

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

«Проектування та розробка web-сервісу для перевірки коду на плагіат»

Студента 2м курсу, 2з групи,
спеціальності 121 «Інженерія
програмного забезпечення»
спеціалізації «Інженерія
програмного забезпечення»

підпис студента

Лісовця
Вадима Івановича

Науковий керівник
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення
та кібербезпеки

підпис керівника

Савченко Тетяна
Віталіївна

Гарант освітньої програми доктор
економічних наук, професор
кафедри інженерії програмного
забезпечення та кібербезпеки

підпис гаранта

Токар Володимир
Володимирович

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

«29» грудня 2020 р.

Завдання

на випускний кваліфікаційний проєкт студентів

Лісовцю Вадиму Івановичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Проектування та розробка
web-сервісу для перевірки коду на плагіат»

Затверджена наказом ректора від «30» листопада 2020 р. № 3224 (стаціонар)

2. Строк здачі студентом закінченого проєкту 25 листопада 2021р.

3. Цільова установка та вихідні дані до проєкту

Мета проєкту проведення глибокого аналізу та порівняння різноманітних
методів та технологій призначених для створення програмного забезпечення,
дослідження як використовуються дані технології

Об'єкт дослідження технологія для перевірки програмного коду на плагіат

Предмет дослідження програмний засіб для перевірки коду на плагіат

4. Консультанти проекту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Аналіз особливостей виявлення програмного плагіату

1.1.1 Варіанти модифікації програмного коду

1.2 Класифікація існуючих підходів для виявлення програмного плагіату

1.3 Існуючі системи пошуку плагіату

РОЗДІЛ 2 ДОСЛІДЖЕННЯ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ ТЕХНОЛОГІЙ ТА МЕТОДІВ ДЛЯ РОЗРОБКИ ВЕБ ДОДАТКУ ДЛЯ ПОШУКУ ПЛАГІАТУ У ПРОГРАМНОМУ КОДІ

2.1. Переваги веб-додатка над іншими типами програмного забезпечення у вирішенні поставленої задачі

2.2 Обґрунтування вибору інструментальних засобів розробки програмного засобу

2.2.1 Інструментальні засоби для Frond-end.

2.2.2 Інструментальні засоби для Back-end

2.3 Вибір СУБД та технологій взаємодії з обраною мовою програмування

РОЗДІЛ 3 РОЗРОБКА ВЕБ ДОДАТКУ ДЛЯ ПЕРЕВІРКИ ПРОГРАМНОГО КОДУ НА ПЛАГІАТ

3.1 Постановка задачі та формулювання вимог до програмного засобу

3.2 Вибір моделі розробки програмного засобу

3.3 Опис проекту

3.4 Проектування бази даних системи

3.5 Програмна реалізація та її особливості

3.6 Тестування та налагодження програмного засобу

3.7 Особливості використання та впровадженню програмного засобу

ВИСНОВКИ ТА ПРПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	21.09.2020	21.09.2020
2.	<i>Розробка та затвердження завдання на проєкт магістра (стац/заоч)</i>	27.02.2021 27.02.2021	22.12.2020
3.	<i>Вступ та перелік літературних джерел</i>	27.02.2021	27.02.2021
4.	<i>Розробка технічного завдання</i>	20.03.2021	20.03.2021
5.	<i>Розділ 1. Аналіз предметної області та існуючих рішень</i>	16.04.2021	16.04.2021
6.	<i>Розділ 2. Дослідження на порівняльний аналіз технологій та методів для розробки веб додатку для пошуку плагіату в програмному коді</i>	24.05.2021	24.05.2021
7.	<i>Розділ 3. Розробка веб додатку для перевірки програмного коду на плагіат</i>	21.06.2021	21.06.2021
8.	<i>Розробка програми та методики тестування</i>	18.10.2021	18.10.2021
9.	<i>Написання наукової статті</i>	22.05.2021	22.05.2021
10.	<i>Керівництво користувача</i>	21.10.2021	21.10.2021
11.	<i>Висновки та пропозиції</i>	01.11.2021	01.11.2021
12.	<i>Здача випускного кваліфікаційного проєкту на кафедрі (перша перевірка)</i>	03.11.2021	03.11.2021
13.	<i>Підготовка автореферату та презентації доповіді</i>	03.11.2021	03.11.2021
14.	<i>Попередній захист випускного кваліфікаційного проєкту</i>	22.11.2021 – 25.11.2021	22.11.2021
15.	<i>Здача зброшурованої випускного кваліфікаційного проєкту</i>	25.11.2021	25.11.2021
16.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>	26.11.2021	26.11.2021
17.	<i>Підготовка до публічного захисту випускного кваліфікаційного проєкту</i>		

7. Дата видачі завдання «29» грудня 2020 р.

8. Науковий керівник випускного кваліфікаційного проєкту
Савченко Т.В.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Токар В.В.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Лісовець В.І.

(прізвище, ініціали, підпис)

11. Відгук керівника випускного кваліфікаційного проєкту

Науковий керівник випускного кваліфікаційного проєкту

_____ (підпис, дата)

Відмітка про попередній захист _____ Савченко Т.В.
(ПІБ, підпис, дата)

12. Висновок про випускний кваліфікаційний проєкт

Випускний кваліфікаційний проєкт студента _____ Лісовець В.І.
(прізвище, ініціали)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми _____ Токар В.В.
(прізвище, ініціали, підпис)

Завідувач кафедри _____ Криворучко О. В.
(підпис, прізвище, ініціали)

« _____ » _____ 20 _____ р.

АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена проведенню глибокого аналізу та порівняння різноманітних методів та технологій призначених для створення програмного забезпечення, дослідження як використовуються дані технології та розробці веб додатку для пошуку плагіату у програмного коді.

В результаті порівняльного аналізу аналогічних рішень визначено, що дане дослідження досить актуальне, так як на ринку не багато якісних програм для пошуку плагіату у програмному коді.

Розробка серверної частини виконана у середовищі PHP.

Готовий програмний продукт було успішно протестовано.

Ключові слова: GPLag, API, PHP, MySql

ABSTRACT

In accordance with the purpose of the study, the work is devoted to in-depth analysis and comparison of various methods and technologies designed to create software, study how these technologies are used and develop a web application to search for plagiarism in software code.

As a result of comparative analysis of similar solutions, it was determined that this study is quite relevant, as there are not many high-quality programs on the market to search for plagiarism in program code.

The development of the server part is performed in the PHP environment.

The finished software package was successfully tested.

Keywords: GPlag, API, PHP, MySql

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ.....	6
1.1 Аналіз особливостей виявлення програмного плагіату	6
1.1.1 Варіанти модифікації програмного коду	7
1.2 Класифікація існуючих підходів для виявлення програмного плагіату... ..	13
1.3 Існуючі системи пошуку плагіату	15
РОЗДІЛ 2 ДОСЛІДЖЕННЯ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ ТЕХНОЛОГІЙ ТА МЕТОДІВ ДЛЯ РОЗРОБКИ ВЕБ ДОДАТКУ ДЛЯ ПОШУКУ ПЛАГІАТУ У ПРОГРАМНОМУ КОДІ.....	18
2.1. Переваги веб-додатка над іншими типами програмного забезпечення у вирішенні поставленої задачі	18
2.2 Обґрунтування вибору інструментальних засобів розробки програмного засобу	21
2.2.1 Інструментальні засоби для Frond-end.....	21
2.2.2 Інструментальні засоби для Back-end	27
2.3 Вибір СУБД та технологій взаємодії з обраною мовою програмування .	28
РОЗДІЛ 3 РОЗРОБКА ВЕБ ДОДАТКУ ДЛЯ ПЕРЕВІРКИ ПРОГРАМНОГО КОДУ НА ПЛАГІАТ	32
3.1 Постановка задачі та формулювання вимог до програмного засобу	32
3.2 Вибір моделі розробки програмного засобу	33
3.3 Опис проекту	36
3.4 Проектування бази даних системи	38
3.5 Програмна реалізація та її особливості.....	39
3.6 Тестування та налагодження програмного засобу	42
3.7 Особливості використання та впровадженню програмного засобу	44
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
ДОДАТКИ	61

					<i>КНТЕУ 121 023-09.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.	Криворучко О.В.			27.02.2021	Проектування та робробка web-сервісу для перевірки коду на плагіат	Стадія	Аркуш	Аркушів
Керівник	Савченко Т.В.			27.02.2021		3	2	56
Гарант	Токар В.В.			27.02.2021	Зміст	Факультет інформаційних технологій 2м курс, 02з група		
Розробив	Лісовець В.І.			27.02.2021				

ВСТУП

В даний час вплив інформаційних технологій на систему освіти дуже великий. Безперечно є позитивні сторони такого впровадження, оскільки відбувається підвищення доступності інформації, а також прискорення та спрощення комунікацій. Студенти мають можливість швидко і практично без витрат користуватися джерелами та необхідною інформацією, для отримання якої раніше потрібно було витратити велику кількість часу та сил. Аналогічне твердження стосується і комунікації: студент може зв'язатися з викладачем з дому, іншого міста і навіть з іншої країни.

Таким чином, основна перевага застосування інформаційних технологій у освіті - це прискорення, спрощення та розширення обміну інформацією. Однак саме це є одночасно і недоліком, мінусом процесів інформатизації, тому що така легкість породжує проблему, яка стає все актуальнішою останнім часом – проблему плагіату в освіті.

Плагіат перетворився на суттєву проблему вищої освіти більшості країн світу. За деякими оцінками частка самостійних робіт учнів, які мають очевидні ознаки плагіату, досягає 70 і більше відсотків. Це пов'язано з поширенням мережі Інтернет, зростаючою навантаженістю викладачів, а також неповагою до авторського права та невисоким рівнем академічної культури [1].

Тому багато освітніх закладів усвідомлюють необхідність проведення політики неприпустимості плагіату і вживають конкретних заходів щодо його виявлення. До більш дієвих з них слід віднести впровадження відповідного програмного забезпечення.

					<i>КНТЕУ 121 02з-09.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		27.02.2021	Проектування та розробка web-сервісу для перевірки коду на плагіат	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Савченко Т.В.		27.02.2021		<i>В</i>	<i>3</i>	<i>56</i>
Гарант		Токар В.В.		27.02.2021		Факультет інформаційних технологій 2м курс, 02з група		
Розробив		Лісовець В.І.						
Вступ								

На жаль, хоча технологія виявлення плагіату програмного забезпечення існує, лише кілька інструментів реалізують її доступною і простою у використанні на загально університетському рівні, або між університетами.

Об'єкт дослідження: технологія для перевірки програмного коду на плагіат.

Предмет дослідження: програмний засіб для перевірки коду на плагіат.

З огляду на це метою роботи є проведення глибокого аналізу та порівняння різноманітних методів та технологій призначених для створення програмного забезпечення, дослідження як використовуються дані технології.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- опрацювати відповідні джерела;
- визначити існуючі підходи, технології і методики розробки програмного засобу;
- проаналізувати сучасні веб-рішення в предметній області;
- обґрунтувати вибір інструментальних засобів розробки програмного засобу;
- розробити програмний засіб для перевірки коду на плагіат;
- показати основні режими функціонування програмного засобу;
- організувати тестування та налагодження програмного засобу.

Використані методи: системно-структурний та порівняльний аналіз – використаний при аналізі проблем програмного забезпечення, призначеного для аналізу програмного коду на плагіат; методи формально-логічного аналізу – при обґрунтуванні вибору методів розробки програмного засобу; економіко-статистичні методи – при дослідження тенденцій програмних засобів; системний підхід – заснований на представленні досліджуваного об'єкта в ясно організованій цілісності.

Практичне значення одержаних результатів полягає у тому, що отримані результати дозволяють ефективніше організувати роботу програмного засобу для аналізу програмного коду на плагіат, також у розробці функціональних можливостей і параметрів які відповідають ефективному програмному засобі. Було проведено дослідження яке полягає в аналізі та узагальненні основних тенденцій розвитку програмних засобів аналізу програмного коду на плагіат.

					<i>КНТЕУ 121 023-09.МР</i>	
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		5

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Аналіз особливостей виявлення програмного плагіату

У сучасному світі будь-який навчальний заклад має справу з проблемою плагіату учнями частин або повних робіт з різних тем, що не лише порушує авторські права, а й негативно позначається на якості здобутої освіти. Дану проблему вирішують різні послуги перевірки текстових робіт на плагіат.

Але самостійна діяльність учнів не обмежується лише текстовими роботами. Так, викладачі навчальних закладів, які ведуть підготовку ІТ-фахівців, проводять навчання та контроль роботи з програмним кодом, перевірка на плагіат якого стає скрутною за допомогою вищевказаних сервісів. Учень може копіювати лише частину вихідного коду, змінити імена змінних та функцій, змінювати розташування блоків, що візуально змінить код роботи, але не позначиться на його виконанні.

Крім того, некоректні запозичення програмного коду є серйозною проблемою і при розробці програмних продуктів.

Таким чином, стає актуальним завдання розробки алгоритму, що забезпечує автоматичне виявлення некоректних запозичень вихідного коду програм різними мовами програмування. Метод повинен дозволяти виявляти різні типи запозичення з достатньою точністю для його використання як під час перевірки навчальних робіт, так і надалі для застосування в галузі аналізу програмного коду програмних продуктів.

					<i>КНТЕУ 121 023-09.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.Б.		16.04.2021	Проектування та розробка web-сервісу для перевірки коду на плагіат	Стадія	Аркуш	Аркушів
Керівник		Савченко Т.В.		16.04.2021		РІ	6	56
Гарант		Токар В.В.		16.04.2021	Аналіз предметної області та існуючих рішень	Факультет інформаційних технологій 2м курс, 02з група		
Розробив		Лісовець В.І.		16.04.2021				

Плагіат у програмуванні - це використання чужого коду з присвоєнням авторства. Поняття плагіату в програмуванні виникло у 1990-х рр., коли було розроблено сервіс MOSS (Measure of Software Similarity, оцінка схожості ПЗ) для перевірки робіт з програмування на плагіат.

