

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Метод покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент групи КН-20-2 Чистяков Валерій ЧИСТЯКОВ
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ
Керівник: к.т.н., доц. каф. КН Пасічник Олександр ПАСІЧНИК
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ
Нормоконтроль: к.т.н., доц. каф. КН Багрий Руслан БАГРІЙ
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор Бармак Олександр БАРМАК
Підпис Ім'я, ПРІЗВИЩЕ
14 червня 2024 р.

Хмельницький 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет інформаційних технологій
Кафедра комп'ютерних наук
Освітній ступінь бакалавр
Галузь знань 12 – Інформаційні технології
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук


(підпис)

д.т.н., професор Олександр БАРМАК

« 16 » лютого 2024 року


**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Метод покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми»
2. Завдання видано студенту Валерію ЧИСТЯКОВУ
(Ім'я, прізвище)
3. Керівник роботи доцент кафедри КН Олександр ПАСІЧНИК
(посада, ім'я, прізвище)
4. Затверджено наказом університету від « 15 » лютого 2024 р. № 8
5. Дата видачі завдання студенту: « 16 » лютого 2024 р.
6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи – покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми. Провести аналіз методів формування вмісту навчальних курсів та підходів щодо їх покращення; провести аналіз інтелектуальних методів для визначення найкращого вмісту навчальних курсів; реалізувати метод покращення вмісту навчальних курсів за генетичним алгоритмом; виконати програмну реалізацію розробленого методу та освітньої вебсистеми для покращення вмісту навчальних курсів за генетичним алгоритмом; виконати експериментальне тестування розробленого методу.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником, складання календарного графіка виконання роботи	січень 2024	виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	лютий 2024	виконано
3	Проектування та розробка загальної архітектури програмного забезпечення, інтерфейсу користувача, вибір засобів реалізації програмного забезпечення	березень 2024	виконано
4	Створення та тестування програмного забезпечення	квітень 2024	виконано
5	Написання пояснювальної записки, урахування зауважень керівника, оформлення згідно вимог	травень 2024	виконано
6	Розробка презентаційних матеріалів та попередній захист кваліфікаційної роботи	травень 2024	виконано
7	Отримання відгуку керівника, рецензії, перевірка на плагіат, нормоконтроль	червень 2024	виконано
8	Підготовка до захисту та захист кваліфікаційної роботи бакалавра	червень 2024	виконано

Виконавець: студент групи КН-20-2  Валерій ЧИСТЯКОВ
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ

Керівник: к.т.н., доц. каф. КН  Олександр ПАСІЧНИК
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

Анотація

Тема кваліфікаційної роботи бакалавра: «Метод покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-20-2 Валерій ЧИСТЯКОВ

Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КН Олександр ПАСІЧНИК


Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
70	23	8	47	7

Метою кваліфікаційної роботи бакалавра є покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми. Для досягнення мети визначено такі задачі дослідження: провести аналіз методів формування та покращення вмісту навчальних дисциплін; провести аналіз інтелектуальних методів для визначення найкращого вмісту навчальних курсів; реалізувати метод покращення вмісту навчальних курсів за генетичним алгоритмом; виконати програмну реалізацію методу покращення вмісту навчальних курсів з адаптацією генетичного алгоритму; провести експериментальне тестування методу та його програмної реалізації; виконати оцінювання покращення вмісту навчальних курсів.

Результатом виконання кваліфікаційної роботи бакалавра є розробка методу покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми.

Ключові слова: генетичний алгоритм, освітня вебсистема, навчальний матеріал, покращення вмісту, ефективність навчання, персоналізація навчання.

Виконавець: студент групи КН-20-2  Валерій ЧИСТЯКОВ
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ

Анотація

Тема кваліфікаційної роботи бакалавра: «Метод покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-20-2 Валерій ЧИСТЯКОВ

Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КН Олександр ПАСІЧНИК

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
70	23	8	47	7

Метою кваліфікаційної роботи бакалавра є покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми. Для досягнення мети визначено такі задачі дослідження: провести аналіз методів формування та покращення вмісту навчальних дисциплін; провести аналіз інтелектуальних методів для визначення найкращого вмісту навчальних курсів; реалізувати метод покращення вмісту навчальних курсів за генетичним алгоритмом; виконати програмну реалізацію методу покращення вмісту навчальних курсів з адаптацією генетичного алгоритму; провести експериментальне тестування методу та його програмної реалізації; виконати оцінювання покращення вмісту навчальних курсів.

Результатом виконання кваліфікаційної роботи бакалавра є розробка методу покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми.

Ключові слова: генетичний алгоритм, освітня вебсистема, навчальний матеріал, покращення вмісту, ефективність навчання, персоналізація навчання.

Виконавець: студент групи КН-20-2 Валерій ЧИСТЯКОВ
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ

Зміст

Перелік скорочень	4
Вступ.....	5
Розділ 1 Характеристика предметної області та постановка задачі.....	7
1.1 Аналіз інформаційних моделей пов’язаних з освітніми вебсистемами	7
1.2 Огляд сучасних підходів щодо формування вмісту навчальних курсів.....	10
1.3 Аналіз існуючих освітніх вебсистем та наукових рішень	13
1.4 Мета та завдання кваліфікаційної роботи.....	16
Розділ 2 Метод покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми	18
2.1 Загальна структура методу	18
2.2 Імплементация генетичного алгоритму в освітню вебсистему	19
2.2.1 Загальна структура генетичного алгоритму	19
2.2.2 Особливості імплементации генетичного алгоритму для реалізації методу генерації курсів.....	21
2.3 Особливості використання спеціалізованих програмних компонентів	25
2.4 Проектування інформаційної структури навчального вебресурсу з генерацією курсів на основі генетичного алгоритму	25
2.5 Підготовка робочих вхідних даних для системи формування курсів.....	28
2.6 Проектування функціональної структури освітньої вебсистеми з автоматичною генерацією курсів	29
2.7 Розробка архітектури навчального вебресурсу з функцією генерації курсів на основі генетичного алгоритму	34
2.8 Критерії оцінювання покращення вмісту навчальних курсів.....	38
2.9 Висновки до розділу 2.....	39
Розділ 3 Програмна реалізація методу генерації курсів.....	41
3.1 Вибір засобів розробки програмної реалізації методу	41
3.2 Структура та функціональне призначення програмних складових системи ..	42
3.3 Особливості реалізації генерації курсів	44
3.4 Тестування працездатності та ефективності генерації курсів	49

	3
3.4 Оптимізація гіперпараметрів.....	54
3.5 Покращення якості вмісту навчальних курсів	59
3.6 Висновки до розділу 3.....	62
Загальні висновки.....	64
Перелік посилань.....	66
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
БД	База даних
ГА	Генетичний алгоритм
ІС	Інформаційна система
ШНМ	Штучна нейронна мережа
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки
СКБД	Система керування базами даних
JS	JavaScript
HTML	HyperText Markup Language
CSS3	Cascading Style Sheets 3
PHP	Personal Home Page Tools

Вступ

Кваліфікаційна робота бакалавра присвячена покращенню вмісту навчальних курсів шляхом розробки методу формування списку навчального матеріалу з використанням генетичного алгоритму, зосередженого на комфортних умовах для людей, які планують розвиватись в нових сферах.

Актуальність. В час швидкого розвитку технологій, людям потрібно навчитись так само швидко адаптуватись до них, що зазвичай може проявлятись у потребі освоювати нові сфери знань.

Розвиток сфери онлайн-платформ для навчання є важливим у цьому завданні, оскільки це сприяє доступності освіти для широкого кола людей незалежно від їх місця проживання або графіку.

Крім того, розвиток таких систем сприяє інноваціям у методах навчання та стимулює викладачів та експертів до пошуку нових підходів до навчання та педагогіки. Такий підхід дозволяє впроваджувати нові технології та методи навчання, що може покращити якість освіти в загальному.

Окрім цього, розвиток онлайн-курсів також сприяє розширенню доступності освіти для людей з обмеженими можливостями, які можуть знайти у навчанні в інтернеті більш комфортну та доступну форму ніж традиційні навчальні заклади.

У контексті економіки розвиток даної сфери може стати стимулом для росту ринку дистанційної освіти та створення нових робочих місць у сфері освіти та технологій. Крім того, це може сприяти зменшенню витрат на навчання для студентів, оскільки вони можуть вибирати більш доступні та ефективні навчальні матеріали онлайн.

Об'єкт дослідження – процес покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми.

Предмет дослідження – методи та засоби кібернетичного аналізу для покращення вмісту навчальних курсів.

Мета кваліфікаційної роботи бакалавра – покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми.

Завдання кваліфікаційної роботи бакалавра.

