000 \text{ грн.}$$

У масштабах підприємств річні витрати на основні та допоміжні матеріали визначаються у розмірі 5% вартості основного устаткування:

$$\text{Вм} = 20000 * 0,05 = 1000 \text{ грн.}$$

Річний обсяг робіт комп'ютера у годинах визначається у наступний спосіб:

$$\Phi_{\text{м}} = 264,5 * T_{\text{з}}$$

Де $T_{\text{з}}$ – це середнє місячне завантаження устаткування(приблизно 4 години);

265,5 – середня кількість робочих днів у році.

Отже, річний обсяг роботи комп'ютера складає:

$$\Phi_{\text{м}} = 264,5 * 4 = 1058 \text{ годин.}$$

Витрати на електроенергію $З_e$ при 1058 годинах роботи устаткування в рік складуть:

$$З_e = 1058 * 1,68 * 0,5 = 888,72 \text{ грн.}$$

Експлуатаційні витрати для комп'ютера за рік складуть:

$$З_{зр} = 12000 + 1000 + 888,72 = 13888,72 \text{ грн.}$$

Отже, у перший рік витрати на розробку й експлуатацію програмного забезпечення складуть:

$$З_{се} = 27160,88 + 13888,72 = 41049,60 \text{ грн.}$$

4.2 Економічна ефективність розробки і впровадження програмного забезпечення

Основним показником економічної ефективності функціонування системи є підвищення ефективності керування інформацією у вигляді зниження витрат на керування при одночасному збільшенні швидкості і якості одержання необхідного результату.

Крім багатьох інших негативних ефектів, ручна обробка інформації спричиняє наступні негативні економічні ефекти:

- високі витрати на складування паперових документів (сейфи, шафи, папки й ін.);
- підвищені витрати на канцтовари;
- витрати, пов'язані з корегуванням раніше допущених помилок (людський фактор).

До числа основних факторів, що визначають приріст прибутку в зв'язку з впровадженням ПЗ, відносяться:

- підвищення продуктивності праці;
- вивільнення робочого часу.

Також, не піддається прямій грошовій оцінці підвищення оперативності керування, якість одержуваних результатів, поліпшення організації праці і т.д. Обов'язковою умовою визначення економічної ефективності ПЗ є порівнянність усіх показників у часі, за цінами й іншими нормами, використовуваними для визначення показників, за змістом і колом елементів витрат.

Визначимо пряму економічну ефективність, ґрунтуючись на тому, що впровадження ПЗ вивільняє 0,6 працівника (за експертною оцінкою фахівців). Зарплата 0,6 працівника в рік складає:

$$(15000 * 12) * 0,6 = 108000 \text{ грн.}$$

Річний економічний ефект розраховується по формулі:

$$\text{Эгод} = \Delta \text{Сп} = \text{Еп} * k$$

Де $\Delta \text{Сп}$ = вивільнені кошти після впровадження програмного забезпечення мінус експлуатаційні витрати, тобто 94111,28 грн.

Еп – коефіцієнт ефективності(0,6).

k – одноразові витрати на впровадження продукту(41049,60 грн.) .

$$\text{Эгод} = 94111,28 - 41049,60 * 0,6 = 69481,52 \text{ грн.}$$

Строк окупності системи розраховується наступним чином:

$$T = \frac{41049,60}{94111,28} = 0,43$$

Отже можна стверджувати, що строк окупності системи складає приблизно 5 місяців.

5 ОХОРОНА ПРАЦІ

5.1 Визначення, цілі, завдання та актуальність питань, пов'язаних з охороною праці

Охорона праці – це система законодавчих актів соціально-економічних, організаційно-технічних і санітарно-гігієнічних заходів, що забезпечують безпеку, збереження здоров'я і працездатність людини в процесі праці []. Загальне керівництво роботою по охороні праці на підприємстві покладається на директора і головного інженера, а безпосереднє керівництво й організацію цієї роботи здійснює особисто головний інженер, а на великих підприємствах – його заступник по охороні праці. У їхньому підпорядкуванні знаходиться відділ охорони праці, на який покладаються наступні основні функції:

- контроль за дотриманням керівниками цехів, відділів і інших структурних підрозділів діючого законодавства, стандартів, правил і норм по охороні праці; за виконанням розпоряджень контролюючих органів і наказів по підприємству;
- розробка заходів щодо створення здорових і безпечних умов праці в цехах, відділах, на споруджуваних об'єктах;
- проведення вступного інструктажу для людей, які надходять на підприємство і контроль за своєчасним і якісним проведенням інструктажу на робочих місцях;
- участь у роботі комісій з перевірки знань інженерно-технічних працівників в області техніки безпеки і виробничої санітарії, по розслідуванню причин аварій і нещасних випадків, по розгляду проектів будівництва, капітального ремонту цехів і устаткування;

- проведення паспортизації санітарно-технічного стану цехів;
- облік потерпілих від нещасних випадків на виробництві, проведення аналізу і складання звітів про виробничий травматизм, контроль за освоєнням коштів і впровадженням заходів, спрямованих на поліпшення умов праці;
- організація виставок і стендів по охороні праці. Поліпшення умов праці сприяє:
 - підвищенню ефективності виробництва;
 - підтримці працездатності людини на високому рівні і його якнайшвидшому відновленні;
 - скороченню виробничого травматизму;
 - зменшенню професійних захворювань.

5.2 Аналіз небезпечних і шкідливих факторів, що впливають на людину при роботі з персональним комп'ютером

На багатьох заводах і фабриках виробництво пов'язане з постійним впливом на працівників несприятливих умов. Шкідливі та небезпечні виробничі фактори нерозривно пов'язані між собою. ВПФ – це ті фактори, які в результаті свого тривалого або короточасного впливу на людину призводять до погіршення стану його здоров'я або до травми. На виробництвах з такими умовами праці різні нещасні випадки відбуваються досить часто.

ВПФ – це фактори, які, діючи на працівника, знижують його працездатність або призводять до різних захворювань, їх часто ще називають професійними хворобами. Варто зазначити, що межа між цими двома групами факторів досить умовна. При деяких умовах шкідливі виробничі фактори

можуть стати небезпечними. Наприклад, підвищена вологість відноситься до несприятливих умов праці, вона може викликати різні захворювання дихальної системи. Якщо людині доводиться в таких умовах працювати з електричним струмом, то це стає вже занадто небезпечно, а не просто шкідливо.

Всі фактори на будь-якому підприємстві можуть мати різне походження. Часто можна стикатися з несприятливими умовами праці, які виникають з вини керівництва. Це питання потребує особливої уваги з боку перевіряючих органів. Хочеться сподіватися, що велика частина небезпечних факторів має природне походження, і людині просто необхідно вжити всі заходи, щоб їх вплив був мінімальним. Всі шкідливі виробничі фактори ГОСТ поділяє на наступні групи:

- Фізичні
- Хімічні
- Біологічні
- Психофізіологічні, до яких можна віднести важкі та напружені умови праці.