1.1.1 Варіанти модифікації програмного коду

Можна виділити чотири типи модифікації програмного коду для приховування факту плагіату:

Перший тип збігів можна представити як повністю скопійовану ділянку коду і, за фактом, є повною копією оригіналу. Проте можуть бути деякі варіації в пробільних символах (пробіли, переноси рядків, символи табуляції тощо), коментарях та/або розмітці. Програми даного типу широко відомі як точні клони [2]. Розглянемо наступний фрагмент коду:

```
if (a >= b) {
    c = d + b; // Comment1
    d = d + 1;}
else
    c = d - a; //Comment2
```

Рис. 1.1. Лістинг оригінального коду

І порівняємо його з точною копією, поданою нижче:

						<i>KHTEY 121 023-09.MP</i>	7
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			

```
if (a>=b) {  
    // Comment1'  
    c=d+b;  
    d=d+1;}  
else // Comment2'  
    c=d-a;
```

Рис. 1.2. Модифікація програмного коду першого типу

Ми бачимо, що ці два фрагменти є текстуально схожими після видалення пробілів і коментарів.

Тим не менш, навіть після видалення коментарів і пробілів, наступний фрагмент коду не схожий на два попередні при порівнянні починаючи з символу "{" і закінчуючи "}". Типові рядкові методи виявлення не зможуть виявити такі ділянки ідентичного коду, але при цьому цей фрагмент також відноситься до першого типу модифікації і є точною копією двох інших.

					КНТЕУ 121 023-09.МР	8
Зм.	Лист	№ докум.	Підпис	Дата		

```

if (a>=b)
{ // Comment1''
    c=d+b;
    d=d+1;
}
else // Comment2''
    c=d-a;

```

Рис. 1.3. Модифікація програмного коду першого типу

Другий тип збігів являє собою фрагмент коду, який збігається з оригінальним за винятком різних імен ідентифікаторів визначених користувачем (імен змінних, констант, класів, методів і т. д.), типів, розмітки, коментарів. Зарезервовані слова і синтаксична структура залишається такою самою, як у оригіналі [3]. Розглянемо наступний фрагмент коду:

```

if (a>= b) {
c = d + b; // Comment1
d = d + 1;} else
    c = d - a; //Comment2

```

Рис. 1.4. Лістинг оригінального коду

Його клоном другого типу може бути фрагмент коду представлений нижче:

```
if (m >= n)
{ // Comment1'
  y = x + n;
  x = x + 5; //Comment3
}
else
  y = x - m; //Comment2'
```

Рис. 1.5. Модифікація програмного коду другого типу

Ми бачимо, що два фрагменти коду змінені за своєю формою, іменам змінних і значенням, що присвоюються. Однак, синтаксична структура, як і раніше, однакова в обох сегментах.

У третьому типі запозичень, скопійований фрагмент додатково модифікований за допомогою зміни, додавання та/або видалення деякої ділянки коду [4]. Розглянемо оригінальний фрагмент:


```
if (a >= b) {  
    c = d + b; // Comment1  
    d = d + 1;}  
else  
    c = d - a; //Comment2
```

Рис. 1.6. Лістинг оригінального коду

Тепер, якщо додати вираз e=1, отримаємо:

```
if (a >= b) {  
    c = d + b; // Comment1  
    e = 1; // Новое выражение  
    d = d + 1; }  
else  
    c = d - a; //Comment2
```

Рис. 1.5. Модифікація програмного коду третього типу

З лістингів видно, що одна ділянка отримана з іншої шляхом повного копіювання та заміни імен деяких ідентифікаторів, а також вставкою додаткового ключового виразу. Тому ці зміни можна віднести до третього типу. Якщо не враховувати додавання додаткового «шуму», то фрагмент, що копіюється, може бути віднесений до другого типу.

									<i>КНТЕУ 121 023-09.МР</i>	
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>						11

Четвертий тип копій є результатом семантичної подібності між двома або більше фрагментами коду. У цьому типі клонів, клонований фрагмент не обов'язково скопійований з оригіналу. Два фрагменти коду можуть бути розроблені двома різними програмістами, реалізовувати однакову логіку та мати аналогічні за своєю функціональністю алгоритми [5].

Розглянемо наступний фрагмент коду, де кінцеве значення змінної 'J' є факторіалом значення змінної VALUE.

```
int i, j=1;
for (i=1; i<=VALUE; i++)
j=j*i;
```

Рис. 1.6. Фрагмент першої частини коду

Тепер розглянемо ще один фрагмент, який є рекурсивною функцією обчислення факторіалу аргументу n.

```
int factorial(int n) {
    if (n == 0) return 1;
else    return n * factorial(n-1);
}
```

Рис. 1.7. Фрагмент другої частини коду

З семантичної точки зору це різні ділянки коду, але вони однакові за своєю функціональністю і відносяться до четвертого типу.

Четвертий вид запозичення можна не враховувати, в даному випадку складно говорити про плагіат у явному вигляді так як людина запозичує код, а копіює лише «ідею» (логіку, функціонал), але при цьому створює власну програму.

1.2 Класифікація існуючих підходів для виявлення програмного плагіату

Існує велика кількість способів представлення подальшого аналізу вихідних кодів програм. В основному всі вони складаються з фази перетворення і фази порівняння. На першому етапі вихідний текст перетворюється на внутрішній формат, який дозволяє використовувати ефективніший алгоритм порівняння. У наступній фазі відбувається безпосередній пошук збігів і виведення деякої кількісної міри, що характеризує їх кількість.

Найбільш простим методом виявлення плагіату є просте посимвольне порівняння програмних кодів. Даний метод не залежить від мови, якою реалізована програма, але досить легко обходиться елементарним перейменуванням змінних, функцій, додаванням та видаленням коментарів. Підходить для виявлення запозичень лише першого типу. Існують різні методи виявлення плагіату на основі токенизації вихідного коду, наприклад алгоритм Хескела, вирівнювання рядків (Greedy String Tiling). Процедура токенизації відбувається так [6]:

- визначаються класи операторів та їм призначаються коди. Всім операторам мови (що не є операндами) надаються відповідні коди класів
- будується рядок отриманих кодів (токенів), при цьому послідовність кодів відповідає порядку появи операторів у коді.

Недоліком методів, в основі яких лежить токенизація, є залежність від конкретної мови програмування, проте цю проблему можна вирішити шляхом створення різних наборів, кодів, що не перетинаються, для різних класів мов, що ускладнює процес автоматизованої перевірки, так як додається необхідність визначення мови, при тому, що існують оператори, які притаманні різним мовам. Ще одним видом методів виявлення плагіату є методи з використанням метрик. Вони використовують сукупність кількості циклів, змінних, умов чи їх кількості окремо. У цих методах немає прив'язки до мови програмування. Однак недоліком є високий рівень помилкових спрацьовувань під час перевірки невеликих програм.

Тому доцільно класифікувати методи виявлення відповідно до їх внутрішнього формату подання тексту. Поширеними методами є такі [7]:

String-based - є поширеним підходом, що використовується в інформатиці. Стосовно проблеми виявлення плагіату, документи порівнюються дослівно з можливим перекриттям тексту. Численні методи були запропоновані для вирішення цього завдання, деякі з яких були адаптовані саме для виявлення плагіату.

Token-based - схожий на String-based підхід, але позбавлений багатьох його недоліків. Код програми насамперед перетворюється на набір токенів з використанням «лексера». А потім застосовується один з алгоритмів рядкового порівняння для пошуку збігу фрагментів коду.

Tree-based - метод, в якому фаза перетворення складається з побудови абстрактного синтаксичного дерева, яке являє собою внутрішню структуру програми у відповідності з деякою контекстно-вільною граматикою.

PDG-based (PDG – ProgramDependencyGraph) – програма представляється як орієнтований граф, вершинами якого будуть: точка входу, безліч точок виходу, команди передачі управління. З кожною вершиною

асоціюється структура, що зберігає адресу цього вузла, і адресу переходу з цього вузла (якщо існує). Дугам цього графа відповідають ділянки програми, які є командами обробки даних. Потім отримані графи кожного файлу з вихідним кодом порівнюються між собою [8].

Metrics-based - цей метод заснований на оцінці різних метрик програми. В якості таких можуть використовуватися кількість викликів функцій, кількість використовуваних змінних, кількість циклів. Далі дві програми порівнюються за відповідними метриками [9].

Hybrid-based - гібридні підходи засновані на поєднання кількох з вище перерахованих підходів і беруть сильні сторони кожного з них для досягнення максимального результату.

1.3 Існуючі системи пошуку плагіату

З погляду підтримуваних мов програмування системи виявлення плагіату у вихідних кодах програм можна розділити на такі групи:

- універсальні, тобто відповідні для будь-якої мови програмування;
- які підтримують кілька мов програмування.
- спеціалізовані системи для певної мови програмування.

Codequiry – це платний веб-додаток, який позиціонує себе як сучасне рішення для виявлення програмного плагіату. В основі системи лежить алгоритм відбитків документа, що використовується системою Measure Of Software Similarity (див. пункт 2.5.3), але з додатковими покращеннями. Codequiry дозволяє швидко приступити до роботи, без попередньої підготовки. Користувач має створити директорії, які дають можливість 17 виконати перевірку плагіату окремо один від одного. Далі необхідно завантажити програмні проєкти в директорію та виконати перевірку.

Веб-додаток пропонує два способи перевірки вихідного коду на:

1. Peer Check – найбільш повний тест для перевірки плагіату серед завантажених в систему проєктів. Порівнює унікальний відбиток однієї кодової бази з іншою, виявляючи логічну схожість і випадки подібності коду серед множини проєктів.

2. Web Check – тест на плагіат, який порівнює код з більш ніж 100 мільйонами джерел з репозиторіїв веб-сервісу GitHub, а також з більш ніж 2 мільярдами примірників вихідного коду в мережі. Основною перевагою Codequiry над конкурентами є спосіб представлення результатів виявлення плагіату, щоб користувач остаточно визначився чи є програмний проєкт плагіатом [10].

MOSS (Measure Of Software Similarity) – веб-сервіс для виявлення плагіату в вихідних кодів програм, розроблений Алексом Айкеном у 1994 році. Це безкоштовний сервіс, але для його використання потрібна реєстрація. MOSS підтримує багато мов програмування, у тому числі Haskell. MOSS використовує алгоритм відбитків з методом просіювання [11].

MOSS не позиціонується як повністю автоматична система виявлення плагіату. Виявлені випадки потребують ручної перевірки.

З недоліків цієї системи можна назвати:

- ця система доступна лише як веб-сервіс;
- не гарантується, що і в майбутньому ця система працюватиме;

JPlag – це додаток, який знаходить подібність між наборами файлів вихідного коду, що дає можливість виявляти програмний плагіат. JPlag не просто порівнює байти тексту, але знає синтаксис мови програмування та структуру програми, тому є стійким до багатьох технік приховування факту плагіату.

Також JPlag дозволяє попередньо налаштовувати процес виявлення плагіату. Користувач може самостійно вказати перелік розширень файлів, що будуть порівнюватися; назви файлів, що будуть виключені з перевірки; кількість лексем для аналізу мінімального збігу; мінімальний відсоток збігу між проектами для генерації звіту; вказати шаблонний код, який не буде рахуватися системою як плагіат [12].

Роблячи висновки щодо розглянутої системи можна сказати, що JPlag добре справляється з задачею виявлення плагіату в навчальних програмних проектах. Результати аналізу дозволяють викладачу переглянути вихідний код, підозрюваний на плагіат. А можливість гнучкого налаштування системи – отримати більш точний результат в кожному окремому випадку.

					<i>КНТЕУ 121 023-09.MP</i>	17
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 2

ДОСЛІДЖЕННЯ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ ТЕХНОЛОГІЙ ТА МЕТОДІВ ДЛЯ РОЗРОБКИ ВЕБ ДОДАТКУ ДЛЯ ПОШУКУ ПЛАГІАТУ У ПРОГРАМНОМУ КОДІ

2.1. Переваги веб-додатка над іншими типами програмного забезпечення у вирішенні поставленої задачі

Сьогодні програмне забезпечення розробляється для широкого діапазону електронних пристроїв, які можуть суттєво відрізнятися продуктивністю, габаритами, функціональним призначенням та іншими характеристиками. Кожен із пристроїв потребує індивідуального підходу до розроблюваного програмного забезпечення. Найпопулярнішими видами додатків є десктопні (мобільні), дані додатки будемо розглядати у групі, так як у них дуже мало відмінностей, та веб програми.

Отже, веб-застосунок працює через браузер, використовуючи його як середовище виконання, десктопний додаток - встановлюється, запускається і працює локально. Порівнюємо їх за основними характеристиками.

Встановлення, оновлення:

Веб-додаток не потребує встановлення, всі оновлення відбуваються на сервері, доставляються користувачам відразу - досить просто перезавантажити сторінку або вийти, а потім знову зайти в обліковий запис. Але іноді для його роботи потрібно встановити додаткові бібліотеки або використовувати захищені протоколи мережі.