Для досягнення мети визначено такі задачі дослідження:

- провести аналіз методів формування та покращення вмісту навчальних дисциплін;
- провести аналіз інтелектуальних методів для визначення найкращого вмісту навчальних курсів;
- реалізувати метод покращення вмісту навчальних курсів за генетичним алгоритмом
- виконати програмну реалізацію методу покращення вмісту навчальних курсів з адаптацією генетичного алгоритму;
- провести експериментальне тестування методу та його програмної реалізації;
- виконати оцінювання покращення вмісту навчальних курсів.

Структура та обсяг кваліфікаційної роботи бакалавра.

Кваліфікаційна робота бакалавра складається з завдання, змісту, переліку скорочень, вступу, 3 розділів, висновків, переліку посилань із 47 найменувань та 7 додатків. Загальний обсяг дипломної роботи бакалавра становить 102 сторінки, з них 61 сторінки основного тексту та 41 сторінок додатків. У роботі наведено 23 рисунки та 8 таблиць.

Ключові слова: генетичний алгоритм, освітня вебсистема, навчальний матеріал, покращення вмісту, ефективність навчання, персоналізація навчання.

Розділ 1 Характеристика предметної області та постановка задачі

1.1 Аналіз інформаційних моделей пов'язаних з освітніми вебсистемами

Сучасні методи комп'ютерних наук знайшли своє застосування у вирішенні широкого спектру завдань, що охоплюють як повсякденні питання, так і специфічні професійні виклики. Особливо значимим є їх внесок у розвиток інтернет-технологій, зокрема у створенні та оптимізації програмних вебзастосунків [1]. Вебсайти являють собою сукупність інформаційних ресурсів, доступних через інтернет, що підтримують різноманітність функцій – від освітніх і комунікаційних до комерційних [2].

Методи комп'ютерних наук дійсно демонструють значний потенціал у вирішенні різноманітних завдань. Значущий внесок цих методів відчувається насамперед у розвитку онлайн-технологій, зокрема у сфері вебдизайну [3], управління вмістом [4] і забезпечення інтерактивності вебсайтів [5].

Вебсайти, як інтегровані ІС, стали основою цифрового взаємодії, забезпечуючи фундамент для бізнесу, освіти, державного управління та соціальних мереж [6]. Кожна з цих сфер використовує унікальні компоненти та функції вебсайтів, що дозволяють користувачам не тільки отримувати інформацію, а й активно взаємодіяти, здійснюючи обмін даними, покупки, навчання та комунікацію в реальному часі [7]. На сьогоднішня, сайти відіграють неймовірно важливу роль у сучасному світі, надаючи різноманітні можливості для спілкування, навчання, розваг та комерційної діяльності [8].

У сфері освіти вебсайти надають доступ до онлайн-курсів, електронних бібліотек, тьюторіалів та інших освітніх ресурсів, значно розширюючи можливості для самоосвіти та дистанційного навчання [9]. Це стало особливо важливим у контексті глобальних викликів, таких як, наприклад, пандемія COVID-19, коли потреба в доступі до якісної освіти незалежно від фізичного розташування стала неймовірно важливою [10].

Освітні ресурси дозволяють навчальним закладам та викладачам

створювати онлайн-курси та матеріали для студентів у будь-якому місці світу. Інформаційні сайти надають доступ до новин, статей та інших джерел інформації для користувачів [11].

Через те що велика кількість інтернет-ресурсів призначена саме для комерційної діяльності та направлена на сферу розваг [12], освітня складова має проблеми з популярністю та доступністю [13]. Освітні платформи дозволяють отримати доступ до широкого спектру курсів, що охоплюють різні галузі знань – від академічних предметів до практичних навичок, але за певні суми, що зменшує їх доступність, при цьому не завжди гарантується якість та гнучкість навчання, під кожен окрему особистість [13].

Інколи, онлайн-ресурси надають можливість обрати власний графік, індивідуальний план навчання, або навіть підібрати ментора [14]. Для осіб із напруженим розкладом, які не можуть відвідувати традиційні навчальні заклади, це має велике значення. Крім того, багато з онлайн-платформ мають форуми або спільноти, де слухачі можуть обговорювати питання та спілкуватися з викладачами та іншими студентами [15].

Навчальні вебресурси можуть приймати різні форми, щоб надати користувачам різноманітні можливості та технології для навчання та підвищення своїх знань [16]. Сайти-підручники можуть бути дуже ефективними для візуального навчання, де студент може переглядати завжди свіжу інформацію, де викладач пояснює матеріал, демонструє приклади тощо [17]. Інтерактивні вправи та завдання допомагають студентам закріпити свої знання та отримати зворотний зв'язок щодо їх розуміння навчального матеріалу [18]. Електронні підручники можуть бути доступними у вигляді електронних книг або онлайн-документів і містити матеріал для самостійного вивчення [19]. Вебінари дозволяють студентам взаємодіяти з викладачем у реальному часі, задавати питання та отримувати відповіді [20].

Комерційні навчальні ресурси можуть пропонувати платний доступ до високоякісного навчального контенту, який може бути більш структурованим і професійно підготовленим, а також може включати додаткові сервіси або можливості, такі як сертифікація або підтримка, хоча як зазначалось вище,

мають проблеми з доступністю, через відносно високі ціни [21].

По при всі типи навчання і їх переваги, відеоуроки в сучасному світі навчання відіграють важливу роль і можуть бути вкрай перспективним типом навчання, особливо коли є вільний час для вивчення [22]. Порівняно з іншими типами навчання, відеоуроки мають кілька особливих переваг. По-перше, такі типи уроків можуть бути дуже ефективними для візуального навчання. По-друге, відеоматеріали можуть бути дуже зручними для самостійного навчання. По-третє, відеокурси можуть бути доволі доступними [23].

В незалежності від типу навчання, у навчальних вебресурсах беруть участь різні типи людей, які спільно працюють над створенням та розвитком освітньої платформи [24]. Викладачі та експерти з галузі відповідають за створення якісного та інформативного контенту [25]. Вони розробляють відеоуроки, тексти, вправи та завдання, спрямовані на ефективне навчання. Слухачі, у свою чергу, мають можливість вивчати цей матеріал, виконувати завдання та вправи, та спілкуватися з іншими учасниками навчального процесу. Адміністратори вебресурсів відповідають за технічну підтримку та адміністрування платформи [26]. Вони забезпечують безперебійну роботу сайту, вирішують технічні проблеми, а також ведуть роботу з рекламою та просуванням ресурсу [27]. Додатково, на проєкті можуть бути залучені інші спеціалісти, такі як дизайнери, аналітики, а також маркетологи, що працюють над просуванням та популяризацією вебресурсу серед цільової аудиторії [28].

У структурі онлайн-курсів відбуваються різні процеси, які спрямовані на створення та підтримку навчального середовища. Перш за все, проводиться аналіз потреб цільової аудиторії та розробляється концепція курсу. Лише після цього відбувається розробка навчального матеріалу. Далі проводиться тестування курсу. Після запуску відбувається процес реклами та просування, щоб залучити нових слухачів. [29].

Під час навчання слухачі також можуть отримувати зворотній зв'язок від викладачів чи адміністраторів курсу, що допомагає вдосконалити навчальний процес [30]. Після завершення курсу проводиться оцінка результатів та аналіз ефективності, який може використовуватися для покращення подальших курсів .

Використання навчальних вебресурсів має величезне значення у сучасному світі освіти. Ці ресурси забезпечують доступ до різної інформації, включаючи, навчальні матеріали, у будь-який час та з будь-якого місця, що робить освіту більш доступною та гнучкою. Вони дозволяють студентам вивчати нові теми та поглиблювати знання у власному темпі, що сприяє кращому засвоєнню матеріалу. Крім того, навчальні вебресурси стимулюють викладачів та експертів до розвитку нових методик навчання та сприяють інноваціям у галузі освіти. В цілому, використання навчальних вебресурсів є ключовим для покращення якості та доступності освіти в сучасному світі.

1.2 Огляд сучасних підходів щодо формування вмісту навчальних курсів

Освітні вебсистеми інтегровано пов'язані з комп'ютерними науками, що відкриває широкі можливості для розробки нових методів та алгоритмів, здатних позитивно вплинути на ефективність навчальних процесів. Зокрема, алгоритми машинного навчання та штучного інтелекту можуть значно оптимізувати системи розкладу занять, інтерактивне навчання, а також забезпечити адаптивність освітнього контенту до потреб студентів [31].

Аналітика даних значно впливає на процес освіти, дозволяючи виявляти тенденції та оцінювати ефективність навчальних курсів. В контексті використання штучного інтелекту в освітньому проекті, аналітика даних може бути збагачена застосуванням алгоритмів машинного навчання. За допомогою даних методів можливо не тільки аналізувати ефективність існуючих курсів, але й розробляти рекомендації для адаптації курсів з метою забезпечення більшої персоналізації та відповідності освітнього контенту потребам студентів [32].