Персональні комп'ютери, периферійні пристрої, інше устаткування (апарати управління, контрольно-вимірювальні прилади, світильники), електропроводи та кабелі за виконанням і ступенем захисту мають відповідати класу зони, мати апаратуру захисту від струму короткого замикання та інших аварійних режимів. Під час монтажу та експлуатації ліній електромережі необхідно повністю унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів, обмежувати застосування проводів з легкозаймистою ізоляцією і, за можливості, застосовувати негорючу ізоляцію. Лінія електромережі для живлення персональних комп'ютерів і периферійних пристроїв виконується як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Не

допускається використовувати нульовий робочий провідник як нульовий захисний провідник. Нульовий захисний провідник прокладається від стійки групового розподільного щита, розподільного пункту до розеток електроживлення. Не допускається підключати на щиті до одного контактного затискача нульовий робочий та нульовий захисний провідники. Площа перерізу нульового робочого та нульового захисного провідника в груповій трипровідній мережі має бути не менше площі перерізу фазового провідника. Усі провідники мають відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників, температурному режиму та типам апаратури захисту. У приміщенні, де одночасно експлуатуються понад п'ять персональних комп'ютерів і периферійних пристроїв, на помітному та доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення. Персональні комп'ютери і периферійні пристрої повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним. Не допускається підключати персональні комп'ютери та периферійні пристрої до звичайної двопровідної електромережі, в тому числі з використанням перехідних пристроїв. Електромережі штепсельних з'єднань та електророзеток для живлення персональних комп'ютерів та периферійних пристроїв потрібно виконувати за магістральною схемою, по 3-6 з'єднань або електророзеток в одному колі. Штепсельні з'єднання та електророзетки для напруги 12В та 42В за своєю конструкцією

мають відрізнятися від штепсельних з'єднань для напруги 127В та 220В. Штепсельні з'єднання та електророзетки, розраховані на напругу 12В та 42В, мають візуально (за кольором) відрізнятися від кольору штепсельних з'єднань, розрахованих на напругу 127В та 220В. Індивідуальні та групові штепсельні з'єднання та електророзетки необхідно монтувати на негорючих або важкогорючих пластинах. Електромережу штепсельних розеток для живлення персональних комп'ютерів і периферійних пристроїв при розташуванні їх уздовж стін приміщення прокладають по підлозі поруч зі стінами приміщення, як правило, в металевих трубах і гнучких металевих рукавах, а також у пластикових коробах і пластмасових рукавах з відводами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. При розміщенні в приміщенні до п'яти персональних комп'ютерів і периферійних пристроїв допускається прокладання трипровідникового захищеного проводу або кабелю в оболонці з негорючого чи важкогорючого матеріалу по периметру приміщення без металевих труб та гнучких металевих рукавів. Не допускається в одній трубі прокладати ланцюги до 42В та вище 42В. При організації робочих місць операторів електромережу штепсельних розеток для живлення персональних комп'ютерів, периферійних пристроїв і у центрі приміщення прокладають у каналах або під знімною підлогою в металевих трубах або гнучких металевих рукавах. При цьому не допускається застосовувати провід і кабель в ізоляції з вулканізованої гуми та інші матеріали, які містять сірку.

5.3 Розрахунок загального рівномірного освітлення люмінісцентними лампами у виробничому приміщенні для роботи за комп'ютером

Приміщення офісу складається з 2 кімнат. Розміри приміщення складають $a \times b \times H = 15,0 \times 25,0 \times 3,0$ м. У приміщенні влаштовано 4 металопластикових вікна (з подвійним склопакетом) розмірами $c \times d = 2,0 \times 2,5$ м. Приміщення зовнішньою стіною спрямовано за азимутом в діапазоні $A_z = 46 \dots 135$ град. Приміщення має сучасний офісний інтер'єр та меблі. Стеля приміщення виконана у вигляді гладкої поверхні пофарбованої у білий колір. Стіни мають гладку поверхню білого кольору із елементами синього. Підлога має покриття із плитки, що імітує паркет світло-сірого кольору. Вікна приміщення обладнані світлозахисними пристроями у вигляді нерегульованих вертикальних жалюзі.

Приміщення (рис. 5.1) нараховує 25 робочих місць, обладнаних сучасними персональними комп'ютерами та мабільними ноутбуками із необхідними периферійними пристроями для роботи, один лазерний принтер, 1 шредер. Для зберігання робочої документації та науково-технічної літератури передбачена відкрита шафа у вигляді полиць спеціального виготовлення. Також за дозволом адміністрації використовується побутова електротехніка (обігрівальна лампа, холодильник, електричний чайник, кавоварка, мікрохвильова піч).



Рисунок 5.1 – Ескіз виробничого приміщення

Напруга джерела живлення електроспоживної техніки – 220 В. Електромережа виконана у вигляді розімкненого типу з дотриманням усіх вимог нормативних документів. За безпекою ураження електричним струмом приміщення відноситься до приміщень без підвищеної небезпеки ураження електричним струмом. Доступ електропостачання забезпечений майже з будь якої точки виробничого приміщення для забезпечення найбільш комфортних умов праці співпрацівників компанії.

Мікрокліматичні умови у літній період (частково у перехідний) забезпечується системою кондиціонування, потужності якої може не вистачати

для забезпечення комфортних умов праці у період активної спеки та підвищеної температури. У зимовий період опалення здійснюється центральною системою, яка цілком задовольняє умови комфортної праці і виконує поставлену задачу на відмінно.

Деякий дискомфорт, пов'язаний із виробничим освітленням, може відчуватися у певні часи робочого дня. Оскільки вікна виробничого приміщення розвернуті до сонячної сторони деякі робочі місця можуть бути не дуже комфортними для роботи із монітором комп'ютеру

Пожежна безпека в обраному виробничому приміщенні відповідає вимогам нормативних документів, зокрема НПАОП 0.00-1.28-10 (Про затвердження правил охорони праці під час експлуатації електронно-обчислювальних машин).

Вентиляція виробничого приміщення виконується за рахунок штучної робочої комбінованої вентеляючої системи офісів. Рівень шуму системи не перевищує норм та не відволікає від роботи. Повітря циркулює із помірною швидкістю. Вентеляційна система є пожежо та вибухобезпечними, мають просте облаштування та надійну конструкцію.

За даними контрольних обстежень, виконанням необхідних вимірів, а також експертних оцінок здійснена оцінка умов праці (за фактором: виробниче освітлення) в обраному виробничому приміщенні. У таблиці 5.1 наведено відомості, що являються вихідними даними для виконання подальших розрахункових робіт даного розділу кваліфікаційної роботи магістра.

Таблиця 5.1 – Вихідні дані для оцінки умов праці у виробничому приміщенні за фактором: виробниче освітлення

№	Параметри	Розмірність	Позначення
1	Довжина приміщення	15	м
2	Ширина приміщення	25	м
3	Висота приміщення	3	м
4	Ширина вікна	2	м
5	Висота вікна	2,5	м
6	Кількість вікон	4	шт.
7	Висота верхнього краю вікна відносно робочої поверхні	3,8	м
8	Відстань розрахункової точки (робочої поверхні) до зовнішньої стіни	2	м
9	Висота карнизу протилежної будівлі відносно підвіконня	1,5	м
10	Відстань до затіняючої будівлі	60	м
11	Орієнтація світлових прорізів (ОСП) за сторонами горизонту	ОСП	—
12	Точність зорових робіт	T	—
13	Вид світлопропускнуго матеріалу	—	—
14	Вид віконної рами	—	—
15	Сонцезахисні пристрої	—	—
16	Стан стелі	—	—
17	Стан стін	—	—
18	Стан поверхні підлоги	—	—
19	Кількість робочих місць	25	шт.

Для перевірного розрахунку необхідно виконати підготовку даних та обрахунок значень.

Нормований коефіцієнт природного освітлення для відповідної (заданої) категорії зорової роботи e_H , %. зорові роботи середньої точності (IV розряд); при цьому $l_{\min} = 0,5 \dots 1$ мм і $e_H = 1,5$ %.

Коефіцієнт світлового клімату m . Для Миколаївської області (як і для решти територій України, окрім Одеської обл. та АР Крим) при орієнтації світлових прорізів у зовнішніх стінах будівель на північ (ПН) $m = 0,9$.

Нормований коефіцієнт природного освітлення для розглянутих умов праці.

$$e_N = e_n \cdot m.$$

e_N – нормований коефіцієнт природного освітлення для розглянутих умов праці;

e_n – нормований коефіцієнт природного освітлення для відповідної (заданої) категорії зорової роботи;

m – коефіцієнт світлового клімату.

Коефіцієнт запасу, що приймається за розрахунками природного освітлення. Для приміщень з нормальними умовами праці (кабінети та робочі приміщення, лабораторії, навчальні приміщення і т. ін.) при боковому освітленні і рекомендованій кількості чищень $n_c = 1$, коефіцієнт запасу складає $\kappa_z = 1,2$.

Геометричні співвідношення, що характеризують виробниче приміщення та розташування робочого місця в ньому: a/b , b/h , l/b .

$$a/b = 1,6$$

$$b/h = 5$$

$$l/b = 0,13$$

a – довжина приміщення;

b – ширина приміщення;

h – висота приміщення;

l – відстань розрахункової точки (робочої поверхні) до зовнішньої стіни.