					<i>КНТЕУ 121 023-09.МР</i>			
<i>Зм.</i>	<i>Аркви</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.	Криворучко О.В.			24.05.2021	Проектування та розробка web-сервісу для перевірки коду на плагіат	<i>Стадія</i>	<i>Аркви</i>	<i>Аркушів</i>
Керівник	Савченко Т.В.			24.05.2021		<i>Р2</i>	<i>18</i>	<i>56</i>
Гарант	Токар В.В.			24.05.2021	Дослідження та порівняльний аналіз технологій та методів для розробки веб додатку для пошуку плагіату у програмному коді	Факультет інформаційних технологій 2м курс, 02з група		
Розробив	Лісовець В.І.			24.05.2021				

Десктопний додаток потрібно встановлювати на комп'ютері або мобільному пристрої, оновлювати щоразу, коли виходить нова версія. Незважаючи на те, що найчастіше процес автоматизований — це все одно займає час користувачів і ресурси пристроїв. Додатково доведеться відстежувати версії на кожному комп'ютері, смартфоні та планшеті.

Публікація / розгортання:

Веб-програма публікується на локальному або хмарному сервері, там же відбувається процес оновлення. У цьому сервер потрібен у разі, навіть якщо рішення дуже просте. Адже крім фронтенду, з яким користувачі працюватимуть через браузер, потрібно десь розміщувати бекенд.

Десктопний додаток доведеться встановлювати вручну на кожному пристрої. У компанії, де багато робочих місць, це може зайняти чимало часу. Плюс у тому, що не обов'язково вибирати сервер або шукати ресурси для публікації, якщо не йдеться про клієнт-серверне рішення.

Надійність:

Робота веб-додатку залежить не тільки від того, наскільки якісно він розроблений і характеристик пристрою, але також від швидкості інтернет-з'єднання, працездатності віддаленого сервера.

Десктопний додаток працює автономно, тому головне - якість коду та стабільність обладнання, на якому цей код виконується. Але якщо зв'язок із сервером необхідний — то виникають ті ж проблеми, що й у веб-додатка.

Доступність:

Веб-додаток доступний з будь-якої точки світу, з будь-якого пристрою, а файли користувача завжди будуть під рукою. Але тільки якщо є інтернет-з'єднання або реалізована можливість роботи офлайн та завантаження даних.

Десктопний додаток доступне завжди, але тільки з пристрою, на якому він встановлений. Щоб працювати з різних пристроїв, його доведеться

					<i>КНТЕУ 121 023-09.МР</i>	
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		19

встановити на кожному, а також вигадати, де зберігати файли, щоб завжди мати до них доступ.

Кросплатформеність:

Веб-додаток однаково добре буде працювати на будь-якому пристрої, чи то стаціонарний комп'ютер, ноутбук, планшет чи смартфон – адже він практично не залежить від «заліза» чи операційної системи. Головне - відповідний браузер.

Десктопний додаток залежить від операційної системи, процесора, відеокарти, інших параметрів. Доводиться враховувати нюанси кожного середовища (у тому числі при відлові помилок), писати код з урахуванням можливих варіантів, наймати окремих розробників або навіть цілі команди для версій під різні ОС.

Функціональність, швидкодія:

Веб-додаток повністю залежить від браузера та технологій його роботи. Тому є низка обмежень, наприклад, у доступі до апаратного забезпечення вашого пристрою.

Десктопний додаток дозволяє реалізувати буквально будь-які функції - у цьому він однозначно перевершує веб-додатки. Також таке програмне забезпечення частіше використовується для специфічних проєктів, з нестандартними інтерфейсами або функціями. Тому web розробка поки не становить небезпеки для desktop програмістів - ці технології розвиватимуться паралельно, просто під різні завдання.

Щодо швидкості роботи все не так однозначно, як може здатися. Незважаючи на те, що веб-додаток постійно обмінюється даними з сервером, швидкодія багато в чому залежатиме від того, наскільки грамотно він спроектований, «чистоти» коду, можливостей обладнання, стабільності

					<i>КНТЕУ 121 023-09.МР</i>	
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		20

каналу зв'язку. Різниця в швидкодії, яка є очевидною при тестуванні, часто непомітна для користувачів.

Безпека:

Веб-додаток, розроблений з використанням сучасних протоколів та засобів захисту, здатний повноцінно забезпечувати збереження даних. Однак на деякі моменти розробники не можуть вплинути: браузер, хмарний сервер, канал зв'язку можуть підвищити рівень безпеки за рахунок додаткових засобів перевірки, але також знизити його за рахунок своїх вразливостей. Безперечний плюс для користувачів: таке програмне забезпечення простіше контролювати. Обмеження середовища знижують ймовірність, що воно приховано отримає доступ до файлів або запустить будь-який процес.

Десктопний додаток налаштовується гнучкіше, а значить — теоретично при його розробці можна передбачити всі потенційні вразливості. Втім, зробити його повністю безпечним все ж таки можна. Але тільки якщо пристрій, на якому він встановлений, не підключатиметься нікуди, навіть до захищеної локальної мережі. В іншому випадку ризик все одно буде.

Але очевидно, що довіра до десктопного програмного забезпечення вище. Деякі організації принципово не погоджуються працювати в браузерах, багато користувачів все ще ставляться до них насторожено. Проте ситуація змінюється — із розвитком технологій зростає лояльність людей до них.

2.2 Обґрунтування вибору інструментальних засобів розробки програмного засобу

2.2.1 Інструментальні засоби для Front-end.

Як ми вже пояснювали, нова специфікація HTML охоплює не тільки теги, вона виходить за межі звичайного коду HTML. Мережа пред'являє високі вимоги не тільки до структурної організації та визначення розділів, але і до

					<i>КНТЕУ 121 023-09.МР</i>	
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		21

дизайну і функціональності. У цій новій парадигмі HTML об'єднується з CSS і JavaScript в один інтегрований інструмент.

Мова HTML5 робить акцент на спрощення розмітки, необхідної для створення відповідних W3C-стандартам сторінок і об'єднання всього необхідного CSS і JavaScript-коду, а також файлів зображень. Якщо вести мову про користувачів, які, ймовірно, переглядають наші сторінки, підключаючись до Інтернету по каналу з невисокою швидкістю передачі інформації, а також про основну мету наших адаптивних веб-дизайнів, то нам необхідно, щоб сайт не просто адаптувався до їх більш обмежених областей перегляду, але і завантажувалася настільки швидко, наскільки це можливо. Незважаючи на те що при видаленні зайвих елементів розмітки обсяг переданих даних зменшується зовсім трохи, навіть мала економія буде корисна.

Офіційно технологія CSS ніяк не пов'язана з HTML5. Вона не є і ніколи не була частиною специфікації HTML5. Насправді це допоміжна технологія, яка розроблялася з метою подолання обмежень і зменшення складності HTML. Спочатку якісь базові стилі зв'язувалися з кожним елементом за допомогою атрибутів в тегах HTML, проте у міру того, як мова розвивалася, код ставало все складніше розробляти і підтримувати, і незабаром виявилось, що одного HTML недостатньо для задоволення всіх вимог веб-дизайнерів. В результаті на озброєння була взята технологія CSS, що дозволяє відокремити структуру від уявлення [13]. З тих пір CSS успішно розвивалася, проте розробка даної технології йшла паралельно HTML і фокусувалася на потреби дизайнерів, а не на необхідності підтримувати еволюцію HTML.

Третя версія CSS слід аналогічним шляхом, проте її розробники беруть набагато більше компромісних рішень. Специфікація HTML5 має на увазі, що за дизайн тепер відповідає CSS, і через це інтеграція між HTML і CSS3 стала вирішальним фактором веб-розробки. Ось чому завжди, коли ми говоримо про

HTML5, ми також згадуємо CSS3. Це природно, не дивлячись на те що офіційно дані технології незалежні [14].

В даний час можливості CSS3 впроваджуються і реалізуються в браузерях, сумісних з HTML5, поряд з іншою функціональністю, описаною в специфікації.

CSS - це мова, що працює спільно з HTML. Вона пов'язує з елементами документа різноманітні візуальні стилі, що визначають їх розмір, колір, фон, рамки і т. д.

Зараз можливості CSS3 вже вбудовані в останні версії більшості популярних браузерів, проте деякі з них все ще знаходяться на стадії розробки. З цієї причини для забезпечення їх ефективної роботи в назвах нових стилів необхідно використовувати браузерні префікси, такі як moz або webkit (в залежності від механізму, який використовують браузер) [15].

Переваги використання CSS:

- код розмітки розділений на вміст і оформлення. Отримуємо більш легкий і зрозумілий код;
- файл з оформленням завантажується тільки один раз і кешується – зменшується трафік та час завантаження сторінок і навантаження на сервер;
- простота самої мови CSS + принцип відділення оформлення від змісту – скорочується час на розробку і підтримку сайту;
- один файл CSS управляє дизайном великої кількості HTML-сторінок [16].

Зміни дизайну робляться швидше;

- CSS надає додаткові можливості форматування, які в атрибутах HTML відсутні;
- реалізований механізм підключення різних CSS для різних типів пристроїв;

– підвищення сумісності з різними платформами за рахунок використання веб-стандартів [17].

Отже, CSS – це загальновизнаний міжнародний стандарт. CSS - це найпотужніший інструмент, один з найосновніших складових практично будь-якої веб-сторінки, без якої неможливо уявити сучасну веб-розробку.

JavaScript – мова програмування, що дозволяє реалізувати ряд складних рішень в web-документах. Вона допомагає зробити сторінки сайту більш інтерактивними, обробляє дії користувачів сайту. Це об'єктно-орієнтована клієнтська мова, яка підтримується додатками, що працюють з дизайном сайту. JavaScript став ще більш популярним в середовищі девелоперів, коли з'явилася AJAX-технологія, що призвело до нового етапу в розробці сайтів.

Ось лише кілька окремих прикладів застосування технології:

- відображення контенту, який періодично оновлюється (інтерактивні карти);
- створення якісної анімації і графічних об'єктів у форматі 2D/3D;
- опція прокрутки відеозапису в медіа програвачі.

Поряд з HTML і CSS, JavaScript – третій важливий блок, на основі якого будується більшість стандартних веб-інтерфейсів.

Щоб самостійно розібратися з роботою мови, користувачеві знадобиться знання основ HTML і CSS.

Ядро JavaScript включає цілий ряд функцій, які дають наступні можливості:

- зберігати дані в змінних;
- активувати частину коду у відповідності з певними сценаріями, які здійснюються на сторінці сайту;
- створювати контент, який оновлюється автоматично;

– управляти мультимедійними можливостями (працювати з відео, анімувати зображення) [18].

Усього кілька рядків коду скрипта роблять дивовижні речі. Саме тому мова настільки популярна серед розробників.

Але ще більше можливостей дає функціонал, який доступний як надбудова щодо основних складових JavaScript. Мова йде про інтерфейси прикладного програмування (API), які істотно розширюють набір інструментів розробника.

API – це готові модулі коду, які допомагають програмісту в реалізації деяких складних завдань. Зазвичай такі «заготовки» діляться на браузерні і API третіх розробників.

До браузерних API-інтерфейсів відносяться:

- API-інтерфейс DOM (Document Object Model);
- модулі геолокації;
- API Canvas і WebGL;
- аудіо та відео API.

А до сторонніх інтерфейсів належать, наприклад, API соціальних мереж Twitter і Facebook.

Коли ми завантажуюмо сторінку в браузері, то спочатку обробляються HTML і CSS, після чого уже JavaScript.

Разом з тим, мова не містить деяких корисних інструментів. Тут немає:

- стандартної бібліотеки;
- стандартних інтерфейсів для роботи з серверами і базами даних;
- системи для керування даними.

JavaScript обробляється в веб-додатках на стороні клієнта, тобто у браузері. Завдяки цьому він може виконуватися на будь-якій операційній системі, а веб-інтерфейси, які працюють на його основі, є кросплатформними.

Тому JavaScript має широке застосування в таких областях:

- AJAX скрипти використовуються в цьому відомому підході, який передбачає асинхронний обмін даними між браузером користувача та ресурсами сервера.

- Comet – спосіб роботи web-додатків, коли під час HTTP-з'єднання сервер відправляє дані браузеру без додаткових запитів.

- Браузерні ОС – код деяких браузерних операційних систем, який переважно (іноді на більш ніж 75%) складається зі скриптів.

- Закладки – JavaScript має широке застосування в роботі програми, що розміщуються в закладках браузера.

- Браузерні скрипти – програмні модулі, які пишуться на цій мові і дають дуже багато можливостей (автозаповнення форм, зміна формату сторінок, приховування небажаного змісту, додавання інтерактивних елементів на сторінках) [19].

- Серверні додатки – фрагменти коду, які виконуються на стороні сервера, де використовується Java.

- Мобільні додатки – JavaScript може бути корисним і в цьому популярному напрямку.

- Віджети – мовою пишуться різні міні-програми, які використовуються в робочому просторі і є дуже зручними.

- Прикладне ПО – об'єктно-орієнтована мова JavaScript використовується для створення окремих програм, в тому числі нескладних ігор.

Переваги Javascript:

					<i>КНТЕУ 121 023-09.МР</i>	26
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

- підтримують усі браузери;
- підтримують такі програми: Adobe Photoshop, Adobe Illustrator, Adobe InDesign та Adobe Dreamweaver;
- перевірка реєстраційних форм на помилки ще до відправлення на сервер;
- створює яскраві та інтерактивні сторінки сайту;
- можна здійснювати різного типу обчислення.

Сьогодні JavaScript є однією з найбільш популярних клієнтських мов.

2.2.2 Інструментальні засоби для Back-end

Головним чинником мови PHP є практичність. PHP повинен надати програмісту засоби для швидкого та ефективного вирішення поставлених завдань. Практичний характер PHP обумовлений п'ятьма важливими характеристиками:

- традиційністю;
- простотою;
- ефективністю;
- безпекою;
- гнучкістю.

Існує ще одна «характеристика», яка робить PHP особливо привабливим: він розповсюджується безкоштовно! Причому, з відкритими вихідними кодами (Open Source).