Штучні нейронні мережі, які є основою сучасних алгоритмів машинного навчання, здатні значно трансформувати процес створення навчального матеріалу в освіті. Одним з найпрогресивніших застосувань ШНМ у створенні навчальних матеріалів є їхня здатність аналізувати існуючий освітній контент та структурувати його відповідно до потреб різних груп слухачів [33]. Це означає,

що ШНМ можуть виробляти освітній контент, який варіюється від простих визначень і пояснень до складних наукових статей, що можуть використовуватися як додаткові навчальні ресурси [34]. Ці системи здатні створювати текст, який є зрозумілим і адаптованим до рівня знань користувача.

Мережі глибокого навчання є одним із найпотужніших інструментів ШНМ, які застосовуються у багатьох галузях, зокрема в освіті. Вони засновані на архітектурі штучних нейронних мереж, які імітують процеси обробки інформації в людському мозку [35]. Глибоке навчання може використовуватися для розробки систем, здатних оцінювати відповіді вільної форми. Моделі глибокого навчання, які використовують такі структури як згорткові нейронні мережі або рекурентні нейронні мережі, можуть аналізувати текст і вирішувати, чи відповідає студентська робота критеріям [36].

Глибоке навчання може використовуватися також і для створення інтерактивних навчальних ігор та симуляцій, що вимагають від студентів розгадування складних проблем або виконання завдань у віртуальному середовищі [37].

Використання методів кластеризації [38] у ІС, що практикують групові заняття, є значущим етапом у процесі підбору навчальних курсів. Застосування алгоритмів кластеризації дозволяє групувати студентів згідно з їхніми атрибутами, такими як рівень знань або інтереси, що підвищує ефективність навчання за рахунок адаптованості.

Додатково, алгоритми машинного навчання, такі як класифікація, можуть бути використані для індивідуальної роботи зі студентами. Застосування класифікаційних моделей дозволяє аналізувати історичні дані та поведінкові патерни студентів, що сприяє прогнозуванню їхньої потенційної успішності в майбутніх курсах [39].

Інтелектуальні системи також можуть бути інтегровані з різними компонентами навчальної екосистеми, такими як електронні бібліотеки та віртуальні навчальні середовища [40], створюючи комплексне рішення, що підтримує всі аспекти освітнього процесу.

Також, актуальні напрямки використання алгоритмів машинного

навчання для підбору навчальних курсів включають в себе розробку інтелектуальних систем рекомендацій, які використовуються для оптимізації процесу вибору навчальних матеріалів [41]. Ці системи спроможні проводити глибокий аналіз індивідуальних освітніх потреб та досвіду студентів.

Підбір курсів, що ідеально відповідають потребам та інтересам слухачів, є важливими для забезпечення ефективного навчального процесу. У цьому контексті, еволюційні алгоритми, особливо генетичні алгоритми, виявляються надзвичайно корисними [42]. Вони використовують принципи еволюції, адаптовані з природи, для створення оптимальних рішень у комплексних і динамічних умовах, які характеризують сучасне освітнє середовище.

ГА імітує процеси природного відбору та генетичного розмаїття, використовуючи мутації та схрещування для відтворення нових поколінь рішень, які з часом стають все більш пристосованими до заданих умов [43]. У контексті освіти це означає, що алгоритм може аналізувати величезну кількість можливих навчальних курсів та їх параметрів, включаючи зміст, складність та стиль навчання, для визначення тих, які найкраще відповідають індивідуальним потребам кожного студента.

Цей підхід дозволяє не тільки підвищити швидкість навчання, але й значно покращити його якість. Студенти отримують доступ до контенту, який максимально відповідає їхнім уподобанням та навчальним потребам, що сприяє підвищенню мотивації та залученості в процесі навчання.

Більше того, постійний аналіз результатів навчання та зворотній зв'язок від студентів дозволяють алгоритму вдосконалюватися, що робить систему підбору курсів все більш точною і ефективною. Наприклад, шляхом аналізу успішності студентів у певних дисциплінах, алгоритм може ідентифікувати та рекомендувати курси, які підтримають їхні сильні сторони або допоможуть подолати виявлені труднощі.

Таким чином, інтеграція генетичних алгоритмів у процес формування навчальних курсів відкриває нові перспективи для персоналізації та оптимізації освітніх траєкторій, роблячи навчання не тільки більш ефективним, але й більш відповідним до індивідуальних особливостей і потреб кожного студента.

Дана КРБ зосереджена саме на розробці методу підбору курсів для користувачів, на основі їх параметрів. Для реалізації цього процесу буде використаний ГА, який може використовуватись для вирішення подібних задач, які не можна вирішити за допомогою традиційних підходів до розробки.

1.3 Аналіз існуючих освітніх вебсистем та наукових рішень

Перед розробкою нового програмного продукту важливо провести аналіз існуючого програмного забезпечення в предметній області. Це дозволяє краще зрозуміти потреби користувачів, ідентифікувати сильні та слабкі сторони існуючих рішень та врахувати їх при проектуванні нового продукту. Також, аналіз існуючого програмного забезпечення допоможе виявити тенденції та перспективи розвитку в галузі покращення вмісту навчальних курсів.

У світі Інтернету існує безліч освітніх вебплатформ, які можуть бути використані для навчання та отримання нових знань. Деякі з них пропонують інтерактивні курси, інші спеціалізуються на відео-лекціях, а є й такі, що зосереджуються на вирішенні практичних завдань та проектів.

Один з відомих прикладів навчальних вебресурсів – Skillshare [44]. Ця платформа є однією з найбільш популярних на ринку, де навчається понад 5 мільйонів студентів за допомогою до 35 000 уроків у різних сферах творчості. Уроки тривають від 20 до 60 хвилин і включають серію коротких відео, цікаві проекти і спільноту студентів, яка сприяє взаємодії через форуми для обговорення. Сторінка вибору курсів зображена на рисунку 1.1.

Як і всі інші, дана платформа має свої переваги та недоліки. Вона пропонує легке та коротке навчання, а також надає можливість обговорення на форумі підтримки, де студенти можуть обмінюватися відгуками. Також вона має зручну систему тегів для вибору курсів по власному бажанню, а також можливість розміщувати власний навчальний матеріал. Крім того, за одну ціну можна пройти багато курсів. Однак у платформи є й недоліки, наприклад, зміст курсу обмежений лише чотирма категоріями. Також відсутнє свідоцтво про закінчення курсів або інші офіційні визнання.



Рисунок 1.1 – Зображення сторінки вибору курсів на Skillshare [44]

Udacity [45] – це освітня платформа, яка навчає слухачів необхідним навичкам і знанням для успішної кар'єри в індустрії технологій. Навчальні програми платформи, відомі як "nanodegrees", розроблені у співпраці з великими технологічними компаніями, такими як Google і Microsoft. Місія Udacity – трансформація талантів у масштабі, що означає прагнення ефективно доносити інформацію до своїх користувачів. Зображення списку запропонованих найпопулярніших курсів зображено на рисунку 1.2.

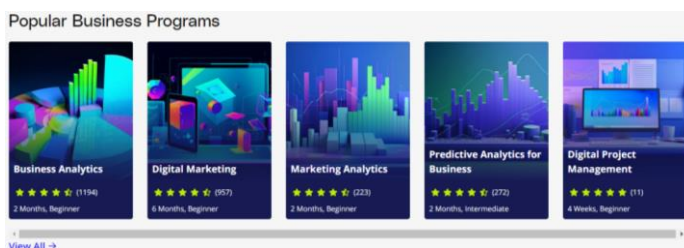


Рисунок 1.2 – Список найпопулярніших запропонованих курсів [45]

Інтерфейс платформи – простий і легко зрозумілий. Крім того, вона пропонує високу якість змісту курсів і додаткові кар'єрні послуги. На платформі доступно понад 200 безкоштовних курсів, і можна отримати нагородні сертифікати, які, як і попередньому випадку, не акредитовані. Також платформа підтримує навчання в реальному часі. Проте варто відзначити й недоліки. Наприклад, різноманіття тем курсів обмежене. Крім того, матеріали курсів головним чином представлені англійською мовою. Мобільний застосунок платформи недоступний, і варто зазначити, що це досить дорогий варіант порівняно з іншими платформами.

LinkedIn Learning [46], в свою чергу – це освітня платформа, яка пропонує професійні курси з бізнесу, цифрового маркетингу, веброзробки та інших творчих галузей, таких як дизайн, у форматі відеоуроків. LinkedIn Learning також співпрацює з провідними компаніями для надання своїм учням найбільш затребуваних професійних сертифікатів. Список представлених рекомендованих курсів зображено на рисунку 1.3.