Світлова характеристика вікна. Виконується за допомогою двумірної лінійної інтерполяції.

$$\eta_s = f(a/b, b/h, l/b)$$

Коефіцієнт світлопропускання матеріалу, коефіцієнт, що враховує втрату світла у віконних рамах, коефіцієнт, що враховує втрати світла у несучих

конструкціях, коефіцієнт, що враховує втрату світла у сонцезахисних пристроях відповідно:

$$\tau_1 = 0.8;$$

$$\tau_2 = 0.78;$$

$$\tau_3 = 0.75;$$

$$\tau_4 = 1;$$

Загальний коефіцієнт світлопропускання. Розраховується на основі попередніх коефіцієнтів.

$$\tau_{заг} = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4$$

τ_1 —коефіцієнт світлопропускання матеріалу;

τ_2 —коефіцієнт, що враховує втрату світла у віконних рамах;

τ_3 —коефіцієнт, що враховує втрати світла у несучих конструкціях;

τ_4 —коефіцієнт, що враховує втрату світла у сонцезахисних пристроях.

Розрахункове значення середньозваженого коефіцієнта відбивання внутрішніх поверхонь виробничого приміщення $\rho_{сер} = 0.3$, оскільки досліджується виробниче приміщення.

Площа підлоги виробничого приміщення

$$S_{підл} = a \cdot b$$

a — довжина приміщення;

b — ширина приміщення.

Коефіцієнт, що враховує підвищення коефіцієнта природного освітлення за рахунок світла, яке відбивається від внутрішніх поверхонь приміщення.

$$r_1 = f(\rho_{сер}, a/b, b/h, l/b)$$

a — довжина приміщення;

b — ширина приміщення;

h — висота приміщення;

l — відстань розрахункової точки (робочої поверхні) до зовнішньої стіни.

$\rho_{сер}$ – розрахункове значення середньозваженого коефіцієнта відбивання внутрішніх поверхонь виробничого приміщення.

Відношення відстані між протилежними будівлями до висоти карнизу протилежного будинку над підвіконням.

$$D/H^I$$

D – відстань до затіняючої будівлі;

H^I – висота карнизу протилежної будівлі відносно підвіконня.

Коефіцієнт, що враховує вплив протилежної будівлі на освітленість у виробничому приміщенні

$$K_{б\gamma\delta} = f(D/H^I)$$

Площа вікон, що необхідна для забезпечення нормованої природної освітленості у виробничому приміщенні

$$S_{\varepsilon} = \frac{e_N \cdot k_z \cdot \eta_{\varepsilon} \cdot S_{\text{підл}} \cdot k_{б\gamma\delta}}{\tau_{\text{заг}} \cdot r_1 \cdot 100}$$

e_N – нормований коефіцієнт природного освітлення для розглянутих умов праці;

k_z – Коефіцієнт запасу, що приймається за розрахунками природного освітлення;

η_{ε} – кількість чищень;

$S_{\text{підл}}$ – площа підлоги;

$k_{б\gamma\delta}$ – коефіцієнт, що враховує вплив протилежної будівлі на освітленість у виробничому приміщенні;

$\tau_{\text{заг}}$ – загальний коефіцієнт світлопропускання;

r_1 – коефіцієнт світлопропускання матеріалу.

Після завершення підрахунків необхідної площі вікон для виробничого приміщення необхідно порівняти її із фактичною площею. Фактична площа складає 5 м², а розрахункова складає 2 м². Виходячи з цього можна стверджувати, що фактична площа відповідає нормованій і навіть перевищує її

2,5 рази. Для запобігання такого перевищення у виробничому приміщенні можна використовувати більш плотні штори або жалюзі.

5.4 Розрахунок загального рівномірного освітлення люмінісцентними лампами у виробничому приміщенні

Оскільки приміщення має гарну природню освітлюватність варто розрахувати систему штучного освітлення, тобто потужність джерел світла. За основу взято метод коефіцієнту використання світлового потоку.

Висота світильника над підлогою. Оскільки стеля є підвісною то зовнішня площа світильника співпадає з площиною стелі(рисунок 5.2).

$$h_0 = H - h_c$$

H – висота стелі;

h_c – зовнішня площа світильника;

h_0 – висота світильника над підлогою.

Висота світильника над робочою поверхнею

$$h = h_0 - h_p$$

$$h = 3,27 \text{ м}$$

h_0 – висота світильника над підлогою;

h_p – висота робочої поверхні.

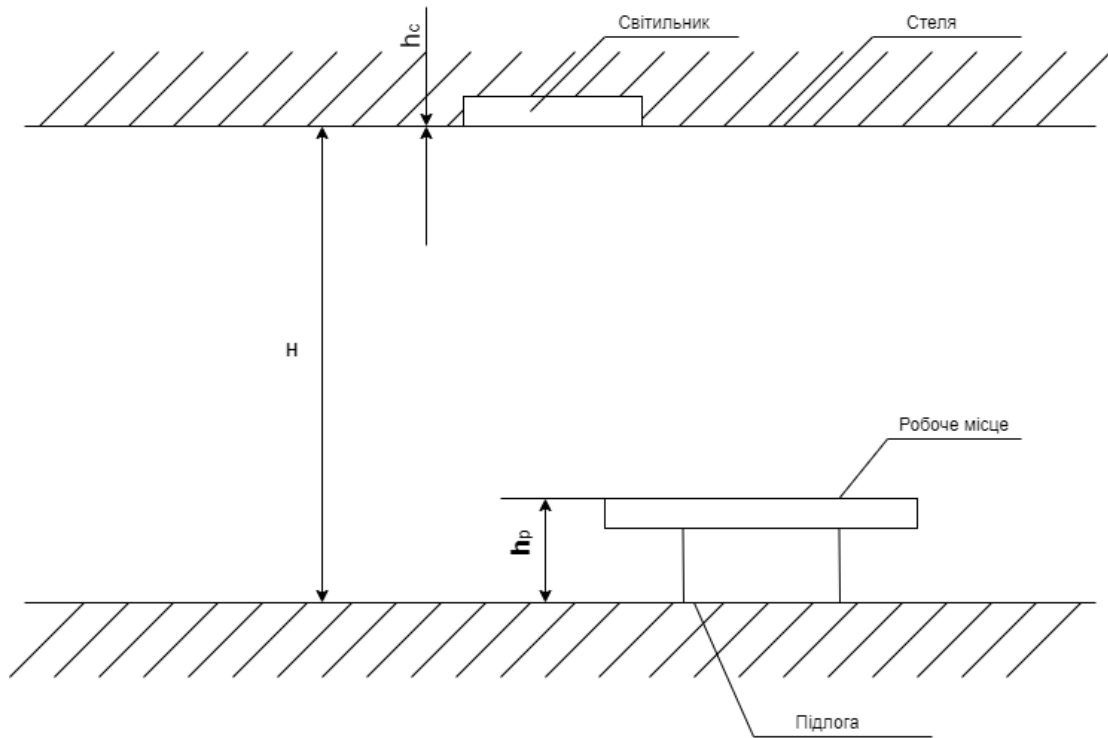


Рисунок 5.2 – Схема для розрахунку штучного освітлення

Коефіцієнт відбиття внутрішніх поверхонь виробничого приміщення.

$$\rho_{\text{стелі}} = 85\%; \rho_{\text{стін}} = 42\%.$$

Осереднене значення коефіцієнтів відбиття стелі та стін.

$$\rho = 0,5 \cdot (\rho_{\text{стелі}} + \rho_{\text{стін}})$$

$$\rho = 45\%$$

Коефіцієнт використання світлового потоку.

$$\eta = f(i, \rho)$$

$$\eta = 64\%$$

i – характеристика приміщення;

ρ – осереднене значення коефіцієнтів відбиття стелі та стін.

Нормоване значення освітленості. $E_H = 200$.