PHP - мова, яка може бути вбудований безпосередньо в html - код сторінок, які, в свою чергу будуть коректно оброблятися PHP-інтерпретатором. Ми можемо використовувати PHP для написання CGI-сценаріїв і позбутися від безлічі незручних операторів виведення тексту. Ми

можемо залучати PHP для формування HTML-документів, позбувшись від безлічі викликів зовнішніх сценаріїв [20].

За деякими оцінками, більшість PHP - сценаріїв (особливо не дуже великих розмірів) обробляються швидше за аналогічні їм програм, написаних на Perl. Однак, щоб не робили розробники PHP, відкомпілювалися виконувані файли будуть працювати значно швидше - в десятки, а іноді і в сотні разів. Але продуктивність PHP цілком достатня для створення цілком серйозних web-додатків [21].

2.3 Вибір СУБД та технологій взаємодії з обраною мовою програмування

Вибір системи управління базами даних є одним з важливих етапів при розробці автоматизованої системи. Обраний програмний продукт повинен задовольняти як поточним, так і майбутнім потребам навчального закладу, при цьому слід враховувати витрати розробку і настройку необхідного програмного забезпечення, а також навчання персоналу.

Найбільш правильний підхід при виборі СУБД заснований на оцінці того, якою мірою існуючі системи задовольняють основним вимогам створюваного проекту інформаційної системи.

Існує кілька критеріїв вибору системи управління базами даних:

- моделювання даних;
- особливості архітектури і функціональні можливості;
- контроль роботи системи;
- особливості розробки додатків;
- продуктивність;

- надійність;
- вимоги до робочого середовища;
- змішані критерії.

Багато виробників СУБД випускають також засоби розробки додатків для своїх систем. Як правило, ці кошти дозволяють найкращим чином реалізувати всі можливості сервера, тому при аналізі СУБД варто розглянути також і можливості засобів розробки додатків. Деякі системи мають засоби автоматичного проектування, як баз даних, так і прикладних програм.

Підтримка великої кількості національних мов і можливості розробки web-інтерфейсу розширюють область застосування системи і додатків, побудованих на її основі, а наявність широкого спектра використовуваних мов програмування підвищує доступність системи для розробників і суттєво впливає на швидкодію і функціональність створюваних додатків [22].

Критерій надійності системи включає можливість:

- відновлення системи після збоїв;
- резервного копіювання;
- наявність багаторівневої системи захисту.

Важливо визначити і вимоги до робочого середовища такі як:

- підтримувані апаратні платформи;
- мінімальні вимоги до обладнання;
- максимальний розмір пам'яті;
- операційні системи, під керуванням яких здатна працювати СУБД.

Чіткий і глибокий порівняльний аналіз на підставі перерахованих вище критеріїв в будь-якому випадку допоможе раціонально вибрати відповідну систему для конкретного проекту, і витрачені зусилля не будуть марними. Перелік критеріїв допоможе усвідомити масштабність завдання і виконати її адекватну постановку.

В якості системи управління базами даних в автоматизованій системі використовується найбільш популярна СУБД MySQL.

MySQL - вільна система управління базами даних. Розробка та підтримка сайту MySQL здійснює корпорація Oracle, що має на даний момент права на торговельну марку. Продукт поширюється як під GNU General Public License, так і під власною комерційною ліцензією. Крім цього розробники створюють функціональність за замовленням ліцензійних користувачів, саме завдяки такому замовленню майже в найраніших версіях з'явився механізм реплікації.

Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Більш того, СУБД MySQL поставляється із спеціальним типом таблиць EXAMPLE, що демонструє принципи створення нових типів таблиць. Завдяки відкритій архітектурі і GPL-ліцензуванню, в СУБД MySQL постійно з'являються нові типи таблиць.

MySQL є найбільш пристосованою для застосування в середовищі веб системою управління базами даних. При цьому вона стала непорушним стандартом в області СУБД для веб, в якій розвиваються можливості для використання її в будь-яких критичних бізнес-додатках, що створює конкуренцію на рівних з СУБД таких виробників, як Oracle, IBM, Microsoft і Sybase.

Основні переваги MySQL:

- підтримка декількох одночасних запитів;
- оптимізація зв'язків з приєднанням багатьох даних за один прохід;
- записи фіксованої і змінної довжини;
- ODBC драйвер;

- гнучка система привілеїв і паролів;
- гнучка підтримка форматів чисел, рядків змінної довжини і міток часу;
- інтерфейс з мовами C і Perl, PHP;
- швидка робота, масштабованість;
- сумісність з ANSI SQL;
- безкоштовна в більшості випадків;
- хороша підтримка з боку провайдерів послуг хостингу;
- швидка підтримка транзакцій через механізм InnoDB.

РОЗДІЛ 3 РОЗРОБКА ВЕБ ДОДАТКУ ДЛЯ ПЕРЕВІРКИ ПРОГРАМНОГО КОДУ НА ПЛАГІАТ

3.1 Постановка задачі та формулювання вимог до програмного засобу

Веб-додаток для пошуку плагіату у програмному коді повинна виконувати ряд функцій, для того щоб вона вважалась придатною до застосування користувачем, в не залежності від виду користувача (будь то учень, чи викладач, чи сторонній користувач). Це такі функції як:

- 1) додавання існуючих чи нових документів, які пройшли перевірку на плагіат до локальної БД програми, для забезпечення більш точної перевірки на плагіат нових документів;
- 2) введення нових, неперевіраних на плагіат документів в чергу для перевірки їх на плагіат;
- 3) власне сама перевірка на плагіат;
- 4) видача результатів перевірки;
- 5) можливість зберегти отримані результати;
- 6) очистити результат перевірки робіт.

Перед тим, як перейти до опису структури бази даних слід визначити, які дані потрібно зберігати в ній, а також зв'язки між колекціями (аналог таблиць в реляційних базах даних). В базі даних потрібно зберігати інформацію про документи, файли цих документів, логи (англ. Logs) програми та IDF-вектор, необхідний для перевірки на плагіат. Інформація про документи

<i>КНТЕУ 121 023-09.МР</i>								
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.Е		21.06.2021	Проектування та розробка web-сервісу для перевірки коду на плагіат	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Савченко Т.В.		21.06.2021		<i>РЗ</i>	<i>32</i>	<i>56</i>
Гарант		Токар В.В.		21.06.2021	Розробка веб додатку для перевірки програмного коду на плагіат	Факультет інформаційних технологій 2м курс, 02з група		
Розробив		Лісовець В.І.						

буде тісно пов'язана з інформацією про файли, оскільки документи будуть (були) отримані з файлу.

Сервіс швидко працює і швидко реагує на будь-які дії користувача. Це важливо для ефективної праці користувача.

Сервіс має достатню надійність, зручність у використанні та передбачає можливість внесення змін для розширення його функціоналу.

Веб-ресурс повинен мати простий, ненав'язливий, але оригінальний дизайн – для проекту це одна з найголовніших критерій.

Сайт розроблено таким чином, що він сумісний з усіма браузерами.

Було проведено оптимізацію зображень.

3.2 Вибір моделі розробки програмного засобу

Для розробки програмного забезпечення ми вибрали каскадну модель – це модель процесу розробки програмного забезпечення, в якій процес розробки виглядає як потік, що послідовно проходить фази аналізу вимог, проектування, реалізації, тестування, інтеграції та підтримки.

Ця модель має на увазі строго послідовне і одноразове виконання кожної фази проекту. Перехід від однієї фази до іншої можливий тільки після успішного завершення попереднього етапу. Кожен етап має на увазі детальне планування і повну коректність результату етапу.

Дотримуючись каскадної моделі, ми переходили від однієї стадії до іншої строго послідовно. Спочатку повністю завершили етап «визначення вимог», в результаті чого вийшов список вимог до програмного забезпечення підтримки курсу «Я досліджую світ». Після того як вимоги повністю були визначені, перейшли до етапу проектування, в ході якого були створені документи, що детально описали спосіб і план реалізації зазначених вимог.

Після того як проектування повністю було виконане, нами було виконано реалізацію проекту. Після того як реалізацію завершили, ми провели тестування і налагодження продукту; на цій стадії були усунені всі недоліки, що з'явилися на попередніх стадіях розробки. Після цього програмний продукт був впроваджений і забезпечується його підтримка – внесення нової функціональності і усунення помилок.

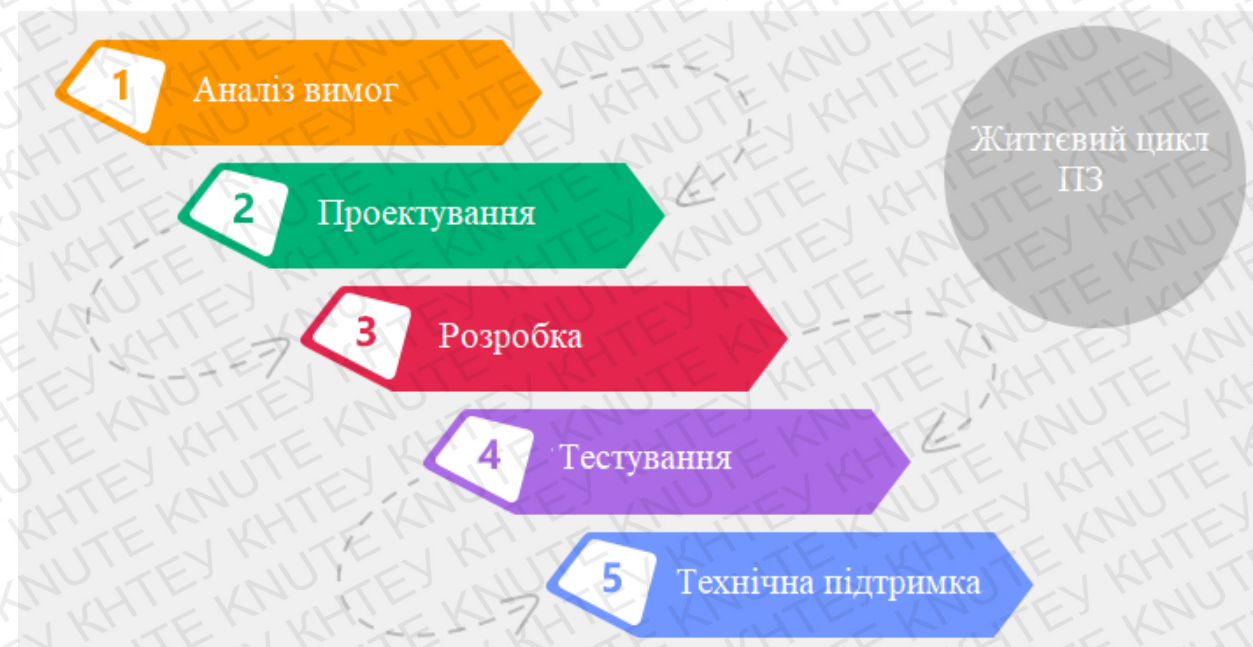


Рис. 3.1 – Візуальний вигляд каскадної моделі

Наведемо переваги каскадної моделі:

Стойка до зміни кадрового складу. Розробники можуть приходити і йти на протязі всього життєвого циклу проекту, але завдяки докладному документуванню це практично не впливає на терміни виконання проекту.

Дисципліна. Модель водоспаду змушує розробників, залучених до проекту бути дисциплінованими, залишатися в рамках наміченого плану. При необхідності в загальній моделі додається орган управління, відповідальний за прийняття рішень, виконавці ж зобов'язані працювати в рамках системи.

Гнучкість на ранніх етапах. Зміни в перших трьох фазах можуть бути зроблені негайно і з мінімальними зусиллями, оскільки вони не підкріплені кодом. Таким чином, замовник і виконавець мають значний часовий запас для кардинальної зміни концепції роботи ПЗ.

Орієнтація на терміни та фінанси. Завдяки тому, що кожен етап повністю окреслює контур майбутнього ПО, всі розробники розуміють свою роль, межі роботи і терміни виконання. Це дозволяє оперувати реальними цифрами перед замовником, що робить модель проекту привабливою.

Недоліки каскадної моделі:

Неадаптивна структура ПЗ. На перших етапах модель водоспаду може бути гнучкою, але якщо на фазі тестування виявляються проблеми в загальній структурі – це тягне за собою плачевні наслідки у вигляді зірваних термінів і навіть відмов замовника. Таким чином, зростає роль керівників і відповідальних розробників, з рівнем компетентності яких в будь-якій компанії часто бувають проблеми.

Пізніше тестування. З опису вище легко виявити самий проблемний етап методології – тестування. Саме тут найчастіше виявляються помилки, допущені на кожному з етапів. Більш гнучкі методології використовують тестування в якості фундаментальної операції, яке відбувається безперервно. «Водоспад» ж допускає низьку кваліфікацію співробітників на кожному етапі і погану якість виконання, адже при запізнілому тестуванні проблеми неможливо вирішити фундаментально, тільки за допомогою «латок» [23].

3.3 Опис проекту

Обираючи майбутню архітектуру системи слід зважати на те, що її ключовими складовими будуть сервер, клієнт та база даних. Оскільки система не містить складної логічної структури буде оптимально використати просту та зрозумілу архітектуру.

Варіант використання описує, з точки зору чинної особи, групу дій в системі, які призводять до конкретного результату. Варіанти використання є описами типових взаємодій між користувачами системи і самою системою. Вони відображають зовнішній інтерфейс системи і вказують форму того, що система повинна зробити (саме що, а не як).