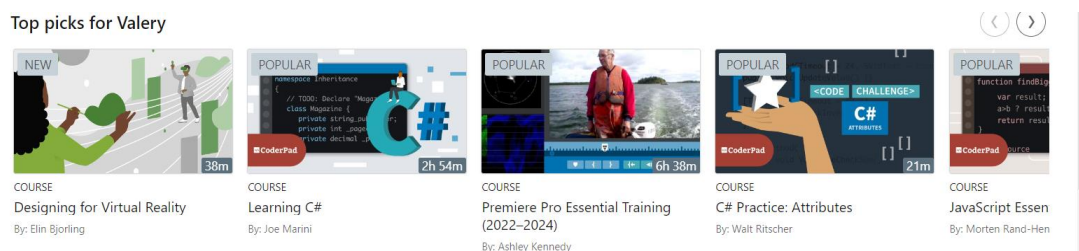


Рисунок 1.3 – Зображення рекомендованих курсів [46]

Даний вебсистема відома та цінується у бізнес-спільноті. Крім того, вона пропонує місячну безкоштовну пробну версію і надає персоналізовані рекомендації курсів для користувачів. Для кожного користувача підбирається список курсів, які можуть його зацікавити. Ви також можете оцінити свій прогрес за допомогою тестів і навчатися офлайн. Проте варто відзначити, що якість курсів є неоднозначною, тому вам, можливо, знадобиться провести деякі дослідження перед записом на них. Крім того, приєднатися в якості інструктора може бути складно.

Також, було проведено пошук різних наукових робіт, пов'язаний з даною тематикою. Найближчою до теми КБР роботою виявилась під назвою “Helping university students to choose elective courses by using a Hybrid Multi-criteria Recommendation System with Genetic Optimization”, написана авторами A. Esteban, A. Zafra та C. Romero [47].

У даній роботі йдеться про те, що університетські програми зазвичай включають в себе різноманітні вибіркові курси. Ці курси обираються студентами серед багатьох варіантів і є обов'язковими для завершення навчання та отримання університетського ступеня, але студенти не завжди роблять доцільний вибір. Для вирішення цієї проблеми було використано системи

рекомендацій, які допомагають студентам приймати рішення, враховуючи їхні специфічні вподобання. У цій роботі було розроблено гібридну рекомендаційну модель, яка поєднує колаборативний фільтр та контент-базовий фільтр з використанням кількох критеріїв, пов'язаних як з інформацією про студента, так і з інформацією про курси. Для автоматичного визначення оптимальної конфігурації був розроблений генетичний алгоритм.

Експериментальне дослідження включало реальну інформацію про студентів та курси зі спеціальності “Computer Science Degree” університету Кордови. Результати показали, що гібридна модель, яка поєднує інформацію про студента та інформацію про курси, забезпечує надійні рекомендації та виявляє важливі критерії для рекомендацій курсів.

Іншими словами, дана наукова робота на практиці підтверджує, що ГА можна ефективно використовувати при роботі з навчальним матеріалом, для його підбору під кожну особистість.

Отже, існує багато варіантів освітніх вебплатформ, і кожна із них має свої сильні та слабкі сторони. На більшості вебресурсів, проблемою є велика сума оплати за їх послуги. Окрім цього, існує проблема з підбором курсів під потреби користувачів, через що навчання може стати не структурованим, і відповідно що неефективним. Це означає, що щоб покращити якість навчання, освітні вебресурси, варто робити безкоштовним і, як було визначено, з можливістю підбору курсів на основі параметрів, які можуть цікавити користувача. Також, великою перевагою буде зосередження на швидкості проходження курсів.

1.4 Мета та завдання кваліфікаційної роботи

Аналіз предметної області підтверджує актуальність та необхідність розробки методу покращення вмісту навчальних курсів за генетичним алгоритмом та його програмну реалізацію як освітню вебсистему.

Метою кваліфікаційної роботи бакалавра є покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми.

Для досягнення мети визначено такі задачі дослідження:

- провести аналіз методів формування та покращення вмісту навчальних дисциплін,
- провести аналіз інтелектуальних методів для визначення найкращого вмісту навчальних курсів,
- реалізувати метод покращення вмісту навчальних курсів за генетичним алгоритмом,
- виконати програмну реалізацію методу покращення вмісту навчальних курсів за генетичним алгоритмом як освітньої вебсистеми,
- провести експериментальне методу та його програмної реалізації,
- виконати оцінювання покращення вмісту навчальних курсів.

Розділ 2 Метод покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми

2.1 Загальна структура методу

Загальна структура методу покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми зображена на рисунку 2.1.

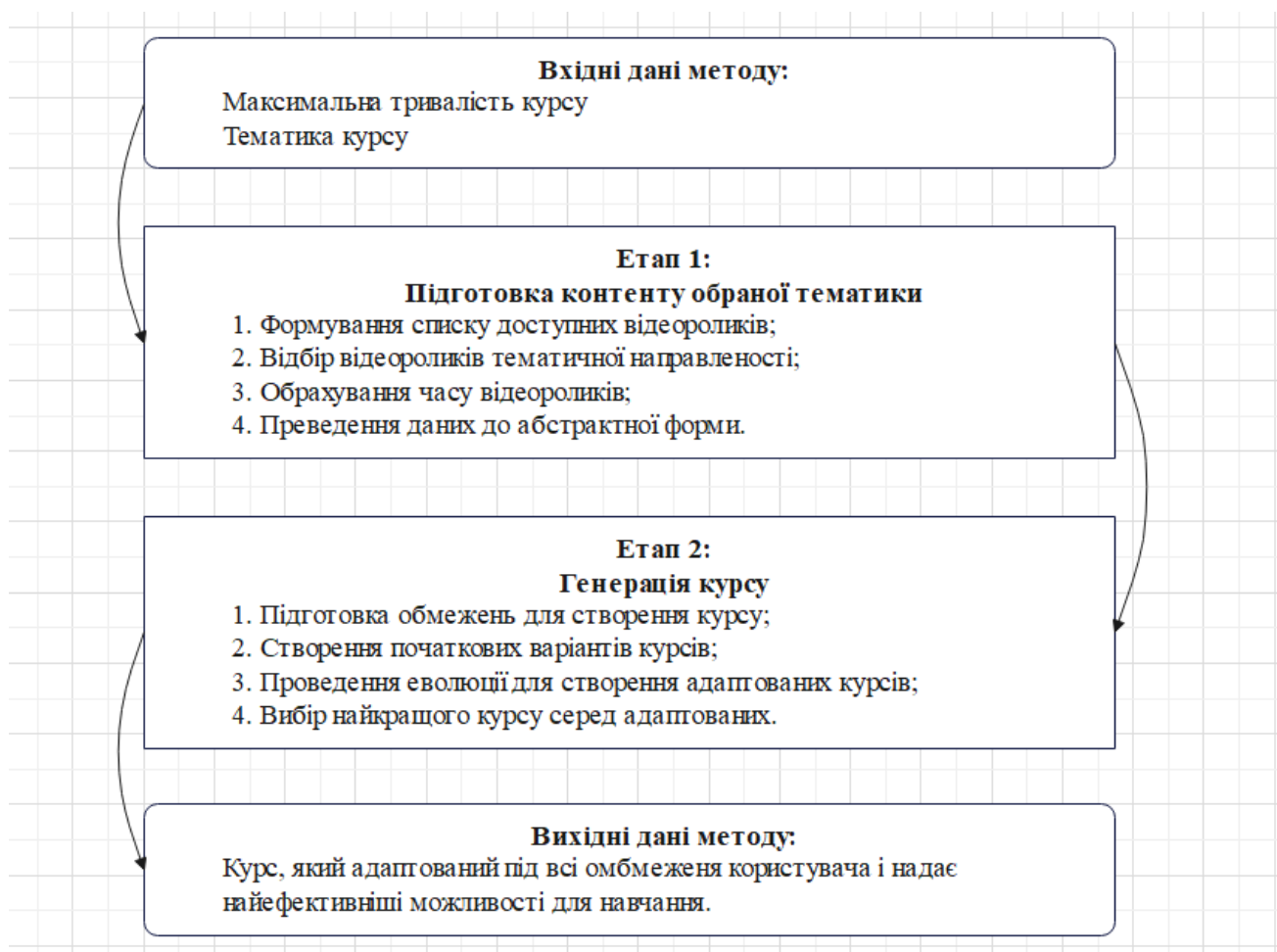


Рисунок 2.1 – Схема методу генерації курсів

Розроблений метод полягає в формуванні переліку навчальних курсів на основі параметрів наданих користувачем. Як було зазначено у першому розділі, для слухачів важливим є час, який вони можуть виділити на своє навчання і тематика, в якій вони хотіли б отримати знання. Саме тому, ці параметри є вхідними.

Перед формування можливих курсів, потрібно провести відбір відеороликів. Даний підхід формування дозволить звільнити генетичний алгоритм від додаткових дій на перевірку відповідності ролика тематиці. В результаті цього процесу отримується список відеороликів, які можуть бути використані для формування курсів.