Коефіцієнт запасу, що використовується при розрахунку штучного освітлення. Для приміщень з нормальними умовами праці (кабінети,

лабораторії, та навчальні приміщення і т. ін.) при рекомендованій кількості чищень $n_{\text{ч}} = 1-2$ коефіцієнт запасу складає: $\kappa_3 = 1,4$.

Коефіцієнт нерівномірності світлового потоку. Згідно з рекомендаціями при освітленні люмінесцентними лампами: $Z = 1,1$.

Площа підлоги виробничого приміщення.

$$S_{\text{підл}} = a \cdot b$$

$$S_{\text{підл}} = 375 \text{ м}^2$$

a – довжина приміщення

b – ширина приміщення

Світловий потік однієї лампи $\Phi_{\text{л}} = 2200$ лм.

Необхідна кількість світильників у виробничому приміщенні при кількості n (шт.) ламп у кожному N , шт.

$$N = \frac{E \cdot \kappa_3 \cdot S_{\text{підл}} \cdot Z \cdot 100}{n \cdot \Phi_{\text{л}} \cdot \eta}$$

$N = 10,24$ шт.

E – нормоване значення освітленості;

κ_3 – коефіцієнт запасу;

$S_{\text{підл}}$ – площа підлоги виробничого приміщення;

Z – коефіцієнт нерівномірності світлового потоку;

η – коефіцієнт використання світлового потоку;

$\Phi_{\text{л}}$ – світловий потік однієї лампи;

n – кількість ламп у світильнику.

Оскільки отриманий результат прийнято округляти до більшого числа, а при розташуванні світильників у два ряди до більшого парного числа, то за результатами підрахунків виробничому приміщенню необхідно 12 світильників для забезпечення найкращої штучної освітлюваності робочих місць при умові, що потужність однієї лампи 40 Вт, а кількість ламп у світильнику складає 8 шт

Потужність загального штучного освітлення люмінесцентними лампами у виробничому приміщенні.

$$P_{\text{заг}} = P_{\text{л}} \cdot N \cdot n$$

$P_{\text{л}}$ – потужність однієї лампи світильника;

N – кількість світильників;

n – кількість лам у одному світильнику.

$$P_{\text{заг}} = 3840 \text{ Вт}$$

Після розрахунку необхідної кількості світильників для виробничого приміщення підраховано потужність загального штучного освітлення люмінесцентними лампами, що складає 3840 Вт при умові, що у приміщенні 12 світильників.

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Вплив людини на навколишнє середовище

Охорона навколишнього природного середовища є системою державних і суспільних заходів, направлених на відтворювання збереження і раціональне використання природних ресурсів, поліпшення полягання природного середовища, і є частиною прикладної екології. Відносини у галузі охорони навколишнього природного середовища в Україні регулюються Законом України № 1268-ХІІ від 26.06.91р. «Про охорону навколишнього природного середовища» [38], а також розроблюваними 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. В міру прискорення темпів науково-технічного прогресу дія людей на природу стає все більш потужною. І в даний час воно вже сумарно із дією природних чинників, що приводить до якісної зміни співвідношення сил між суспільством і природою. На сучасному етапі людство поставлено перед чинником виникнення в природі незворотних процесів, нових шляхів переміщення і перетворення енергії та речовини. В природу втручається все більше і більше нових речовин, чужих їй, часом сильно токсичних для організмів. Частина з них не включається в природний колообіг і накопичується в біосфері, що приводить до небажаних екологічних наслідків. Накопичення промислових відходів сприяє підвищенню захворюваності людей і тварин, прискоренню корозії машин і металевого устаткування, зниженню врожайності сільськогосподарських культур і продуктивності тваринництва, прискореному і нераціональному використуванню природних ресурсів і

енергії, погіршенню багатьох властивостей екологічних систем, загибелі деяких унікальних природних територіальних комплексів, зникненню окремих видів тварин і рослин. Основну загрозу для навколишнього середовища в межах галузі розробки програмного забезпечення становлять комплекси електронно-обчислювальних машин (надалі ЕОМ), а саме електромагнітні випромінювання від них. Дослідження останніх років показали, що електромагнітні випромінювання вищезгаданих електронних пристроїв містять торсіонову компоненту, що несе інформацію про процеси, які відбуваються в тому чи іншому електронному пристрої. Торсіонові поля мають високу проникаючу здатність і їх неможливо екранувати. Останніми роками при проектуванні (конструюванні), виготовленні (будівництві) і експлуатації технічних і систем управління в різних сферах діяльності надзвичайно широко застосовуються персональні комп'ютери і всілякі комп'ютерні програми.

Робота з комп'ютерами розробників, операторів і інших користувачів пов'язана з додатковими шкідливими і небезпечними чинниками, що негативно впливають на організм. Наприклад, шкідлива дія на зір надає не при тримання стандартних візуальних ергономічних параметрів екрану, розмір мінімального елементу відображення, мерехтіння зображення, відбивна здатність (відблиски) та ін. Працюючий комп'ютер створює електромагнітне поле, що шкідливо діє на організм людини. Це поле може викликати радіоперешкоди, тобто заважати роботі радіо і телеапаратури, що призводить до зниження надійності технічної системи або системи управління, до збільшення ризиків виникнення аварійної ситуації у виробничому середовищі. Для забезпечення безпеки роботи з комп'ютером розроблені і повинні всюди застосовуватися стандарти на монітори, вимоги до приміщень для експлуатації комп'ютерів, ЕОМ загалом і до організації і устаткування робочих місць.

6.2 Утилізація відходів електронної та комп'ютерної техніки

Використання комп'ютерів вимагає вирішення таких важливих питань, як утилізація відходів (мікросхеми з вмістом кольорових металів, плати, дискети). При утилізації старих комп'ютерів відбувається їх розділ на сім груп: метали, пластмаси, штекери, дроти, батареї, скло. Жодна деталь не йде для повторного використання, оскільки не можна гарантувати їх надійність, але у формі вторинної сировини вони йдуть на виготовлення нових комп'ютерів або інших пристроїв та ЕОМ.

Детально розглянемо декілька прикладів переробки відходів обчислювальної техніки. Гадолінієво-галлеві гранати (ГГГ) використовуються у виробництві компонентів пристроїв, що запам'ятовують. В ході обробки біля 80% вихідного матеріалу перетворюється на відходи або є браком. ГГГ мають високу вартість і їх виділення з відходів представляє інтерес з економічної точки зору.

При здобутті досить чистих продуктів можливі повторне їх використання як вихідного матеріалу. При цьому значно підвищується економічність виробництва заготівель з ГГГ. Під терміном відходи маються на увазі кристалічні залишки, а також дрібний порошок, що виходить при різанні, шліфовці, поліровці кристалів граната або подібних матеріалів. Переробка цих відходів протягом ряду останніх років викликає труднощі і не вирішена до сьогодні. Всі попередні спроби були безуспішними із-за низької розчинності цих складних оксидів.

Вдосконалений процес, розроблений Е.Гуссетом (патент США 4-198231 від 15 квітня 1980 року, фірма «Свісс Алюмініум Лтд.», Швейцарія), призначений для виділення галію і гадолінія з відходів, що містять обидва ці елементи у вигляді оксидів або з'єднань, що перекладаються в оксиди. Відходи

дрібно подрібнюються і потім розчиняються в сильних мінеральних кислотах. Гадоліній 61 осідає з очищених розчинів у вигляді оксалата, галій виділяється в металевому вигляді електролітично. Електролітичне виділення галію може проводитися до виділення гадолінія у вигляді оксалата з розчину.