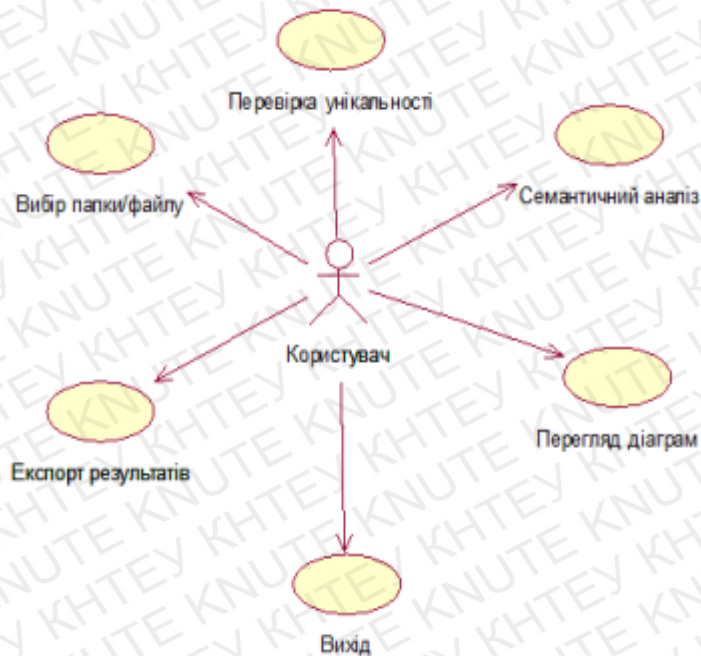


Рис. 3.2. Діаграма варіантів використання

При роботі з варіантами використання важливо пам'ятати декілька простих правил:

–кожен варіант використання відноситься як мінімум до одної дійової особи;

–кожен варіант використання має ініціатора;

–кожен варіант використання призводить до відповідного результату.

Варіанти використання також можуть взаємодіяти з іншими варіантами використання. Єдиний актор проєктованої системи, буде взаємодіяти із всіма варіантами використання. У модель включено асоціації, що відповідають за напрямки передачі інформації між актором і варіантами використання.

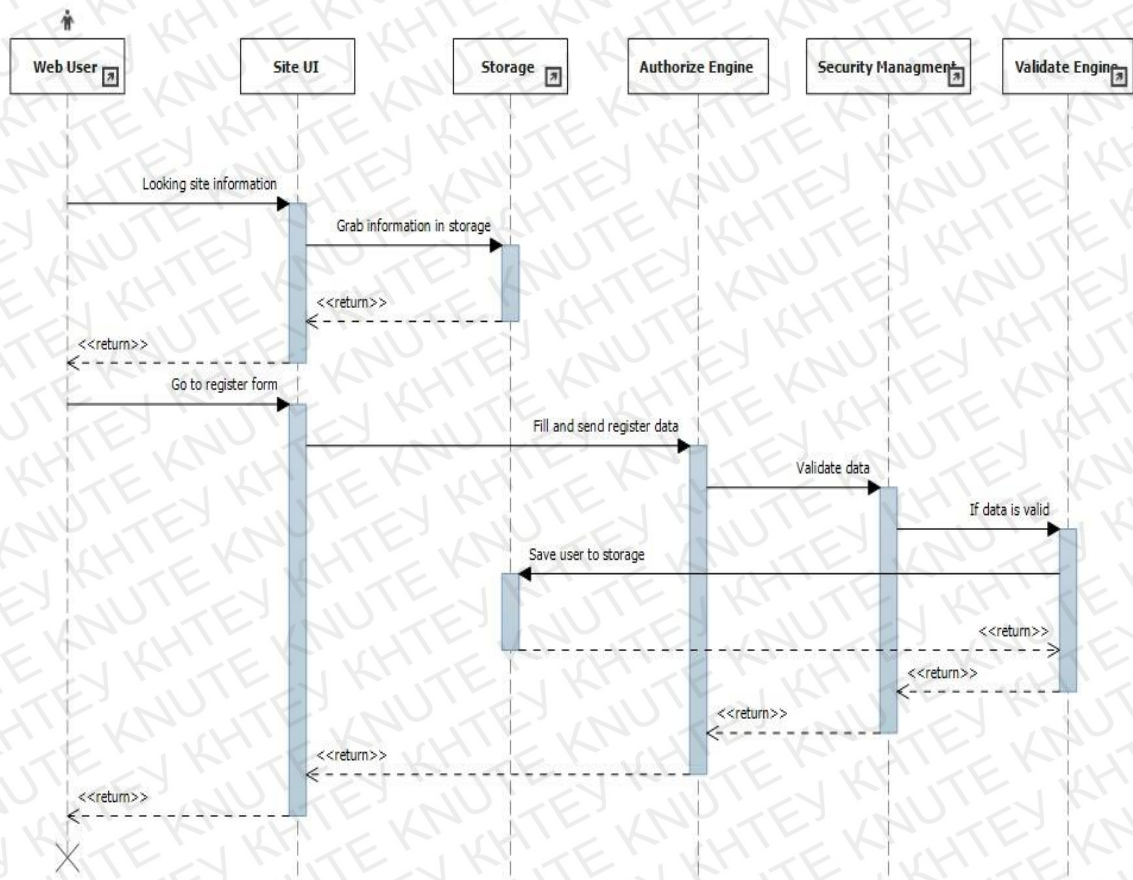


Рис. 3.3. Діаграма послідовностей

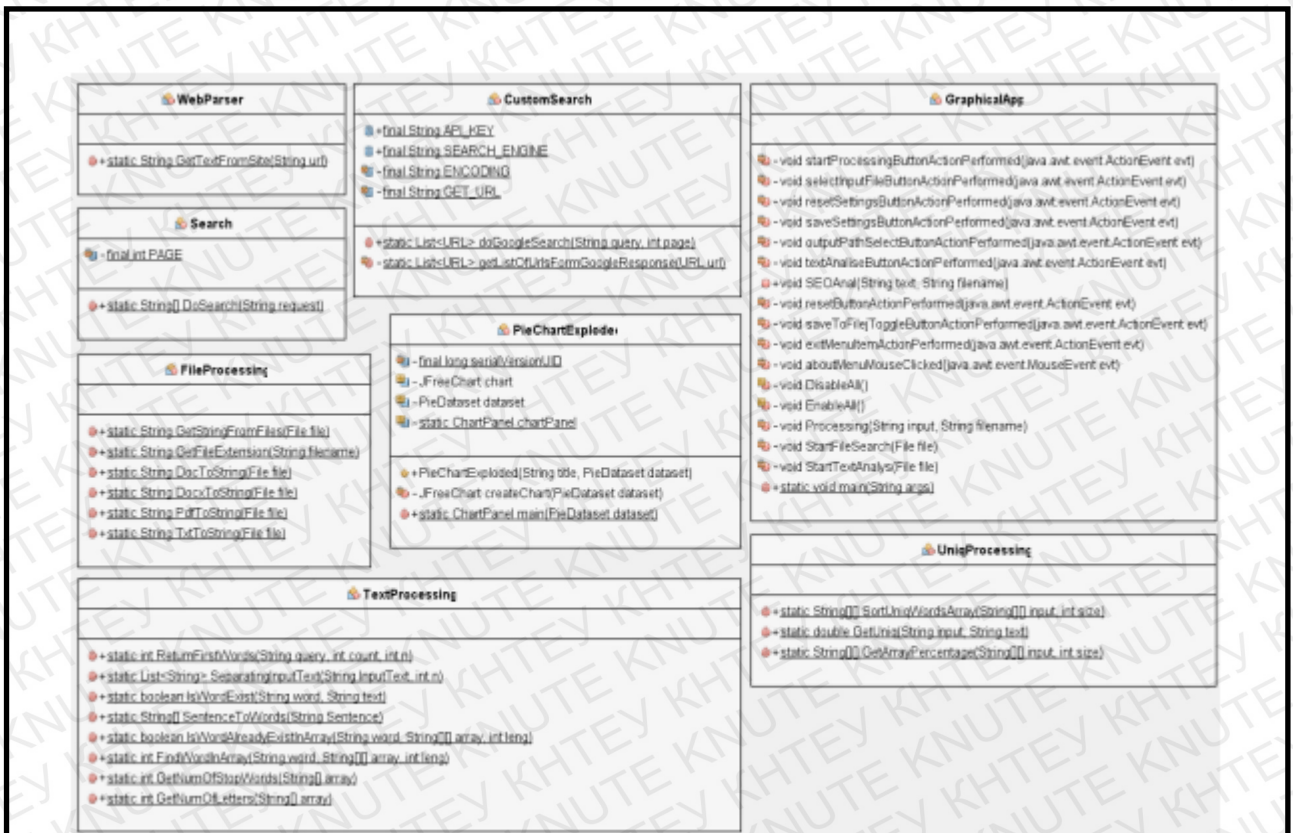


Рис. 3.4. Діаграма класів

3.4 Проектування бази даних системи

Архітектура системи пошуку плагіату в наукових та студентських роботах передбачає, що база даних може бути розміщена як на локальній машині так і на видаленому (remote) сервері. Програмні засоби, реалізовані в системі, формують і відсилають запити до віддаленого серверу БД, який забезпечує виконання запиту і повертає їм результат.

Тому, коли з'являється необхідність змінити модель даних, то єдина достатня дія – відобразити зміни в коді додатку. Визначивши функціональні вимоги до системи пошуку плагіату, описані в підрозділі 3.1 виконаємо проектування БД за допомогою СКБД MySQL.

Структура бази даних виглядає наступним чином:

1) Список документів:

- ID документу;
- хеш-сума документу;
- автор документу;
- навчальна група (кафедра, факультет) автору документу;
- всі слова документу;
- дата додавання документу;
- ID файлу документу.

2) Колекція файлів:

- ID файлу;
- ім'я файлу;
- шлях до файлу;
- розширення файлу.

3) Колекція логів:

- ID логу;
- повідомлення логу;
- дата створення логу;
- тип логу (повідомлення, інформація, помилка).

А оскільки в якості СКБД була вибрана MySQL, то при необхідності, в майбутньому можна буде з легкістю змінити структуру.

3.5 Програмна реалізація та її особливості

Реалізація можливості завантаження файлу студентами на веб-сервіс.

```
<?php
include('nav.php');
if(isset($_POST['submit']))
```

```

{
    $id = $_POST['id'];
    $username = $_SESSION['username'];
    $url = $_POST['url'];
    $file_name = $_FILES['file']['name'];
    $file_type = $_FILES['file']['type'];
    $file_size = $_FILES['file']['size'];
    $file_temp_loc = $_FILES['file']['tmp_name'];
    $file_store = "upload/".$file_name;
    move_uploaded_file($file_temp_loc,$file_store);
    $query15 = "SELECT * FROM upload WHERE a_id = '$id' and
user = '$username'"
    $result = mysqli_query($conn, $query15);
    $num = mysqli_num_rows($result);
    if($num == 1){
        $query14 = "UPDATE upload SET name='$file_name'
,type='$file_type' , size='$file_size' ,a_id='$id'
,user='$username' WHERE a_id = '$id' and user = '$username'";
        mysqli_query($conn , $query14);
        // header("location:home.php");
    }else{
        $query13 = "INSERT INTO upload (name, type, size,
a_id, user, url) VALUES
('$file_name','$file_type','$file_size','$id','$username','$url'
)";
        mysqli_query($conn , $query13);
        // header("location:home.php");
    }
}
}

```

```

if(isset($_GET['id'])){
    $id = $_GET['id'];
    $query12 = "select * from assignment where id = '$id' ";
    $result = mysqli_query($conn, $query12);
    $num = mysqli_num_rows($result);
    if($num > 0)
    {
        while ($row = mysqli_fetch_array($result))
        {
            ?>

```

Реалізація можливості редагувати інформацію у своєму профілі.

```

<?php
include('nav.php');
if(isset($_POST['submit'])){
    $name = $_POST['name'];
    $email = $_POST['email'];
    $role = $_POST['role'];
    $department = $_POST['department'];
    $block = $_POST['block'];
    $gender = $_POST['gender'];
    $age = $_POST['age'];
    $address = $_POST['address'];
    $mobile = $_POST['mobile'];
    $username = $_SESSION['username'];
    $query5 = "UPDATE profile SET
name='$name',email='$email',role='$role',department='$department
',block='$block',gender='$gender',age='$age',address='$address',
mobile='$mobile' WHERE username = '$username'";

```

```

        $result1 = mysqli_query($conn , $query5);
        header('location:profile.php');
    }
    $username = $_SESSION['username'];
    $query4 = "select * from profile where username =
'$username'";
    $result = mysqli_query($conn, $query4);
    while ($row = mysqli_fetch_array($result)){
        ?>

```

3.6 Тестування та налагодження програмного засобу

На кожному кроці розробки алгоритму і програми можуть бути допущені помилки. Причини помилок різноманітні: неадекватне розуміння задачі, неправильно складені специфікації, невдале проектування, погане знання мови програмування, механічні помилки тощо. Найсерйознішими є помилки, зроблені на перших кроках. Тому створену програму треба довести до робочого стану – відтестувати та налагодити.

Тестування та налагодження програми – складна і трудомістка робота. Необхідно відтестувати роботу кожного модуля, усі передачі керування між модулями. Це найпростіше можна зробити за умови, що проектування і програмування виконувались з використанням структур керування та структур даних. Для кожного модуля треба перевірити, чи реалізує він ті функції, які сформульовані в специфікації модуля. Крім цього, треба перевірити логіку (алгоритм) модуля: відтестувати усі операції, усі передачі керування, тобто перевірити усі шляхи потоків інформації. Цим самим перевіряється, як реалізується функція модуля, що визначена в специфікації модуля. Ця робота найпростіше виконується для структурної програми.

						<i>КНТЕУ 121 023-09.МР</i>	
Зм.	Лист	№ докум.	Підпис	Дата			42

У даному програмному засобі ми провели такі етапи тестування:

Тестування правильності. Мета – підтвердити, що функції, описані у специфікації вимог до ПЗ, відповідають очікуванням замовника.