Формування варіацій курсів уже передбачає використання генетичного алгоритму. Під час цього етапу, будуть формуватись різні комбінації переліків навчальних курсів, які допустимі для надання користувачеві.

Курси, через обмеження в часі, не завжди будуть мати всі можливі відеоролики, але все одно мають дотримуватись логічної послідовності матеріалу. В результаті, користувачу надаватиметься список курсів, які уже можуть претендувати на можливість бути унікальними і відповідають поставленим обмеженням.

На останньому етапі, має обиратись лише один курс з найкращих. Тому, з кращих можливих комбінацій, буде обиратись курс з сумарною тривалістю відеороликів найбільш наближеною до максимально визначеної слухачем.

Отже, запропонований метод покращення вмісту навчальних курсів базується на формуванні саме унікальних курсів, які генеруватимуться під потреби користувачів. Метод складається з кількох етапів: формування доступних відеороликів, формування списку курсів, які найкраще підходять під поставлені умови, відбір відеоматеріалу серед кращих варіацій для перегляду для швидкого навчання.

2.2 Імплементация генетичного алгоритму в освітню вебсистему

2.2.1 Загальна структура генетичного алгоритму

Генетичний алгоритм складається з кількох етапів, які можуть бути реалізовані з загальною структурою зображеною на рисунку 2.2.

Процес пошуку рішення в генетичному алгоритмі розпочинається з ініціалізації нульової популяції. Популяції, у випадку, генерації курсів – це списки можливих курсів, які складаються з наданих відео.

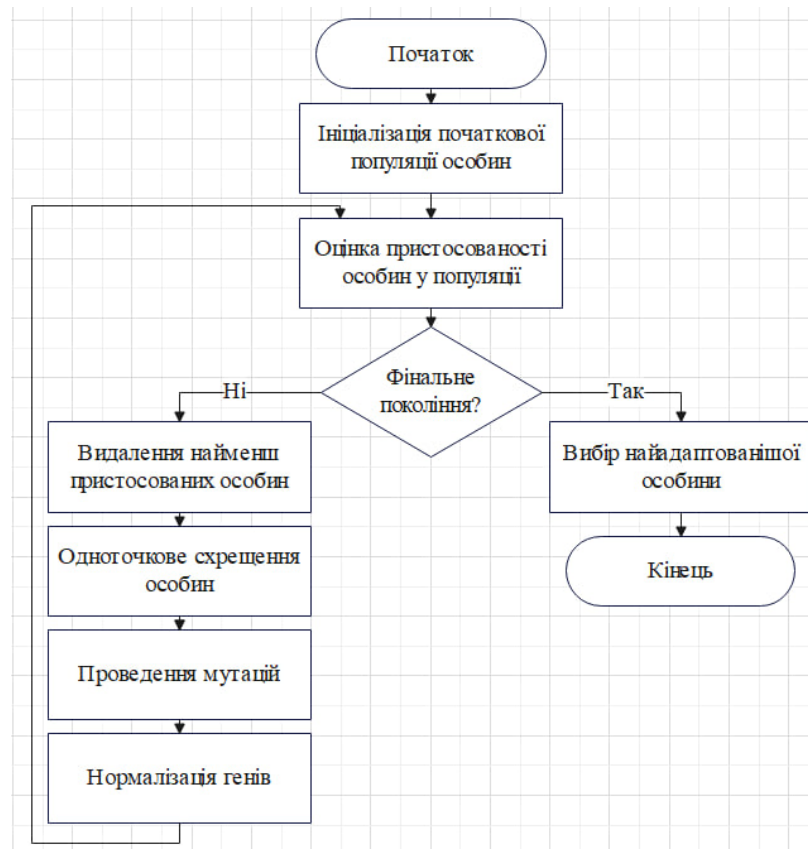


Рисунок 2.2 – Загальна структура генетичного алгоритму

Популяції складаються з особин, які в свою чергу – це кожний конкретний курс, який бере участь у формуванні нових поколінь, або виступають в ролі рішень.

Популяція потенційних рішень, може мати різноманіття особин, які відрізняються структурою. Ці рішення кодуються за допомогою генів. В контексті розробленої методики, гени – це відеоролик, який визначається, як тривалість з ідентифікатором. Список генів кожної особини – це геном.

Відеоролики відносяться до певних компетенцій, які визначаються, як тематики, до яких вони відносяться. Один відеоролик може відноситись до кількох компетенцій.

Мутація представляється, як випадкове створення нового курсу, шляхом заміни випадкового відеоролику на інший, або додаванням нового, у середині уже існуючого курсу.

Еволюційний процес, відбувається до того моменту, доки не будуть

виконані певні критерії досягнення цільової функції.

Отже, терміни генетичного алгоритму можуть співвідноситись до термінів освітніх платформ. Це дозволяє зрозуміти, що даний алгоритм може бути імплементований, як система для генерації курсів.

2.2.2 Особливості імплементації генетичного алгоритму для реалізації методу генерації курсів

Відповідно до визначених термінів, структуру генетичного алгоритму можна адаптувати наступним чином, відповідно до умов поставлених задач (рисунок 2.3).

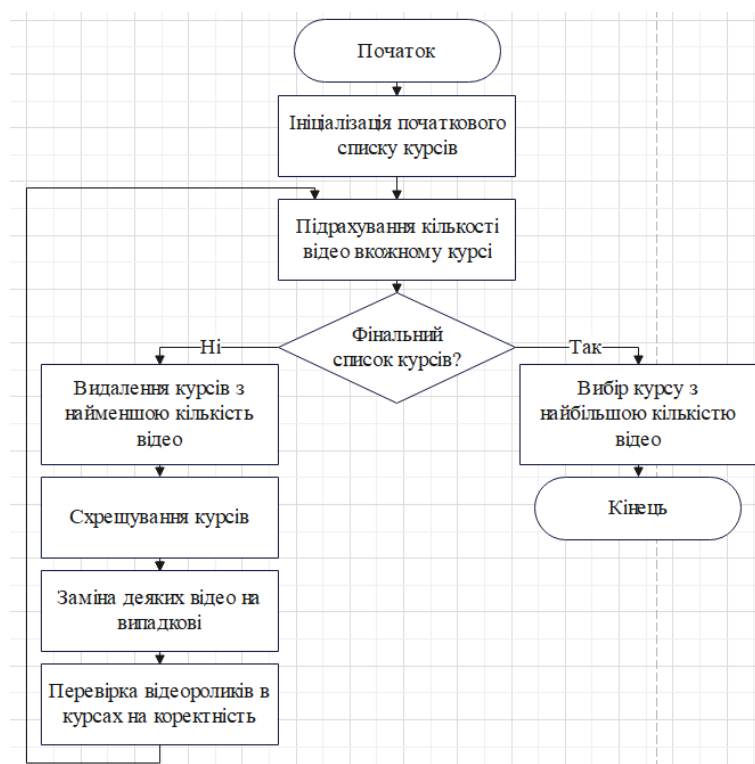


Рисунок 2.3 – Адаптований генетичний алгоритм під умови задач

Відштовхуючись від дослідження, розглянутого в попередньому розділі, а також інших подібних додатків, запропоновано метод відбору навчального відеоматеріалу для унікального переліку навчальних курсів, для формування індивідуального курсу користувача, на основі його вподобань, який зображено

нижче, при цьому етапи пов'язані з генетичним алгоритмом, виділені рожевим кольором (рисунок 2.4).