Розглянемо приклад проведення такого процесу. Відходи піддаються переробці, є залишки завантаження в пристрої для зростання кристалів, роздроблені частини кристалів зі всіх стадій переробки, дрібнозернисті порошки і пудри після операції різання, шліфовки і поліровки гранатів GdGaO . Подрібнений порошок після обробки кристалів ГГГ висушується при 1200 С і потім нагрівається при 6000 С протягом декількох годин для розкладання летких забруднень. Дрібнозернисті відходи в кількості 1000 г розміром менше 40 мкм., що містять 34% галію і 46% гадолінія кип'ятяться із зворотним холодильником протягом двох годин в 2100 мл 35%-ої соляної кислоти. Це відповідає 99% завантаження. Після кип'ятіння частина відходів, що не розчинилася, фільтрується та промивається. Після висушування залишок важить 20 р. Залишки такого типа об'єднуються і піддаються повторній обробці кип'ятінням. Фільтрат, що включає промивні води, об'ємом 2300 мл, обробляється 4000 г металевого галію при 500 С в течії 45 хвилин. Метал має максимально диспергувати. В ході цієї операції благородніші елементи, присутні в розчині, виділяються і частково розчиняються в галії до його насичення і далі охолоджуються у вигляді інтерметалевих включень або в елементарній формі. Відділення можна проводити з меншою кількістю галію. Метал може періодично замінюватися на нову порцію до досягнення необхідної міри очищення. В результаті процесу очищення виходить розчин з вмістом галію 140 г/л і гадолінія 190 г/л.

Встановлюється величина $\text{pH} = 1$ шляхом додавання 900 мл 4%-ного розчину перекису водню для окислення домішок заліза. Осадження гадолінія проводиться при 500 С шляхом додавання 1500 г кристалічної технічної

щавлевої кислоти $\text{COH} \cdot 2\text{HO}$; суспензія акуратно перемішується 12 годин для повторного осадження у вигляді оксалата гадолінія. 62 Оксалат гадолінія $\text{Gd}(\text{CO}) \cdot 10\text{HO}$ відділяється центрифугуванням, промивається 20 мл розбавленої щавлевої кислоти (6 г/л) і висушується при 1300 С; перетворення на оксид гадолінія досягається підігрівом при 8000С. Подальше очищення може проводитися розчиненням в кислоті і осіданням у вигляді оксалата гадолінія. До рідини після центрифугування і промивань осаду (3300 мл) додають 2300 г КОН при інтенсивному перемішуванні до $\text{pH} = 12,6000$ мл розчину з вмістом галію 55 г/л і гадолінія 1 г/л піддається електролізу при 600 Із з використанням катода з нержавеючої сталі і графітового анода при щільності струму близько 0,1 А/кв.см. Після 48 годин осідає 325 г галію і залишається розчин з вмістом 0,4 г/л, що не піддається подальшій переробці. Збагачений метал має чистоту 99,99% і може бути безпосередньо перетворений в оксид. При експлуатації персонального комп'ютера витрачаються наступні ресурси:

- електроенергія;
- папір для принтера;
- картриджи з фарбувальною стрічкою.

Для раціонального використання електроенергії не слід залишати включеним комп'ютер і принтер, якщо відсутня потреба у їх використанні на тепершній момент часу. Друкувати можна з двох сторін. Витрати на папір навряд чи вдасться скоротити удвічі, проте економія буде вельми істотною. Також можна спробувати спеціальну бумагу, що виготовляється з пластику, але є більш дорогою за ціною. Проблему з утилізацією паперових відходів може вирішити вторинна переробка. Для повторного використання картриджів з фарбувальною стрічкою необхідно купити пристрій для просочення стрічок і тоді картриджи можна буде використовувати від 20 до 35 разів.

Сучасна технологія виготовлення елементів засобів обчислювальної техніки (СВТ) дозволяє досягти дуже низького рівня відмов елементів під час

експлуатації. У зв'язку з цим відпадає необхідність проведення ремонтних робіт на місці експлуатації сучасних засобів обчислювальної техніки і як наслідок не утворюються зайві відходи (несправні мікросхеми), що містять дорогоцінні і рідкоземельні метали. Природно, в сервісних центрах, що спеціалізуються на ремонті і технічному обслуговуванні СВТ, має бути організований збір і облік матеріалів, що містять коштовні метали, з подальшою обробкою цих матеріалів на спеціалізованих підприємствах з метою їх вилучення. У зв'язку з тим, що виробництво сучасних компонентів інформаційних технологій сьогодні знаходиться в на початковому рівні, СВТ складаються з парку імпортованих машин та іншого устаткування. Через відсутність інформації про вміст коштовних металів в елементах устаткування, суворий облік не представляється можливим і має бути покладений на фахівців експортних фірм. В умовах ринкової економіки підприємства мають бути самі зацікавлені у вторинній переробці, що містять дорогоцінні метали вузлів і елементів за умови неможливості їх використання. Сумарна вага дорогоцінних металів в комп'ютері типа IBM PC/XT, використовуваному в розробці даної кваліфікаційної роботи складає: золото – 0,22958 г; срібло – 5,091346 г.

Технологічний процес відділення дорогоцінних металів з друкарських плат здійснюється за наступною схемою. Друкарські плати сортуються по переважанню в них кількості дорогоцінних металів, роздроблюються і подрібнюються, обпалюються і плавляться. В процесі випалення перолітичному розкладанню піддається пластмасова основа, а основа дорогоцінних металів у вигляді металевих залишків відновлюється до оксидів. Металевий залишок подрібнюється, гранулюється, обробляється через магнітну сепарацію і відбувається відділення магнітних від немагнітних часток. Отриманий таким чином порошок, розділений по видах дорогоцінних металів, у вигляді гранул розплавляється в індукційних плавильних печах з подальшим розділенням кожного металу окремо.

ВИСНОВКИ

В кваліфікаційній роботі вирішене практичне завдання з підвищення достовірності оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue за допомогою нелінійної регресійної моделі за рахунок використання нормалізуючого перетворення на основі десяткового логарифму і викидів регресії. Ця нелінійна регресійна модель у порівнянні з однофакторною нелінійною регресійною моделлю має кращі показники якості R^2 , $MMRE$ та $PRED(0,25)$.

Проведено аналіз існуючих регресійних моделей для оцінювання розміру веб-застосунків. Було виявлено, що вихідні емпіричні дані, на яких будується регресійна модель, не відповідають гаусівському закону розподілу, що призвело до необхідності нормалізації вихідних емпіричних даних, тож був обран шлях нормалізації через десятковий логарифм.

Здійснено збір набору даних, за якими будується нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue.

На підставі запропонованої нелінійної регресійної моделі розроблено програму для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue. Була розроблена наступна програмна документація: технічне завдання, опис програми, інструкція користувача, методика випробувань та текст програми, що наведені у додатках А, Б, В, Г, Д. Проведено випробування програми згідно з методикою випробувань. За результатами розробки та випробувань сформовано висновок про працездатність розробленої програми для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Vue.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рейтинг мов програмування 2022. С# обійшов Java, TypeScript зрівнявся з PHP, а Dart – найбільш комфортна мова. URL: <https://dou.ua/lenta/articles/language-rating-2022/> (дата звернення: 05.10.2022).
2. Топ 10 самых популярных JavaScript-фреймворков для веб-разработки. URL: <https://vc.ru/dev/147263-populyarnye-freymvorki-javascript> (дата звернення: 05.10.2022).
3. Лучшие JavaScript-фреймворки для использования в 2022 году – полное руководство. URL: <https://appmaster.io/ru/blog/luchshie-javascript-freimvorki-dlia-ispol-zovaniia-v-2022-godu-polnoe-rukovodstvo> (дата звернення: 05.10.2022).
4. The Progressive JavaScript Framework. URL: <https://vuejs.org/> (дата звернення: 05.10.2022).
5. Briand L.C., Morasca S., Basili V.R. Property Based Software Engineering Measurement. *IEEE Transaction on Software Engineering*. 2009. vol. 22, no. 1. pp. 68-86.
6. Конишев О.В., Макарова Л.М. Нелінійна регресійна модель для оцінювання розміру веб застосунків, що створюються з використанням фреймворку Vue. *Матеріали V Всеукраїнської науково-практичної інтернет-конференції студентів, аспірантів та молодих вчених за тематикою «Сучасні комп'ютерні системи та мережі в управлінні»*: збірка наукових праць / Під редакцією А.А. Григорової. Херсон: Видавництво ФОП Вишемирський В.С. 2022. с. 138-139.
7. Frontend Frameworks Popularity. URL: <https://gist.github.com/tkrotoff/b1caa4c3a185629299ec234d2314e190> (дата звернення: 10.10.2022).