Важливим елементом підтвердження правильності є перевірка конфігурації ПЗ. Конфігурацією ПЗ називають сукупність усіх елементів інформації, що виробляються в процесі конструювання ПЗ.

Перевірка конфігурації гарантує, що всі елементи конфігурації ПЗ правильно розроблені, враховані і достатньо деталізовані для підтримки етапу супроводу в життєвому циклі ПС.

Тестування відновлення. На даному етапі було проведення тестування на скільки швидко можна відновити роботу програмного засобу після атаки хакерів чи збою системи.

Стресове тестування. На даному етапі було проведено тестування системи на усі можливі нештатні дії користувачів.

Тестування продуктивності. На даному етапі програмний засіб було протестовано на продуктивність та швидкодію.

Наступні етапи – це етапи Usability Testing, дослідження, що виконується з метою визначення зручності програмного засобу для кінцевого користувача:

- функціональне тестування;
- тестування посилань;
- навігаційне тестування;
- тестування контенту;
- тестування інтерфейсу користувача;
- тестування сумісності;
- кросплатформове тестування;
- кросбраузерні тестування;

- оптимізація розміру всіх зображень;
- бета-тестування.

Налагодження програми. Налагодження є наслідком успішного тестування. Це означає, що якщо тестовий варіант виявляє помилку, то процес налагодження знищує її.

3.7 Особливості використання та впровадженню програмного засобу

Після того, як програмний засіб пройшов успішно усі етапи тестування, він є повністю готовим до використання.

Даний програмний засіб призначений для пошуку плагіату у програмному коді, який можна використовувати у власному користуванні так і викладачам у вищих навчальних закладах. Найпоширеніші користувачі це викладачі та студенти також за бажанням працівники ІТ компаній.

При розробці програмного забезпечення дипломного проекту була реалізована стандартна для веб-додатка функціональність, така як: реєстрація користувача, механізми автентифікації та авторизації, набір CRUD операцій для наявних сутностей в системі.

На рисунку зображено головну сторінку веб-ресурсу, на цій сторінці знаходиться можливість логінації або реєстрації.



Рис. 3.5. Головна сторінка веб-ресурсу

На вкладці «Sign Up» викладач або студент мають можливість зареєструватись у системі.

Регістрація

teacher-1

.....

teacher-1

teacher@teacher.com

Викладач

Software engineering

Information Technology

Регістрація

Рис. 3.6. Регістрація викладача

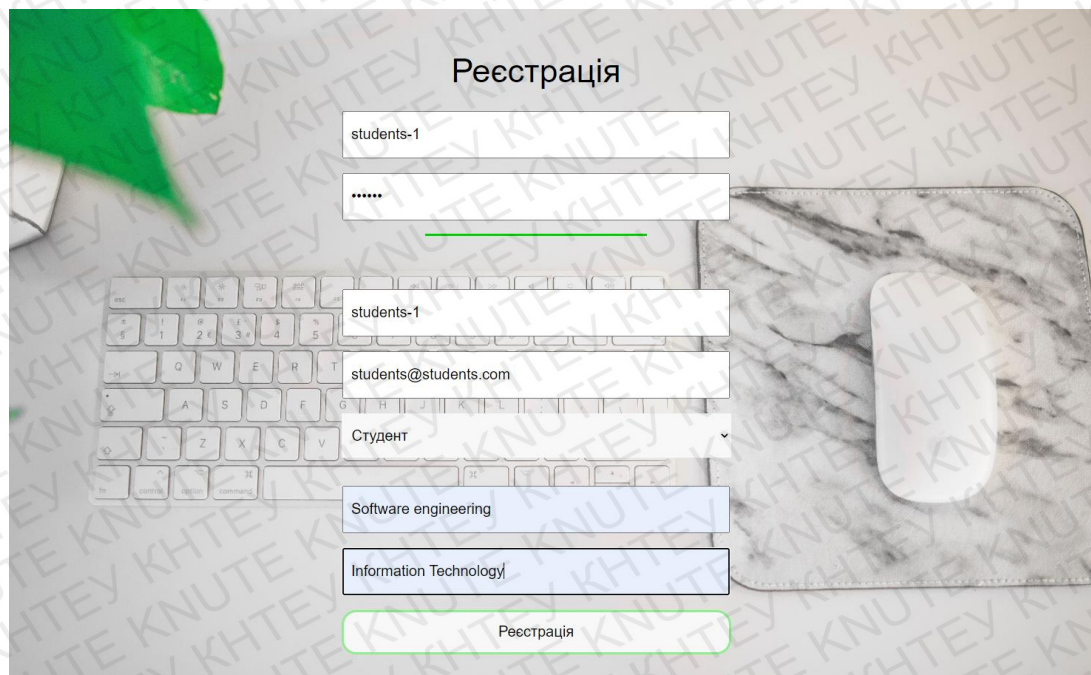


Рис. 3.7. Реєстрація студента



Рис. 3.8. Логінація викладача

Перейшовши на вкладку «Профіль», перед нами появляється уся доступна інформація про викладача з можливістю редагування.

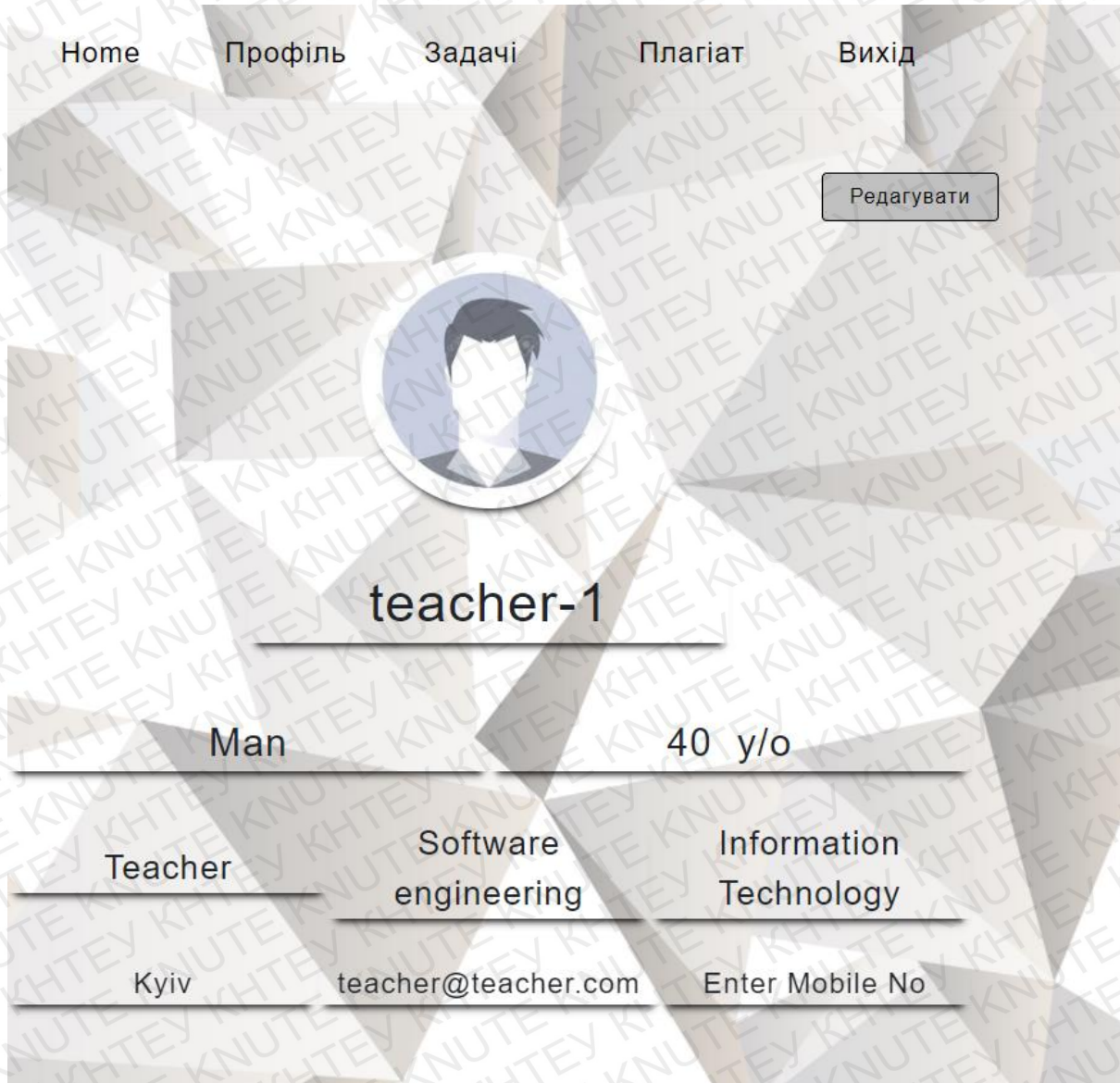


Рис. 3.9. Вкладка «Профіль»

Перейшовши на вкладку «Задачі» вчитель має змогу створити задачу для учнів своєї групи.

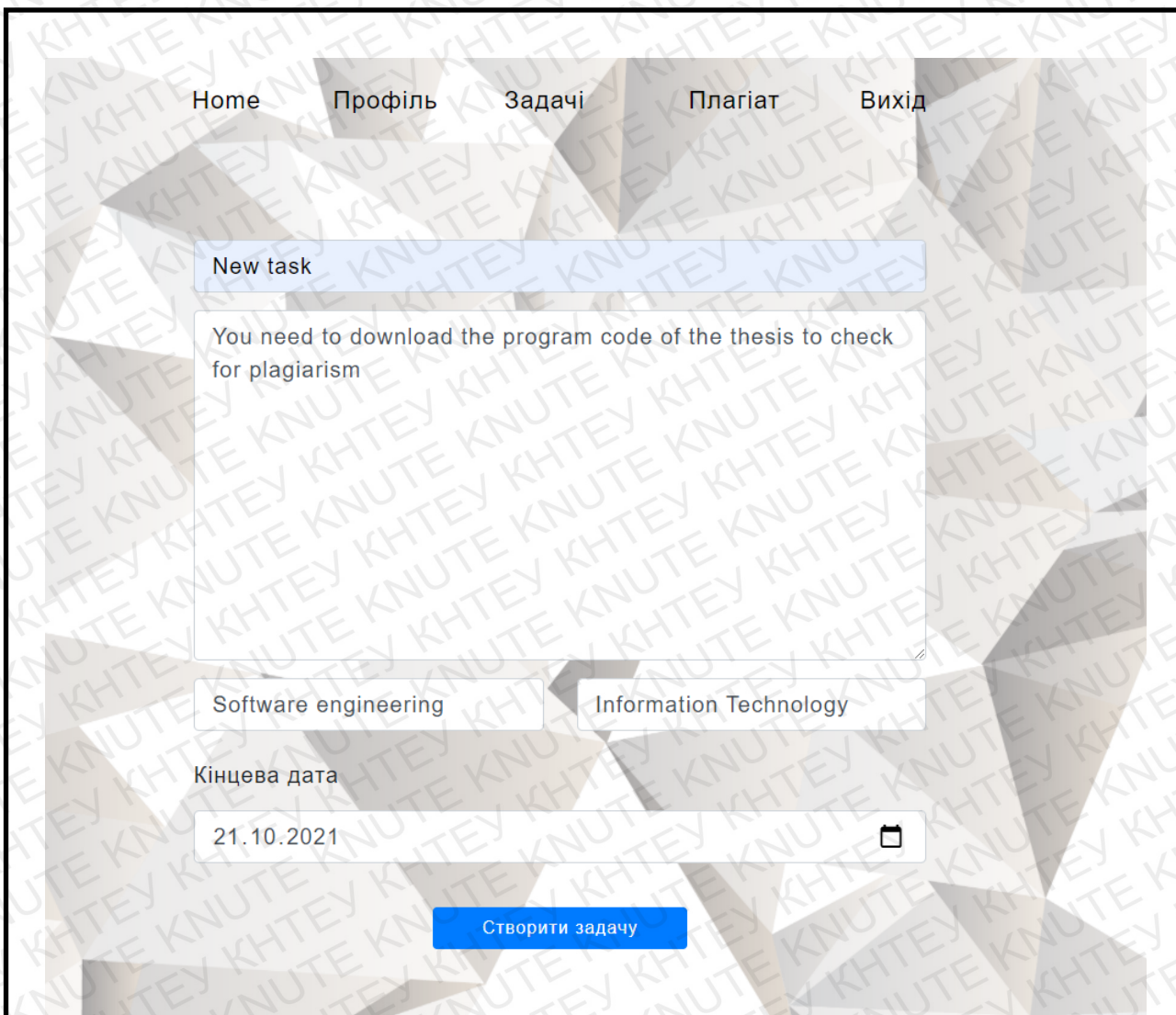


Рис. 3.10. Створення нової задачі

Після того, як студент увійшов до системи, на головній сторінці він отримає задачу, яку йому потрібно виконати.