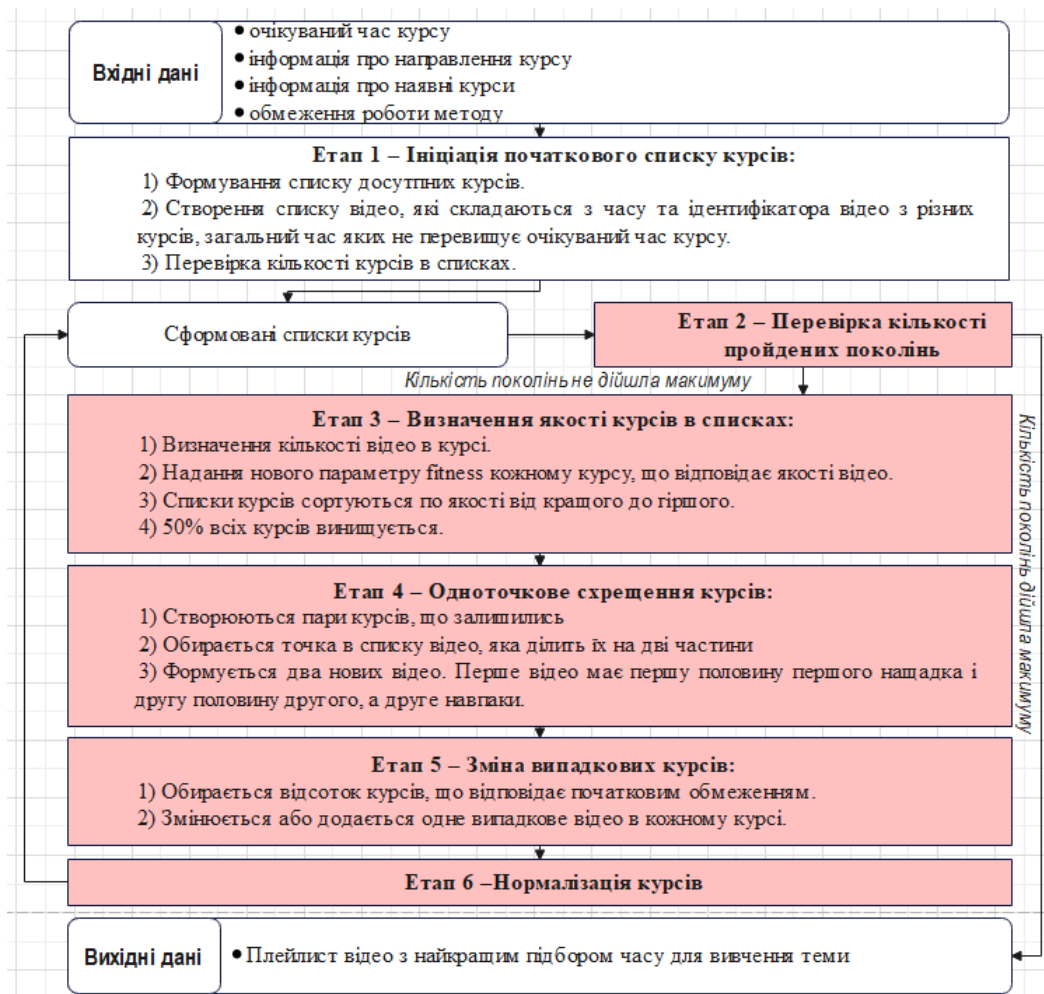


Рисунок 2.4 – Структура методу створення персоналізованих курсів

На початковому етапі реалізації генетичного алгоритму формуються початкові варіанти курсів. Вже на цій стадії необхідно виключити потребу постійної перевірки відповідності компетентності відеоматеріалів, інтегруючи вхідні дані, які задають тематичний критерій не лише при формуванні, але і під час інших операцій.

У разі якщо поточний список курсів не є завершальним, алгоритм переходить до наступного етапу, який полягає у визначенні якості кожного курсу. Цільова функція алгоритму в цьому випадку базується на максимальній кількості відеоматеріалів у кожному курсі довжина яких не перевищує заданий часовий ліміт.

Після аналізу, курси ранжуються за кількістю і здійснюється відсів менших за розміром курсів, що забезпечує простір для репродукції більш вдалих курсів. Передбачається, що кількість варіантів завжди буде сталою.

На наступному етапі відбувається створення нових курсів, на основі відібраних. Курси паруються по два елемента, після чого відбувається схрещення, під час якого в батьківських курсів обирається позиція, що ділить відеоролики на дві частини і комбінує їх.

Наступним етапом є забезпечення нових курсів, шляхом заміни в них відеороликів. При цьому процесі береться випадкова позиція у списку відеороликів і замінюється на випадковий відеоматеріал. Відповідно, зберігається різноманіття відеороликів.

Після проведення описаних операцій, потрібно провести нормалізацію курсів (рисунок 2.5), тому що потенційно може відбуватись комбінування схожих курсів, що може призвести до створення списків відео, які містять повторення. Така ж ситуація може статись і на етапі заміни відео, через що курси, під час наступних схрещувань можуть ставати гіршими.

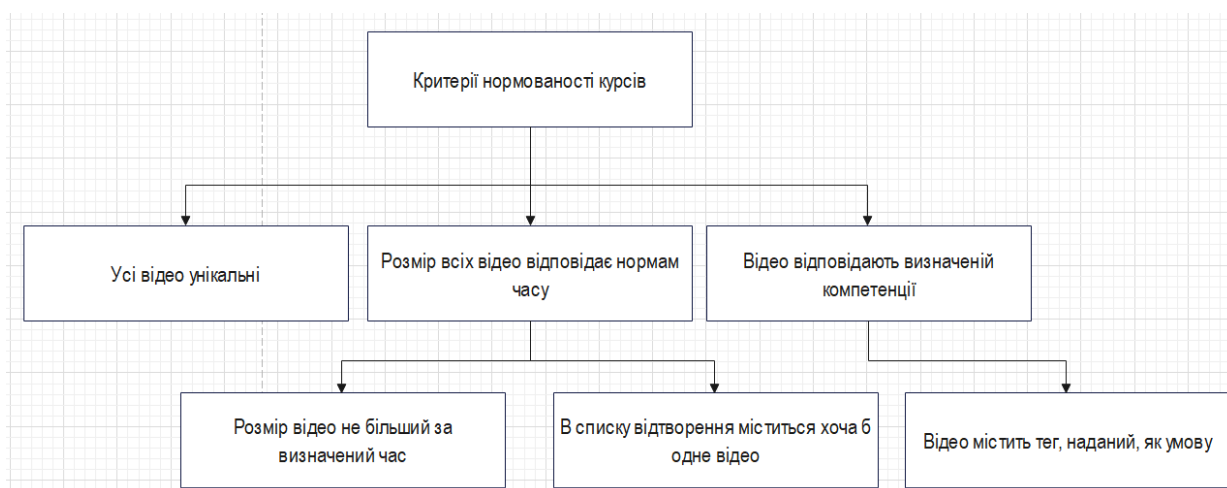


Рисунок 2.5 – Критерії нормальності відеороликів у курсах

Таким чином на шостому кроці потрібно модифікувати відео курсів одразу по двом критеріям, які можуть відбуватись як роздільно, так по черзі. Шостий етап зображено на наступному рисунку (рисунок 2.6).

Знищення курсів, які отримали некоректні списки відео може призвести

до зменшення кількості варіантів курсів, тому має проводитись модифікування, щоб вони могли підпадати під задані критерії. Тому, якщо курс має проблеми з дубльованими відео, тоді залишаються лише одиничні їх екземпляри.

Якщо курс навіть після зміщення відео не адаптований під поставлені критерії, або не коректний по при початкове різноманіття відео, випадковим чином починають прибиратись відео з переліку навчальних курсів, до того часу, доки загальна тривалість не стане задовільною.

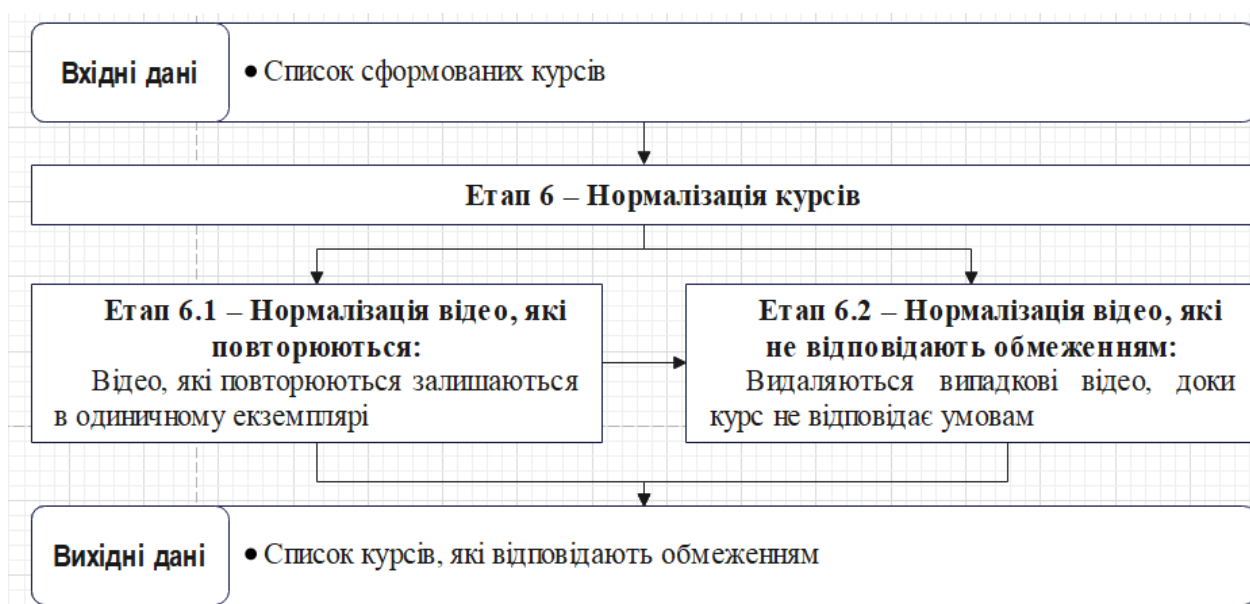


Рисунок 2.6 – Процес нормалізації курсів

Далі, потрібно перевірити, чи номер сформованого списку варіантів курсів допускає створення ще одного. Якщо після проходження кількох циклів номер стає рівний визначеному в обмеженні, то відбувається вихід з циклу і найкраща курс вважається результатом роботи алгоритму.