8. vuejs / vue. URL: <https://github.com/vuejs/vue> (дата звернення: 10.10.2022).
9. 10 лучших фреймворков для фронтенда и бэкенда. URL: <https://blog.back4app.com/ru/10лучших-фреймворков-для-фронтенда-и-бэ/> (дата звернення: 10.10.2022).
10. Что нужно знать о популярных JS-фреймворках. URL: https://habr.com/ru/company/yandex_praktikum/blog/533702/ (дата звернення: 10.10.2022).
11. 1061-1998-IEEE Standard for a Software Quality Metrics Methodology. URL: <https://ieeexplore.ieee.org/document/749159> (дата звернення: 10.10.2022).
12. IFPUG. Function Point Counting Practices Manual Release 4.0. Int. Function Point Users Group. Westerville, Ohio, 1994. 370 p.
13. Briand L.C., Wieczorek I. Resource Estimation in Software Engineering. Encyclopedia of Software Engineering. John Wiley & Sons, Inc., 2002. 83 p.
14. Chatterjee S., Price B. Regression Analysis by Example. New York: John Wiley & Son, 1977. 228 p.
15. Приходько С.Б., Макарова Л.М., Пугаченко К.С. Методичні вказівки та завдання до виконання лабораторних робіт з дисципліни "Обробка експериментальних даних на комп'ютері". Миколаїв: НУК, 2018. 76 с.
16. Гмурман В.Е. Теория вероятностей и математическая статистика. М.: Высшее образование, 2010. 479 с.
17. Prykhodko, S., Prykhodko, N., Makarova, L., Pugachenko, K., Detecting outliers in multivariate non-Gaussian data on the basis of normalizing transformations, in: *Proceedings of the 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, Kyiv, Ukraine, 2017, 846-849. <https://doi.org/10.1109/UKRCON.2017.8100366>
18. Prykhodko S., Prykhodko N., Makarova L., Pukhalevych A. Outlier Detection in Non-Linear Regression Analysis Based on the Normalizing

Transformations. *Proceedings of the 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), IEEE.* (Lviv-Slavske, 2020). P. 407-410.

19. Магнус Я.Р., Катышев П.К., Пересецкий А.А. Эконометрика. М.: Дело, 2004. 576 с.

20. Грешилов А.А., Стакун В.А., Стакун А.А. Математические методы построения прогнозов. М.: Радио и связь, 1997. 112 с.

21. Демиденко Е.З. Линейная и нелинейная регрессии. М.: Финансы и статистика, 1981. 302 с.

22. Bates D.M., Watts D.G. Nonlinear Regression Analysis and Its Applications. Wiley, 1988. 384 p.

23. The world's leading software development platform. URL: <https://github.com/> (дата звернення: 27.07.2022).

24. Приходько С.Б., Приходько Н.В., Ворона М.В., Беловол І.О. Нелінійна регресійна модель для оцінювання розміру Web-застосунків, що створюються з використанням фреймворку Laravel. *Інформаційні технології та комп'ютерна інженерія*. 2021. Т.50. № 1. С. 115-121. ISSN 1999-9941.

25. Ворона М.В., Приходько А.С., Шутко І.С. Трьохфакторні нелінійні регресійні моделі для оцінювання розміру web-застосунків, що створюються з використанням php-фреймворків. *Інформатика, інформаційні системи та технології: тези доповідей вісімнадцятої всеукраїнської конференції студентів і молодих науковців (Одеса, 23 квітня 2021 р.)*. Одеса, 2021. С. 111-113.

26. Prykhodko S., Prykhodko N., Makarova L. Estimating the Software Size of Open-Source PHP-Based Systems Using Non-Linear Regression Analysis. *Proceedings of International Conference "Advanced Computer Information Technologies" (ACIT-2018)*, (Ceske Budejovice, Czech Republic). CEUR Workshop Proceedings. 2019. Vol. 2300. CEUR-WS.org P.199-202. ISSN 1613-0073.

27. Приходько С.Б., Приходько Н.В., Фаріонова Т.А., Ворона М.В. Трьохфакторна нелінійна регресійна модель для оцінювання розміру Php-застосунків з відкритим кодом. *Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки*. Київ, 2020. № 4 (56). С. 124-131.
28. Kaczmarek J., Kucharski M. Size and effort estimation for applications written in Java. *Information and Software Technology*. 2004. Vol. 46, Issue 9. P. 589–601.
29. Макарова Л.М., Приходько Н.В., Кудін О.О. Побудова нелінійної регресійної моделі для оцінювання розміру веб-додатків, реалізованих мовою Java. *Вісник Херсонського національного технічного університету*. 2019. Вип. 2 (69). С.145-153.
30. Laravel – The PHP Framework For Web Artisans. URL: <https://laravel.com/docs/9.x/lifecycle> (дата звернення: 21.09.2022).
31. Axios usage. Nuxt. URL: <https://nuxtjs.org/examples/modules/axios#axios-usage> (дата звернення: 25.09.2022).
32. Collings T. Controller-Service-Repository. Medium. URL: <https://tom-collings.medium.com/controller-service-repository-16e29a4684e5> (дата звернення: 23.09.2022).
33. Bhargava A. *Grokking Algorithms: An illustrated guide for programmers and other curious people*. Manning Publications, 2016. 256 с.
34. Приходько С.Б., Макарова Л.М., Пухалевич А.В. Методичні рекомендації щодо підготовки звіту з наукового стажування за спеціальністю 121 «Інженерія програмного забезпечення». Миколаїв :НУК, 2021. – 24 с.
35. Діаграми UML для моделювання процесів і архітектури проекту. Evergreen – web розробка і діджиталізація бізнесу за допомогою AI продуктів. URL: <https://evergreens.com.ua/ua/articles/uml-diagrams.html> (дата звернення: 18.10.2022).

36. APServer – MySQL : Зовнішні ключі. APServer.
URL: <http://apserver.org.ua/peregl.php?d=view&tid=91> (дата звернення: 28.09.2022).

37. Закон України «Про охорону праці» від 14.10.1992р № 2694-XII. (зі змінами від 19.08.2022). URL: <https://zakon.rada.gov.ua/laws/show/2694-12#Text>

38. Закон України «Про охорону навколишнього природного середовища» від 26.06.91р № 1268-XII. (зі змінами від 10.07.2022). URL: <https://zakon.rada.gov.ua/laws/main/1264-12#Text>

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

Вступ

Назва розробки – «Програма для оцінювання розміру web застосунків, що створюються з використанням фреймворку Vue», далі застосунок.

Підстави для розробки

Завдання на кваліфікаційну роботу.

Призначення розробки:

Програма призначена для оцінювання розміру web застосунків, що створюються з використанням фреймворку Vue.

Вимоги до застосунку

Вимоги функціональних характеристик.

Вимоги до складу функцій, що виконуються.

Застосунок має виконувати такі функції:

1. Виводити з інтерфейсу застосунку довірчу ймовірність, метрики, за якими необхідно оцінювати розмір ПЗ web застосунків, що створюються з використанням фреймворку Vue в тисячах строк коду.
2. Виконувати обчислення розміру web застосунка, що створюються з використанням фреймворку Vue.
3. Виконувати обчислення довірчого інтервалу нелінійної регресії.
4. Виконувати обчислення інтервалу передбачення нелінійної регресії.

Вимоги до організації вхідних та вихідних даних:

Вхідні дані:

- файл із метриками, за якими потрібно оцінити розмір web застосунку який створюється з використанням фреймворку Vue.
- значення довірчої ймовірності.

Вихідні дані:

– значення нелінійної регресії та границь довірчого інтервалу нелінійної регресії розміру web застосунку, що створюється з використанням фреймворку Vue.

– значення коефіцієнту детермінації.

Вимоги до надійності не висуваються.

Вимогу до умов експлуатації:

– наявність з'єднанням із мережею Інтернет.

Вимоги до складу й параметрів технічних засобів.

Застосунок має бути розгорнутий на обладнанні з параметрами не нижчими за наступні: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам'ять 4Гб, простір на жорсткому диску 150 Мб.

Вимоги до інформаційної та програмної сумісності.