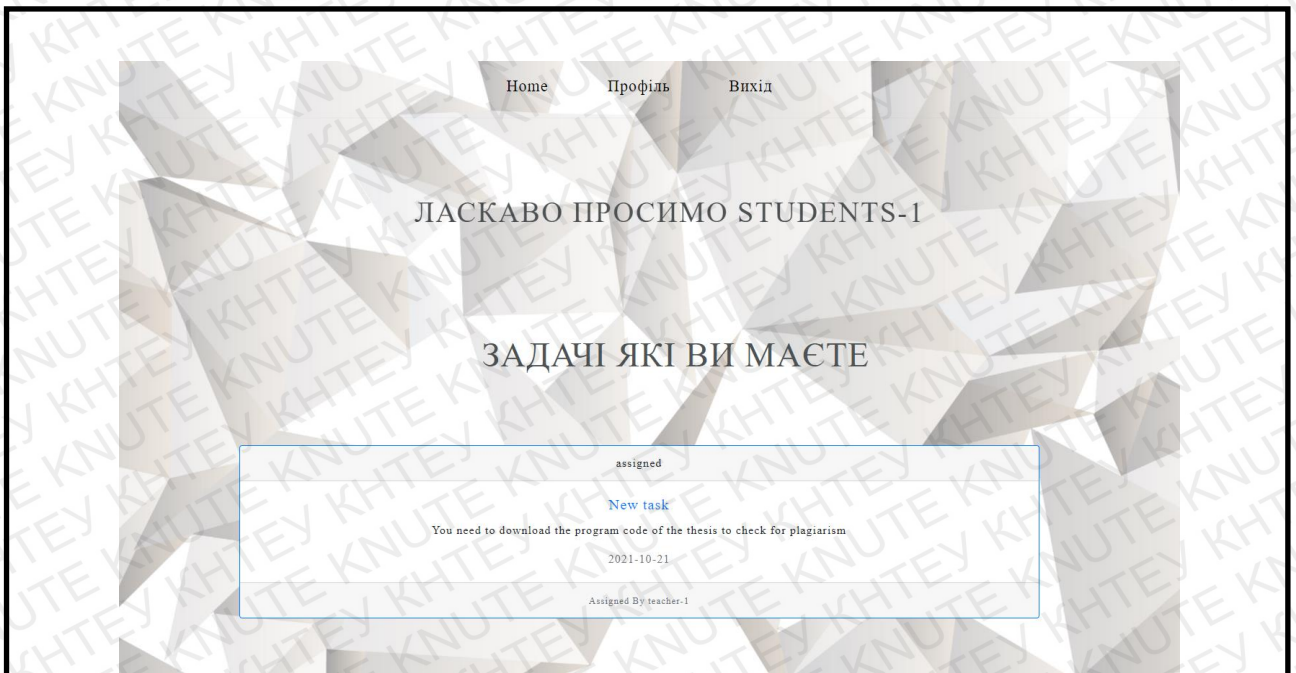


Рис. 3.11. Створені задачі для студента

Після цього студент заходить у задачу і завантажує відповідний файл.

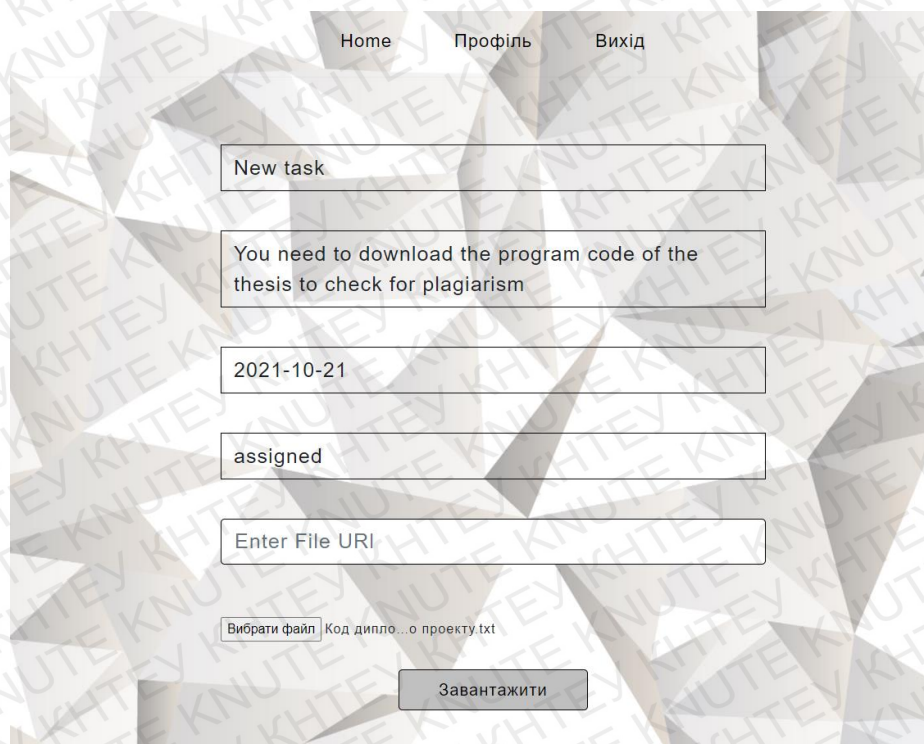


Рис. 3.12. Вікно завантаження відповідного файлу

Зайшовши у відповідну задачу, викладач має можливість переглянути, який файл прикріпив студент, також він має можливість відправити файл на перевірку на плагіат.

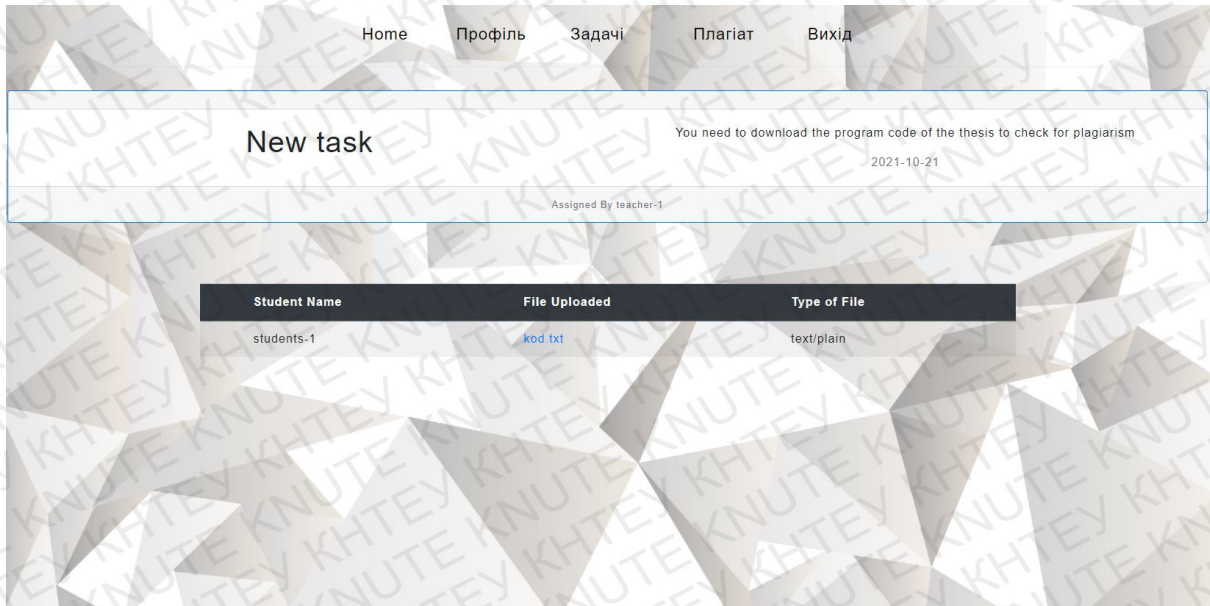


Рис. 3.12. Вікно задачі з прикріпленими файлами



Рис. 3.13. Вікно результату перевірки на плагіат

Для правильного використання можливостей веб-додатка користувачам варто пам'ятати декілька правил. Необхідно уникати використання кирилиці в програмному коді. Хоча веб-додаток попередньо видаляє заборонені символи, але це позначається на часі виконання перевірки. Також в назвах фалів і директорій варто використовувати лише латинські символи, це дозволить уникнути ситуацій, коли система в результатах покаже некоректні символи в назвах проєктів.

Архітектура веб-додатка була спроектована таким чином, щоб дозволити в майбутньому без особливих труднощів розширити функціональність системи або повністю змінити реалізацію деяких компонентів.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В ході виконання випускної кваліфікаційної роботи був отриманий повнофункціональний веб-сервіс, повністю готовий до застосування. Даний сервіс орієнтований на викладачів, які зацікавлені у перевірці програмного коду своїх студентів. З його допомогою викладачі зможуть отримувати необхідну інформацію про програмний код своїх студентів.

В ході даної кваліфікаційної роботи було здійснено аналіз ринку веб-ресурсів у сфері перевірки на плагіат програмного коду. Під час огляду статистичних даних та вже існуючих систем було виявлено, що дана тема є актуальною, оскільки на українському ринку не багато існує якісних сервісів. При огляді технологій для створення майбутньої системи стало зрозуміло, що найкращим рішенням буде використовувати мову програмування PHP.

Опісля було сформовано специфікацію вимог, яка містила точно сформульоване технічне завдання та вичерпний опис усіх функцій, необхідних у майбутній системі. Було вирішено, що система міститиме великі переваги через присутність у ній таких функцій як реєстрація викладача та студента. Згодом було обрано клієнт-серверну архітектуру, спроектовано інтерфейси користувача.

Використовуючи зібрану інформацію та обрані проектні рішення було розроблено систему, яка повністю відповідає вимогам, описаним у специфікації вимог. Весь функціонал розробленої системи, описаний у специфікації вимог, успішно реалізований. Варто зауважити, що ідея створення веб-сервісу може активно розвиватись та доповнюватись новим функціоналом, що забезпечить системі подальший розвиток.

					<i>КНТЕУ 121 023-09.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Проектування та розробка web-сервісу для перевірки коду на плагіат	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.	Криворучко О.В.			01.11.2021		<i>ВП</i>	52	56
Керівник	Савченко Т.В.			01.11.2021				
Гарант	Токар В.В.			01.11.2021				
Розробив	Лісовець В.І.			01.11.2021	Висновки та пропозиції	Факультет інформаційних технологій 2м курс, 023 група		

В роботі проведено аналіз предметної області, існуючих варіантів розв'язання досліджуваної задачі. Проведено порівняльний аналіз аналогічних веб-ресурсів. Був проведений аналіз методів розв'язання задачі.

В роботі було досліджено та проаналізовано найбільш важливу і актуальну інформацію щодо розробки веб-ресурсів, а саме основні принципи створення веб-ресурсів, їх структуру і функціональність, взаємодію основних компонентів. Також були розглянуті новітні та найбільш перспективні веб-технології, які з успіхом вже використовуються користувачами по всьому світі.

Для кращої реалізації власного проекту було проведено огляд новітніх інформаційних технологій, та програмного забезпечення, що могли б бути використані для розробки сучасного, ефективного та головне конкурентоздатного програмного продукту.

					<i>КНТЕУ 121 023-09.MP</i>	
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		53

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. M. Palmer. (2018, May) Chimpsky plagiarism detection. [Online]. Available: <http://chimpsky.uwaterloo.ca/>
2. Roy C. K. and Cordy J. R.. A survey on software clone detection research, Tech. Rep. 2017-541, School of Computing, Queen's University, Kingston, Ontario, Canada, 2017., pages 43-59.
3. Chanchal K. Roy, James R. Cordy, Rainer Koschke, Comparison and Evaluation of Code Clone Detection Techniques and Tools: A Qualitative Approach. Science of Computer Programming Volume 74, Issue 7, 1 May 2019, pages 470–495.
4. Mohand-Said Hacid, Zbigniew W Ras, Shusaku Tsumoto. Foundations of Intelligent Systems – 15th International Symposium ISMIS 2015, Saratoga Springs, NY, USA, May 25-28, 2015, Proceedings, pages 641-656.
5. Merlo E., Detection of Plagiarism in University Projects Using Metrics-based Spectral Similarity, 2017, pages 3-10
6. Огляд алгоритмів виявлення плагіату у вихідних кодах програм URL: <http://rain.ifmo.ru/cat/data/theory/unordered/plagiarism-2006/article.pdf>
7. Г.В. Стрельченко, Т.А. Лепихін, К.С. Лабзо Використання різних способів виявлення плагіату вихідних кодів у навчальному процесі // Сучасні інформаційні технології та ІТ-освіта, 2019. — Т. 1,— No 11. — С. 211-214.
8. Bellon S. et al. Comprasion and evaluation of clone detection tools // IEEE Transactions on software engineering. – 2019. – Т. 33. - №.9.

<i>КНТЕУ 121 023-09.МР</i>										
Зм.	Аркуш	№ докум.	Підпис	Дата	Проектування та розробка web-сервісу для перевірки коду на плагіат			Стадія	Аркуш	Аркушів
Зав. каф.	Криворучко О.В.							СВД	54	56
Керівник	Савченко Т.В.				Список використаних джерел			Факультет інформаційних технологій 2м курс, 02з група		
Гарант	Токар В.В.									
Розробив	Лісовець В.І.									