Отже, можна зробити обґрунтовані висновки, що ГА ефективно адаптується для формування навчального контенту відповідно до варіативних критеріїв. Описаний метод демонструє здатність генетичних алгоритмів до вирішення складних оптимізаційних завдань у сфері освітніх технологій, підкреслюючи їхню універсальність та адаптивність у формуванні персоналізованих освітніх траєкторій.

2.3 Особливості використання спеціалізованих програмних компонентів

У програмній реалізації методу, має бути реалізована взаємодія з інформацією про тривалість відео, які надходять до генетичного алгоритму. Для того щоб зменшити кількість інформації, що знаходиться в базі даних, пропонується знаходити дані про відео динамічно, при запитах на формування бажаних курсів різними користувачами.

Найкращим вибором є серверна утиліта FFmpeg, яка надається, яка відкрите програмне забезпечення, призначене для запису, конвертації та потокової обробки аудіо- та відеофайлів. Воно має великий набір функцій і може використовуватися для різноманітних завдань, від простого конвертування форматів до складних операцій обробки відео та аудіо.

FFmpeg може працювати з різними форматами відео та аудіо, включаючи популярні формати, такі як MP4, AVI, MKV, MP3, AAC та іншими. Він підтримує різні кодеки для кодування та декодування відео та аудіо, що робить його дуже гнучким для використання в різних сценаріях.

Однією з ключових можливостей FFmpeg є можливість здійснювати потокову обробку відео та аудіо, що дозволяє виконувати обробку без зберігання проміжних файлів. Це корисно для швидкого отримання інформації про тривалість медіаматеріалів з метаданих. Для проекту важливо те, що FFmpeg може бути використаний в командному рядку. Це робить його доступним для зручного використання з серверною частиною додатку.

2.4 Проектування інформаційної структури навчального вебресурсу з генерацією курсів на основі генетичного алгоритму

В сучасному світі, більшість вебсистем використовують централізовані сховища зберігання інформації з використанням СКБД, які дозволяють зручно оперувати даними для потреб користувача. Відповідно до цього, потрібно розробити базу даних, яка відповідатиме потребам навчального вебресурсу, який

забезпечить нормальне функціонування всіх розроблених компонентів системи. Для програмної реалізація методу, яка розробляється в даній КБД запропоновано варіант сутностей, які будуть використані при розробці (додаток А.1).

Сутність “Слухач” є важливим компонентом системи, оскільки вона забезпечує зручну реєстрацію та авторизацію користувачів. Кожен з її атрибутів виконує конкретну роль у забезпеченні безпеки та зручності використання платформи. Ім'я користувача та його електронна пошта використовуються для персоналізації та зв'язку з користувачем, а поле “Пароль” забезпечує захист доступу до облікового запису. Крім того, наявність поля “Фото” дозволяє користувачам завантажувати своє фото для профілю, що робить взаємодію з платформою більш особистою та приємною.

Відповідно, значення “Викладачі” використовується, як і у випадку з слухачів, для зберігання інформації про викладачів та забезпечення їхньої можливості реєстрації та входу до системи. Кожен з атрибутів таблиці має своє значення для забезпечення зручності та безпеки використання платформи. Ім'я викладача використовується для персоналізації та звернення до викладача, а поле “Професія” містить інформацію про професію чи предмет, які викладає цей викладач. Електронна адреса викладача та його пароль потрібні для авторизації. Поле “Фото” містить фотографію викладача, що може бути використано для візуального впізнавання або персоналізації профілю.

Елемент діаграми “Курси” дозволяє викладачам створювати та редагувати списки відтворення курсів, а користувачам – відтворювати та переглядати їх. Кожен з атрибутів цієї таблиці має важливе значення ефективності використання платформи. Поле “Назва” містить заголовок списку відтворення, що дозволяє користувачам швидко зрозуміти його зміст. Опис списку відтворення знаходиться у полі “Опис”, що може містити додаткову інформацію про список. Поле “Обкладинка” містить мініатюру списку відтворення, що може бути використано для швидкого візуального впізнавання. Дата створення списку відтворення зберігається у полі “Дата викладення”, що дозволяє користувачам відслідковувати, коли був створений список. Статус списку відтворення позначається, як “Статус активності”, що може вказувати на

його поточний стан або доступність для перегляду.

Сутність “Контент” використовується для зберігання інформації про матеріали курсу, що дозволяє викладачам створювати та редагувати їх, а користувачам - переглядати. Ця таблиця має ключове значення для забезпечення доступу до відео та інших матеріалів курсу. Дана таблиця нерозривно пов’язана з курсом зв’язком, при якому один курс може містити багато відеороликів. Опис матеріалу курсу знаходиться у полі “Опис”, що може містити додаткову інформацію про матеріал. Відео матеріалу курсу зберігається у властивості “Відео”, що дозволяє користувачам переглядати відеоконтент. Мініатюра матеріалу курсу міститься у полі “Обкладинка”, що може бути використано для візуального впізнання. Дата створення матеріалу курсу зберігається, як “Дата створення”, що дозволяє відслідковувати, коли був створений матеріал. Статус матеріалу курсу знаходиться у полі “Статус активності”, що може вказувати на його поточну доступність для перегляду.

Для того, щоб класифікувати перелік навчальних відео, потрібно використовувати сутність “Теги”. Один перелік навчальних курсів з відео може відноситись до багатьох відео, що дозволить використовувати один і той же матеріал для багатьох згенерованих курсів з освітньою інформацією.

Сутність “Контакти” призначена для забезпечення можливості користувачам зв’язатися із платформою, надсилаючи своє повідомлення. Кожен з атрибутів цієї таблиці відіграє роль у забезпеченні спілкування з адміністрацією додатку. Електронна адреса відправника та номер телефону забезпечують засіб зв’язку з відправником повідомлення. Поле “Повідомлення” містить саме повідомлення від відправника, що дозволяє передати необхідну інформацію чи запит коректно та точно.

На основі описаних сутностей, їх властивостей, а також зв’язків між ними, можливо сформуванати даталогічну модель бази даних (додаток А.2), яка буде використана у навчальному вебзастосунку. Також, додані поля, які однозначно ідентифікують кожну сутність, а назви перекладені на англійську мову, для кращої сумісності з різними СКБД. Для формування зв’язків між даними, для об’єднаних даних створені додаткові поля, що відповідають назвам

зовнішніх ключів. Для забезпечення нормалізації бази даних, та уникнення зв'язку “багато-до-багатьох”, було додано таблицю під назвою “Playlist_tag”, що відповідає призначенню класифікаційних тегів для різних курсів.

Нормалізація бази даних – це процес організації структури даних у базі даних для зменшення повторення даних та забезпечення їхньої цілісності. Головна мета нормалізації – розбиття таблиць на менші, більш компактні та зв'язані за допомогою ключів, що дозволяє ефективніше управляти даними та зменшує ймовірність помилок при їхньому збереженні та оновленні.

Як можна побачити у додатку А.2, дані відповідають нормам нормалізації. Для цього було розбито сутності на таблиці, які мають зв'язок один до багатьох, що означає приведення БД до третьої нормальної форми, що забезпечує надійний доступ до управління даними, без можливих аномалій.

Використання бази даних у сучасних проектах є важливою складовою, адже дозволяє отримати сховище інформації, яке може бути використано у будь-який момент часу. Спроектвана БД для програмної реалізації методу враховує сутності, які потрібні для успішної роботи алгоритму. Загальну систему зв'язків було приведено до третьої нормальної форми, що дозволяє позбутись аномалій в процесі обробки інформації, які можуть виникнути під час реалізації вебсистеми.

2.5 Підготовка робочих вхідних даних для системи формування курсів

Для підготовки до реалізації методу, потрібно визначити, як будуть додаватись дані. Це необхідно для коректного тестування визначених методів, а також дієздатності проекту, при використанні користувачами.

Викладачі в системі мають можливість створювати і редагувати навчальні курси та додавати до них відео вручну. Тому для перевірки працездатності ресурсу, потрібно ввести деяку частину даних вручну, що дозволить наглядно спостерігати поведінку системи, при використанні різних функцій.

Для покращення вмісту навчальних курсів метод використовує генетичний алгоритм. Підготовка робочих вхідних даних для цього процесу є

ключовим етапом у розробці методу, оскільки від правильності та повноти даних залежить ефективність алгоритму. Користувачі системи, будучи викладачами або слухачами, мають можливість самостійно вносити необхідні обмеження. Ці обмеження складаються з двох основних параметрів: обмеження тривалості на перегляд бажаного переліку навчальних відео та тегу, який відповідає курсам чи відео, які бажає переглянути користувач.