Для запуску та роботи з програмним забезпеченням необхідна операційна система Windows 7/8/10/11, Ubuntu 18/19/20, або Mac OS.

Вимоги до маркування та пакування.

Визначаються вимогами до пакування USB накопичувачів.

Вимоги до транспортування та зберігання.

Програма може зберігатись та транспортуватись на CD дисках або USB накопичувачах.

Вимоги до програмної документації

Програмна документація має включати: технічне завдання на розробку програми, опис програми, інструкція користувача, програма та методика випробувань, текст програми.

Техніко-економічні показники не розраховуються.

Стадії та етапи розробки

Стадії та етапи розробки програми представлені у таблиці А.1.

Таблиця А.1 – Стадії та етапи розробки

Стадія розробки	Етапи роботи	Термін виконання
1. Технічне завдання	Розробка та затвердження технічного завдання.	19.09.2022
2. Ескізний проект	Розробка та затвердження ескізного проекту.	23.09.2022
3. Технічний проект	Розробка технічного проекту.	04.10.2022
	Затвердження технічного проекту.	06.10.2022
4. Робочий проект	Розробка застосунку.	20.10.2022
	Розробка програмної документації.	28.10.2022
	Випробування застосунку.	05.11.2022

Порядок контролю і приймання

Порядок контролю та 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.

ДОДАТОК Б – ОПИС ПРОГРАМИ

Загальні відомості

Застосунок для оцінювання розміру застосунків, що створюються з використанням фреймворку Vue.

Функціональне призначення.

Застосунок призначений для оцінювання розміру web застосунків, що створюються з використанням фреймворку Vue.

Технічні засоби, що використовуюються

Застосунок має експлуатуватись на обладнанні з параметрами не нижчими за наведені: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам'ять 4Гб, простір на жорсткому диску 150Мб.

Виклик та завантаження

Для запуску застосунку його необхідно розгорнути на сервері із Linux подібною операційною системою (наприклад, Ubuntu 18). Для роботи із застосунком необхідно мати браузер та з'єднання із мережею Інтернет.

ДОДАТОК В – ІНСТРУКЦІЯ КОРИСТУВАЧА

Загальний опис роботи із ПЗ

Після завантаження сторінки застосунку з'являється форма для відображення початкової вкладки із даними, як зазначено на рисунку В.1.

The screenshot shows a web application interface for data analysis. At the top, there's a browser address bar showing 'localhost:3000'. Below it is a button labeled 'Завантажити файл' (Load file). The main area has four tabs: 'Емпіричні дані' (Empirical data), 'Нормалізація' (Normalization), 'Регресія' (Regression), and 'Оцінювання' (Evaluation). The 'Емпіричні дані' tab is active. It contains three radio buttons: 'Y', 'X₁' (which is selected), and 'X₂'. Below these are five input fields with the following labels and values: 'Вибірк. сер.' (Sample size) with value 83,73913; 'Дисперсія' (Dispersion) with value 1824,110; 'Серед. квадр.' (Mean square) with value 42,70960; 'Асиметрія' (Asymmetry) with value 0,605812; and 'Ексцес' (Kurtosis) with value 2,7936. At the bottom of the form is a green button labeled 'Далі' (Next).

Рисунок В.1 – Вкладка «Емпіричні дані»

Використовуючи кнопку «Завантаження файлу» необхідно завантажити файл із даними метрик для розразунків. Після заповнення полів можна перемикати наступну вкладку «Нормалізація», що відображає нормалізовані початкові дані (рисунки В.2).

Емпіричні дані Нормалізація Регресія Оцінювання

☐ Y ☒ X₁ ☐ X₂

Вибірк. сер. 1,865647

Дисперсія 0,054888

Серед. квадр. 0,234281

Асиметрія -0,214951

Ексцес 2,734505

Далі

Рисунок В.2 –Вкладка «Нормалізація»

Тим же шляхом треба перейти на вкладку «Регресії», щоб побачити обрахунки нелінійної та лінійної регресії (рисунок В.3).

localhost:3000

Емпіричні дані Нормалізація **Регресія** Оцінювання

Лінійна регресія

b_0 -0,3625

b_1 0,9437

b_2 -0,2965

R^2 0,8442

MMRE 0,0878

PRED 1,0000

Нелінійна регресія

R^2 0,9170

MMRE 0,1922

PRED 0,6522

Далі

Рисунок В.3 – Вкладка «Регресія»

На останній вкладці користувач отримує результати розрахунків, які може як передивлятись так і завантижити у вигляді файлу, а саме загальна кількість компонентів, кількість плагінів, кількість сторінок та число строк коду у тисячах. Також у результатах представляється довірчий та прогнозуючий інтервал із верхньою на нижньою границею (рисунок В.4).

Емпіричні дані	Нормалізація	Регресія	Оцінювання
NC	45		
CP	60		
KLOC	4,683		
Довірчий інтервал	3,7179	5,8975	
Інтервал прогнозування	2,6191	8,3706	
KLOC емпіричне			
MRE			

Експортувати результати

Рисунок В.4 – Вкладка «Оцінювання»

Варто відмітити якщо поле KLOC емпіричне не заповнено, то програма розраховує прогноз по кількості строк та інтервалам. Якщо користувач заповнить поле значенням із справжнього проекту, тоді можливо розрахувати наскільки прогнозоване значення(MRE), що отримано за допомогою побудованої моделі, відрізняється від реального значення по завершеному проекту, що не був доданий до даних побудови моделі(рисунок В.5).

Емпіричні дані	Нормалізація	Регресія	Оцінювання
NC	45		
CP	60		
KLOC	4,683		
Довірчий інтервал	3,7179	5,8975	
Інтервал прогнозування	2,6191	8,3706	
KLOC емпіричне	4,832		
MRE	0,03		

Експортувати результати

Рисунок В.5 – Вкладка «Оцінювання» із MRE розрахунком

ДОДАТОК Г – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ

Об’єкт випробувань: Програма для оцінювання розміру web застосунків, що створюються з використанням фреймворку Vue.

Ціль випробувань: перевірка відповідності застосунку до вимог технічного завдання і повноти реалізації.

Склад програмної документації:

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

Технічні вимоги

Застосунок має експлуатуватись на обладнанні з параметрами не нижчими за наведені: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам’ять 4 Гб, простір на жорсткому диску 150 Мб.

Методи випробувань

Випробування програми виконується представником замовника в присутності представника розробника. Відповідно до опису програми виконується запуск застосунку у браузері. Далі виконується перевірка роботи програми. Виконується перевірка всіх екранних форм застосунку, а також всіх допустимих для виконання операцій всередині екранних форм згідно до посібника користувача. Якщо програма демонструє адекватне функціонування та справність згідно до вимог ТЗ та представленої програмної документації, то робота з розробки програми вважається виконаною, і виконується

впровадження програми до експлуатації. В іншому випадку програма має бути відправлено на доробку розробникові.

ДОДАТОК Д – ТЕКСТ ПРОГРАМИ

```

<template>
  <div class="b-container">
    <div class="form">
      <span>Завантажити файл</span>
      <input
        type="file"
        accept=".xls, .xlsx"
        aria-label="upload image button"
        @change="uploadFile"
      />
      <b-button-group>
        <b-button>Емпіричні дані</b-button>
        <b-button>Нормалізація</b-button>
        <b-button>Перспекія</b-button>
        <b-button>Оцінювання</b-button>
      </b-button-group>
      <b-form-radio-group
        v-model="selectedVariable"
        class="form__variables"
        v-show="false"
      >
        <b-form-radio value="Y">Y</b-form-radio>
        <b-form-radio value="X-first">X<sub>1</sub></b-form-radio>
        <b-form-radio value="X-second">X<sub>2</sub></b-form-radio>
        <b-form-radio value="X-third">X<sub>3</sub></b-form-radio>
      </b-form-radio-group>
      <div class="form__empiric-data" v-show="false">