9. Roy C. K., Cordy J. R. survey on software clone detection reseach // Queen School of Computing TR. – 2020. – Т. 541. - №.115.
10. [Электронный ресурс] — Режим доступа : <https://codequiry.com/>
11. [Электронный ресурс] — Режим доступа : <http://theory.stanford.edu/~aiken/moss/>
12. [Электронный ресурс] — Режим доступа : <https://github.com/jplag/JPlag>
13. [Электронный ресурс] — Режим доступа : <http://html5ru.com/css-i-html.html>
14. [Электронный ресурс] — Режим доступа : <http://html5ru.com/preimushhestva-html5.html>
15. [Электронный ресурс] — Режим доступа : <https://php-academy.kiev.ua/uk/blog/html-and-css-without-guilt>
16. [Электронный ресурс] — Режим доступа : https://css.in.ua/article/shcho-take-html_10
17. [Электронный ресурс] — Режим доступа : <https://studfiles.net/preview/1624162/>
18. [Электронный ресурс] — Режим доступа : <https://sites.google.com/site/-webtehnologiietawebdizajn/mova-javascript-ta-ieie-mozlivosti>
19. [Электронный ресурс] — Режим доступа : <https://astwellsoft.com/uk/blog/tehnology/javascript.html>
20. [Электронный ресурс] — Режим доступа : <http://web.if.ua/15-perevagi-php.html>
21. [Электронный ресурс] — Режим доступа : <http://it-devgroup.com/blog/blog-razrobotka/obzor-laravel-5.-osnovnyie-preimushhestva-i-nedostatki/>

22. [Електронний ресурс] — Режим доступу :
<http://savelink.org.ua/laravel-lider-sered-php-frejmworkiv-shvalenij-rozrobnikami/>

23. Чверкун Денис. Основні властивості каскадної моделі. 2018 рік. –
[Електронний ресурс] – Режим доступу: <https://geekbrains.ru/posts/waterfall>

					<i>КНТЕУ 121 063-19.МР</i>	56
Зм.	Лист	№ докум.	Підпис	Дата		

ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПРОГРАМНОГО ПРОДУКТУ ДЛЯ ПОШУКУ ПЛАГІАТУ У ПРОГРАМНОМУ КОДІ

1. Вступ

Програмне забезпечення являє собою веб-сервіс для пошуку плагіату у програмному коді.

2. Підстави для розробки

Підставою для розробки є необхідність виконання завдання зі спецкурсу «».

3. Призначення розробки

Розроблений програмний ресурс призначений для покращення вивчення ІТ технологій і перевірки роботи студентів на плагіат.

4. Вимоги до програми чи програмного продукту

4.1. Вимоги до функціональних характеристик

Програмний продукт повинен володіти наступними функціональними характеристиками:

- кросбраузерність;
- адаптивний веб-дизайн;
- можливість реєстрації викладача;
- можливість реєстрації студента;
- можливість перевірки програмного коду на плагіат.

4.2. Вимоги до надійності

Надійність програмного засобу має бути забезпечена шляхом:

- забезпечення неперервного живлення технічних засобів;

					<i>КНТЕУ 121 023-09.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.	Криворучко О.В.			20.03.2021	Проектування та розробка web-сервісу для перевірки коду на плагіат	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник	Савченко Т.В.			20.03.2021		<i>ТЗ</i>	<i>57</i>	<i>56</i>
Гарант	Токар В.В.			20.03.2021		Факультет інформаційних технологій 2м курс, 02з група		
Розробив	Лісовець В.І.			20.03.2021				
					Технічне завдання			

- контролю вхідних даних – перевірка на коректність та повноту всіх даних, що вводяться користувачами;
- ведення архіву дій користувачів;
- відновлення роботи системи після відмови;
- періодичне створення резервних копій;
- виконання вимог щодо захисту інформації.

4.3. Умови експлуатації

Необхідною умовою для експлуатації веб-сервісу є наявність Інтернету на комп'ютері чи іншому пристрої.

4.4. Вимоги до складу і параметрів технічних засобів

Для виконання ролі технічного засобу необхідний персональний комп'ютер чи інший пристрій з встановленим веб-браузером.

4.5. Вимоги до інформаційної і програмної сумісності

Для успішного функціонування програмного продукту ніяких додаткових програмних засобів, крім браузера, не вимагається.

4.6. Вимоги до маркування і упаковки

Не ставиться ніяких вимог до маркування та упаковки для програмного забезпечення.

4.7. Вимоги до транспортування і збереження

Транспортування програмного продукту здійснюється мережею Інтернет.

5. Вимоги до програмної документації

До складу програмної документації повинно входити:

- технічне завдання;
- керівництво користувачу;
- інструкція для адміністратора сайту.

6. Техніко-економічні показники

Техніко-економічні показники веб-сервісу відсутні, оскільки розробка є не комерційною і використовується абсолютно безкоштовно.

7. Стадії і етапи розробки

Стадії створення програмного продукту:

- формування вимог до розробки веб-ресурсу;
- розробка концепцій програмного засобу (розробка варіанта концепції, оформлення звіту про виконання роботи);
- технічне завдання (розробка технічного завдання, затвердження технічного завдання);
- ескізний проект (розробка попередніх рішень);
- технічний проект;
- введення в дію (підготовка веб-сайту, проведення попередніх випробувань, проведення експлуатації);

Строки реалізації

Датою початку робіт необхідно вважати дату затвердження цього ТЗ та договору на розробку між замовником та головним розробником системи. В зв'язку з чим вказані строки реалізації в днях без уточнення конкретних дат.

Склад і зміст робіт зі створення системи

Організаційна структура має відповідати наступним вимогам:

- дотримання принципу єдиноначальності і легалізацію персональної відповідальності;
- забезпечення надійного функціонування системи;
- раціональні інформаційні зв'язки.

На етапі розробки технічного завдання мають бути виконані перераховані нижче роботи:

- постановка завдання;

- визначення і уточнення вимог до технічних засобів;
- визначення вимог до веб-сайту;
- визначення стадій, етапів, термінів розробки і документації.

Узгодження і затвердження технічного завдання

На етапі розробки програми має бути виконана робота по програмуванню (кодуванню) і відладці програми. На етапі розробки програмної документації має бути виконана розробка програмних документів відповідно до вимог до складу документації. На етапі випробувань програми мають бути виконані перераховані нижче види робіт:

- розробка, узгодження і твердження і методики випробувань;
- проведення приймально-здавальних випробувань;
- коректування за результатами випробувань.

На етапі підготовки і передачі програми має бути виконана робота з підготовки і передачі веб-сайту і документації в експлуатацію на об'єктах замовника.

8. Порядок контролю і приймання

Остаточні випробування повинні проводитися на об'єкті замовника в обумовлені терміни згідно розробленої виконавцем і погодженої замовником програми і методики випробувань.

На підставі протоколу проведення випробувань виконавець спільно із замовником підписує акт приймання-здачі веб-сайту в експлуатацію.

ДОДАТКИ

ЛІСТИНГ ПРОГРАМИ

```
<?php
//dependencies and autoload
include_once( getcwd().'./autoload.php');
include_once( __DIR__.'./vendor/php-plagiarism-checker/
autoload.php ');
include('nav.php');
if(isset($_GET['id'])){
    $id = $_GET['id'];
    // echo $id;
    $query100 = "select * from upload where a_id = $id ";
    $result = mysqli_query($conn, $query100);
    $num = mysqli_num_rows($result);
    if($num > 0)
    {
        while($row = mysqli_fetch_array($result))
        {
            try{
                $clCloud = new CopyleaksCloud($email, $apiKey,
                Products::Education);
            }catch(Exception $e){
                echo "<Br/>Failed to connect to Copyleaks Cloud with
                exception: ". $e->getMessage();
                die();
            }
            //validate login token
            if(!isset($clCloud->loginToken) || !$clCloud->loginToken-
            >validate()){
                echo "<Br/><strong>Bad login credentials</strong>";
                die();
            }
        }
    }
}
```

```

    }

    echo "<BR/><BR/><strong>Results:</strong>";
    $results = $process->getResult();
    // Print the results
    if ($results){
        echo $results[0];
    }
    else{
        echo "Plagiarism not found";
    }
    echo "</div>";
    echo "<br>";
} catch (Exception $e) {
    echo "<br/>Failed with exception: ". $e->getMessage();
}
}
}
}
?>

<?php
//dependencies and autoload
include_once( getcwd().'./autoload.php');
include_once( __DIR__.'./vendor/copyleaks/php-plagiarism-
checker/autoload.php');

$additionalHeaders = array($clConst['HTTP_CALLBACK'].':
http://your.website.com/callbacks/');

$process = $clCloud-
>createByURL('https://www.copyleaks.com',$additionalHeaders);try
{

```

```
        $clCloud = new CopleaksCloud($email, $apiKey,
Products::Businesses);
    }catch(Exception $e){
        echo "<Br/>Failed to connect to Copleaks Cloud with
exception: ". $e->getMessage();
        die();
    }
    //validate login token
    if(!isset($clCloud->loginToken) || !$clCloud->loginToken-
>validate()){
        echo "<Br/><strong>Bad login credentials</strong>";
        die();
    }
    echo "<Br/><strong>Logged in successfully</strong><Br/>";
    try{
        echo "<BR/><strong>Process created!</strong> (PID = '" .
$process->processId . "') - You will get notified with a callback
soon";
    }catch(Exception $e){
        echo "<br/>Failed with exception: ". $e->getMessage();
    }
    //build table from PHP array
    function build_table($array){
        // start table
        $html = '<table>';
        // header row
        $html .= '<tr>';
        foreach($array[0] as $key=>$value){
            $html .= '<th>' . $key . '</th>';
        }
    }
}
```

```

        $html .= '</tr>';
    // data rows
    foreach( $array as $key=>$value){
        $html .= '<tr>';
        foreach($value as $key2=>$value2){
            $value2 = is_array($value2) ? json_encode($value2)
: $value2;
            $html .= '<td>' . @$value2 . '</td>';
        }
        $html .= '</tr>';
    }
    // finish table and return it
    $html .= '</table>';
    return $html;
}

//print process list as HTML table
if(isset($plist,$plist['response']) &&
count($plist['response'])>0)
    echo build_table($plist['response']);
?>

<?php
session_start();

// header('location:login.php');
$conn = mysqli_connect('localhost','root','root');
if($conn){
    // echo "Connection sucessfull";
}

```

```
else{
    echo "No Connection established";
}
if(isset($_POST['submit']))
{
    mysqli_select_db($conn, 'wt');
    $username = $_POST['username'];
    $password = $_POST['password'];
    $query1 = "select * from profile where username =
'$username' && password = '$password'";
    $result = mysqli_query($conn, $query1);
    $num = mysqli_num_rows($result);
    $row = mysqli_fetch_array($result);
    if($num == 1){
        $_SESSION['username'] = $username;
        $_SESSION['role'] = $row['role'];
        header('location:home.php');
    }
    else{
        header('location:login.php');
    }
}
?>

<?php
namespace App\Http\Controllers;
use Session;
use Redirect;
use Illuminate\Http\Request;
```

```

use Illuminate\Support\Facades\DB;
class LoginController extends Controller
{
    public function login(Request $request)
    {
        $email = $request->email;
        $password = $request->password;
        $db = DB::table('users')->where('email', $email)-
>first();
        if($db->email == $email && $db->password ==
$password){
            $request->session()->put('user_id', $db->id);
            return redirect('dashboard');
        }else{
            Session::flash ( 'message', "Invalid
Credentials , Please try again." );
            return Redirect::back ();
        }
    }
    public function logout(Request $request)
    {
        $request->session()->forget('user_id');
        Session::flash ( 'message', "Logout Successful." );
        return redirect('/');
    }
}

```

```
<?php
```

```
include('nav.php');
```

```
if(isset($_POST['submit']))
```

```
{
```



```
$id = $_POST['id'];
$username = $_SESSION['username'];
$url = $_POST['url'];
$file_name = $_FILES['file']['name'];
$file_type = $_FILES['file']['type'];
$file_size = $_FILES['file']['size'];
$file_temp_loc = $_FILES['file']['tmp_name'];
$file_store = "upload/".$file_name;
move_uploaded_file($file_temp_loc,$file_store);
$query15 = "SELECT * FROM upload WHERE a_id = '$id' and
user = '$username'";
$result = mysqli_query($conn, $query15);
$num = mysqli_num_rows($result);
if($num == 1){
    $query14 = "UPDATE upload SET name='$file_name'
,type='$file_type' , size='$file_size' ,a_id='$id'
,user='$username' WHERE a_id = '$id' and user = '$username'";
    mysqli_query($conn , $query14);
    // header("location:home.php");
}else{
    $query13 = "INSERT INTO upload (name, type, size,
a_id, user, url) VALUES
('$file_name','$file_type','$file_size','$id','$username','$url'
)";
    mysqli_query($conn , $query13);
    // header("location:home.php");
}
}
if(isset($_GET['id'])){
    $id = $_GET['id'];
    $query12 = "select * from assignment where id = '$id' ";
    $result = mysqli_query($conn, $query12);
```

```
$num = mysqli_num_rows($result);
if($num > 0)
{
    while ($row = mysqli_fetch_array($result))
    {
?>

<?php
$client = new http\Client;
$request = new http\Client\Request;
$body = new http\Message\Body;
$body->addForm(NULL, array(
    array(
        'name' => 'file_path',
        'type' => 'application/octet-stream',
        'file' => 'hodlistemp.html',
        'data' => null
    )
));
$request->setRequestUrl('https://plagspotter-duplicate-content-checking.p.rapidapi.com/v2.add-file-for-checking');
$request->setRequestMethod('POST');
$request->setBody($body);

$request->setHeaders(array(
    'x-rapidapi-host' => 'plagspotter-duplicate-content-checking.p.rapidapi.com',
    'x-rapidapi-key' => 'b5155e42c6msh87aca93b5da4ec3p1aafc7jsn3d4cbe2b75d4',
    'authorization' => 'Basic TmlzaGFkMDA6TmlzaGFkQDEyMw==',
    'content-type' => 'multipart/form-data'
```

```
));  
$client->enqueue($request)->send();  
$response = $client->getResponse();  
echo $response->getBody();
```

```
<?php
```

```
include('nav.php');
```

```
?>
```

```
<?php
```

```
if(isset($_GET['id'])){
```

```
    $id = $_GET['id'];
```

```
    echo $id;
```

```
    $query100 = "select * from upload where a_id = $id ";
```

```
    $result = mysqli_query($conn, $query100);
```

```
    $num = mysqli_num_rows($result);
```

```
    if($num > 0)
```

```
    {
```

```
        while($row = mysqli_fetch_array($result))
```

```
        {
```

```
            echo $row['url'];
```

```
        }
```

```
    }
```

```
?>
```