```

```

<span>
  <p>Вибірк. сер.</p>
  <b-form-input type="text"></b-form-input>
</span>
<span>
  <p>Дисперсія</p>
  <b-form-input type="text"></b-form-input>
</span>
<span>
  <p>Серед. квадр.</p>
  <b-form-input type="text"></b-form-input>
</span>
<span>
  <p>Асиметрія</p>
  <b-form-input type="text"></b-form-input>
</span>
<span>
  <p>Асиметрія<sup>2</sup></p>
  <b-form-input type="text"></b-form-input>
</span>
<span>
  <p>Ексцес</p>
  <b-form-input type="text"></b-form-input>
</span>
  <b-button variant="success">Далі</b-button>
</div>
<div class="form__regression" v-show="false">
  <div class="form__line">
    <p>Лінійна регресія</p>
    <span>
      <p>b<sub>0</sub></p>

```

```

        <b-form-input type="text"></b-form-input>
    </span>
    <span>
        <p>b<sub>1</sub></p>
        <b-form-input type="text"></b-form-input>
    </span>
    <span>
        <p>b<sub>2</sub></p>
        <b-form-input type="text"></b-form-input>
    </span>
    <span>
        <p>b<sub>3</sub></p>
        <b-form-input type="text"></b-form-input>
    </span>
    <span>
        <p>R<sup>2</sup></p>
        <b-form-input type="text"></b-form-input>
    </span>
    <span>
        <p>MMRE</p>
        <b-form-input type="text"></b-form-input>
    </span>
    <span>
        <p>PRED</p>
        <b-form-input type="text"></b-form-input>
    </span>
</div>
<div class="form__non-line">
    <p>Нелінійна регресія</p>
    <b-form-textarea></b-form-textarea>
    <span>

```

```

        <p>R<sup>2</sup></p>
        <b-form-input type="text"></b-form-input>
    </span>
    <span>
        <p>MMRE</p>
        <b-form-input type="text"></b-form-input>
    </span>
    <span>
        <p>PRED</p>
        <b-form-input type="text"></b-form-input>
    </span>
    <b-button variant="success">Далі</b-button>
</div>
</div>
<div class="form__estiamte">
    <span>
        <p>NC</p>
        <b-form-input type="text"></b-form-input>
    </span>
    <span>
        <p>CPL</p>
        <b-form-input type="text"></b-form-input>
    </span>
    <span>
        <p>CP</p>
        <b-form-input type="text"></b-form-input>
    </span>
    <span>
        <p>KLOC</p>
        <b-form-input type="text"></b-form-input>
    </span>

```

```

<span class="spec">
  <p>Довірчий інтервал</p>
  <div>
    <b-form-input type="text"></b-form-input>
    <b-form-input type="text"></b-form-input>
  </div>
</span>
<span class="spec">
  <p>Інтервал прогнозування</p>
  <div>
    <b-form-input type="text"></b-form-input>
    <b-form-input type="text"></b-form-input>
  </div>
</span>
<span>
  <p>KLOC емпіричне</p>
  <b-form-input type="text"></b-form-input>
</span>
<span>
  <p>MRE</p>
  <b-form-input type="text"></b-form-input>
</span>
<b-button variant="outline-primary" class="form__file-uploader"
  >Експортувати результати</b-button
>
</div>
</div>
</div>
</template>

<script lang="ts">

```

```

import Vue from "vue";

export default Vue.extend({
  name: "IndexPage",
  data() {
    return {
      selectedVariable: "X1",
      inputFile: "",
    };
  },
  methods: {
    uploadFile(e) {
      this.inputFile = e.target.files[0];

      const formData = new FormData();

      formData.append("file", this.inputFile);

      this.$axios({
        method: "POST",
        url: `/api/upload-file`,
        type: "multipart/form-data",
        data: formData,
      })
        .then((res) => {
          clearFile();
          empiricNextTab(res.data);
        })
        .catch((err) => this.$toast.error(err.response.data.message));
    },
  },
});

```

```

});
</script>

<?php

use App\Http\Controllers\EmpiricController;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

Route::middleware('jwt.verify')->prefix('/empiric-data')->group(function () {
    Route::post('/upload-file', EmpiricController::class, 'uploadFile');
    Route::post('/calculate', EmpiricController::class, 'calculate');
});

Route::middleware('jwt.verify')->prefix('/normalization')->group(function () {
    Route::post('/calculate', NormalizationController::class, 'calculate');
});

Route::middleware('jwt.verify')->prefix('/regression')->group(function () {
    Route::post('/calculate', RegressionController::class, 'calculate');
});

Route::middleware('jwt.verify')->prefix('/estamation')->group(function () {
    Route::post('/calculate', EstimationController::class, 'calculate');
    Route::post('download-file', EstimationController::class, 'downloadFile');
});

namespace App\Http\Controllers;

use App\Repositories\EmpiricRepository;
use App\Services\EmpiricService;

```

```

use Illuminate\Http\JsonResponse;

use Illuminate\Http\Request;

class EmpiricController extends Controller
{
    protected $empiricRepository;

    protected $empiricService;

    public function __construct(
        EmpiricRepository $empiricRepository,
        EmpiricService $empiricService
    ) {
        $this->empiricRepository = $empiricRepository;

        $this->empiricService = $empiricService;
    }

    public function uploadFile(Request $request)
    {
        $this->validate($request, [
            'uploaded_file' => 'required|file|mimes:xls,xlsx'
        ]);

        $response = $this->empiricService->uploadFile($request->file('uploaded_file'));

        return response()->json([
            'status' => 'success',
            'data' => $response
        ], JsonResponse::HTTP_OK);
    }
}

```

```

        public function calculate(Request $request)
        {
        }
    }

namespace App\Services;

use App\Repositories\EmpiricRepository;
use Illuminate\Support\Facades\DB;
use PhpOffice\PhpSpreadsheet\Spreadsheet;
use PhpOffice\PhpSpreadsheet\Reader\Exception;
use PhpOffice\PhpSpreadsheet\Writer\Xls;
use PhpOffice\PhpSpreadsheet\IOFactory;

class EmpiricService
{
    protected $empiricRepository;

    public function __construct(EmpiricRepository $empiricRepository)
    {
        $this->empiricRepository = $empiricRepository;
    }

    public function uploadFile($file)
    {
        try {
            $spreadsheet = IOFactory::load($file->getRealPath());
            $sheet        = $spreadsheet->getActiveSheet();
            $row_limit    = $sheet->getHighestDataRow();

```

```

$column_limit = $sheet->getHighestDataColumn();
$row_range    = range(2, $row_limit);
$column_range = range('F', $column_limit);
$startcount = 2;
$data = array();

foreach ($row_range as $row) {
    $data[] = [
        'NC' => $sheet->getCell('A' . $row)->getValue(),
        'CPL' => $sheet->getCell('B' . $row)->getValue(),
        'CP' => $sheet->getCell('C' . $row)->getValue(),
        'LKOC' => $sheet->getCell('D' . $row)->getValue(),
    ];
    $startcount++;
}

$this->empiricRepository->insert($data);
$this->calculate($data);

return true;
} catch (Exception $e) {
    $error_code = $e->errorInfo[1];
    return back()->withErrors('There was a problem uploading the data!');
}
}

public function calculate($data)
{
    $values = [];

    foreach ($data as $empiricData) {

```

```

        if (strpos($empiricData, ',')) {
            str_replace(',', '.', $empiricData);

            $value = (10 * $intercept) * ($empiricData[0] * $coefs[0]) * ($empiricData[1] *
            $coefs[1]) * ($empiricData[2] * $coefs[2]);

            if ($value !== null) {
                array_push($values, $value);
            }
        }
    }

    return $values;
}
}

```

```
<?php
```

```
namespace App\Repositories;
```

```
use App\Models\EmpiricModel;
```

```
class EmpiricRepository
```

```
{
```

```
    protected $empiric;
```

```
    public function __construct(EmpiricModel $empiric)
```

```
    {
```

```
        $this->empiric = $empiric;
```

```
    }
```

```
public function insert($data)
{
    return $this->empiric
        ->create([
            'user_id' => Auth::user()->id,
            'components_count' => $data['NC'],
            'plugin_count' => $data['CPL'],
            'page_count' => $data['CP'],
            'code_lines' => $data['KLOC']
        ]);
}
```