Параметри, які відповідають за процес еволюції в генетичному алгоритмі, мають бути внесені у код самої системи, та протестовані для визначення оптимальності його роботи. До цих параметрів відносяться наступні пункти: шанс мутації, розмір списку варіантів курсів та кількість поколінь.

Враховуючи усі потреби до вхідних даних, як висновок можна зробити те, що частина даних, які зможуть вносити користувачі, потрібно вписати до бази даних вручну. Інформація, яка стосується генерації відеокурсів, має бути введена в реалізовану систему невід'ємно від коду. Такий підхід до використання інформації в додатку, дозволить надалі провести тестування та визначити оптимальність спроектованого навчального вебресурсу.

2.6 Проектування функціональної структури освітньої вебсистеми з автоматичною генерацією курсів

Для того, щоб розробити програмну реалізацію методу, потрібно визначити функціональні особливості усього додатку, включаючи стани та процеси, що будуть відбуватись.

Управління функціоналом та доступом, у програмній реалізації методу, відіграє важливу роль у забезпеченні зручності та ефективності користування. Головним аспектом цього процесу є концепція ролей користувачів, що дозволяє визначити, які можливості та опції доступні для різних категорій користувачів.

На початковій стадії розробки важливо чітко визначити ролі користувачів, кожна з яких має унікальний набір функціональних можливостей. Виділяються три основні ролі: гість, слухач та викладач. Функціонал, притаманний кожній з цих ролей, стратифікований та специфічний, що

відображає різноманіття доступу та взаємодій в середині платформи.

Функціональні можливості гостя обмежені в основному переглядом вмісту додатку. Ця роль дозволяє взаємодіяти тільки з базовими процедурами, такими як реєстрація та авторизація, які доступні без обмежень. Якщо користувач, який знаходиться у ролі гостя, намагається доступити функції, зарезервовані для інших ролей, система спонукає його до проходження процесу аутентифікації чи реєстрації.

Авторизація та реєстрація виступають як позаролеві функції, оскільки вони впливають на переконфігурацію доступу користувачів до різних інструментів і ресурсів платформи. Однією з ключових особливостей системи є здатність слухача перейти у роль викладача, що надає йому можливість ділитися власними навчальними матеріалами з іншими користувачами. Це розширення функціональності підтримує гнучкість програмної реалізації методу і сприяє більш активній участі користувачів у освітньому процесі.

Візуалізація концепції зміни ролей користувачів може бути представлена на рисунку 2.7, що детально висвітлює механізми переходу між ролями та інтерактивність відповідного функціоналу.

Також, варто зауважити, що враховуючи специфіку вебзастосунків, доступ до функціоналу залежить не лише від ролі користувача, а й від сторінок, на яких знаходиться користувач. Відповідно, ці елементи можна назвати функціональними підблоками, які розділяються інструменти по логічних категоріях для більш зручного доступу до них.

На основі ролей, можна зробити висновки щодо даних, якими будуть оперувати користувачі різних ролей. Наприклад, користувач може надавати особисті дані, для входження в систему, інформацію для генерації персонального курсу, дані для запитів до різних сторінок, в результаті чого буде отримувати відповідь в залежності від типу запиту. Інформація яка проходить через систему може мати різні формати, як текстові, так і медіа-матеріали.

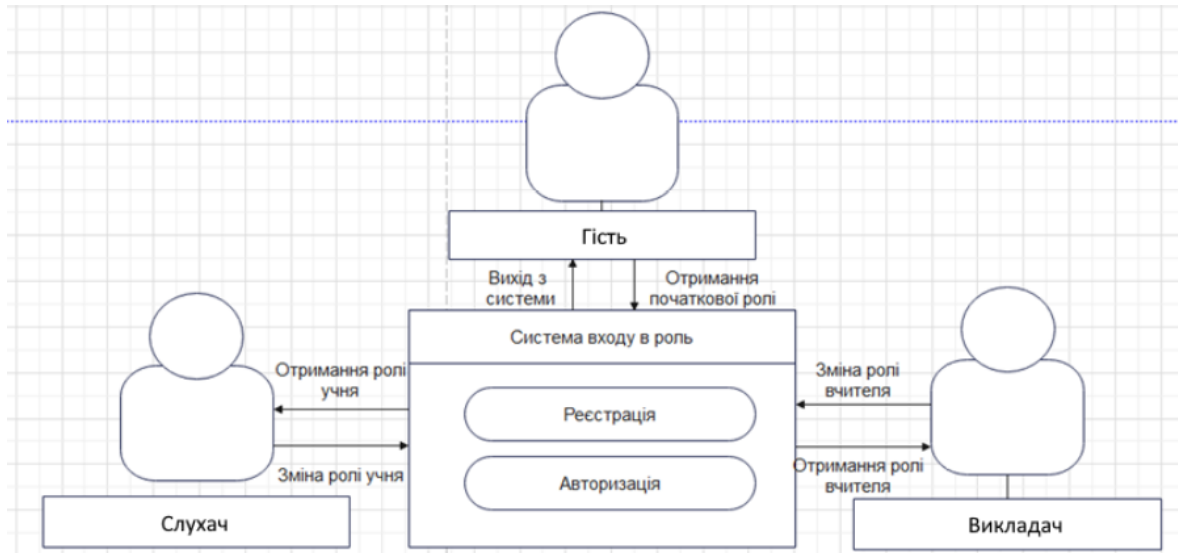


Рисунок 2.7 – Процес зміни ролі

Якщо ж користувач отримує роль викладача, окрім цього він може надавати в систему дані для створення нових курсів та відео. Повний список даних, які може надати системі користувач, можна побачити в першій частині декомпонованої діаграми потоків даних (додаток Б.1). Створена діаграма потоків даних відображає загальну інформацію користувача.

Особливий статус мають гості-користувачі, так як вони можуть лише надати персональну інформацію програмній реалізації методу для отримання доступу до інструментів через реєстрацію та авторизацію. В залежності від введення даних, гість може отримати доступ до певної ролі, або, якщо данні будуть введені не вірно, отримати відмову в якості помилки.

Як зазначалось раніше, уже авторизований користувач також може змінювати власні ролі, через що йому доступні процеси авторизації. Для сутності користувача також можуть бути доступні процеси як і слухача, так і викладача, через описану особливість. Список процесів, які може ініціалізувати користувач, можна побачити на наступному у додатку Б.2.

При створенні через інтерфейс запитів до БД, користувач може отримати або сторінку з відеороликом, або інформаційну сторінку. Термін “Інформаційна сторінка” – це, у даному контексті, термін, який надає будь-яку нединамічну інформацію, таку як наприклад дані про курси, викладачів або вебсайт.

Також можливий варіант, при якому запит до БД буде недійсний, через

що формується звіт про помилку, який повідомляє про причину відмови. Наприклад, при запиті до неіснуючого відео, може відправитись помилка з кодом 404, або 400, що значить помилку в запиті.

Якщо користувач спробує створити персональний курс, він сам відправить дані обмежень, після чого відбудеться виконання генетичного алгоритму для створення списку ідентифікаторів відео. Дана інформація має бути записана в даних користувача, наприклад в браузері, в вигляді cookie-файлів, або в базі даних, для повторного використання уже створеного раніше переліку навчальних відео. Після того, як дані записані, ідентифікатори відправляються разом з відеоматеріалом до представлення, яке в свою чергу, направляється до користувача у вигляді списку.

Якщо користувач має роль викладача, він може отримати доступ до відповідних процесів, які зосереджені на створенні нового навчального матеріалу у вигляді курсів, які містять в собі відеоролики. Діаграма з обробкою наданих викладачам даних зображено у додатку Б.3.

Як і у випадку з процесами користувача, надсилання даних може викликати помилки, через що результатом може бути як і звіт про коректно додану інформацію, так і про не проходження даними валідації.

Концепція ролей у користувачів на платформі передбачає обмеження функціоналу залежно від статусу. У користувачів, які не є викладачами, доступ до функцій обмежений, оскільки вони не є довіреними особами для програмної реалізації методу. Таким користувачам доступні такі блоки, як "Головна сторінка", "Про нас", "Навчальні курси", "Викладачі" та "Зворотній зв'язок". Також вони можуть виконувати певні дії, які обмежуються переглядом відеокурсів, а також генерацією власних курсів та аутентифікацією. Повний перелік функціоналу доступного слухачу, зображено на рисунку 2.8.

У викладачів є майже повний доступ до програмної реалізації методу, оскільки вони вважаються особами, які мають можливості як для навчання, так і для створення нового освітнього матеріалу. Вони можуть взаємодіяти з такими блоками, як "Головна сторінка", "Списки відтворення", "Навчальний матеріал" та "Вихід із опублікованого запису".

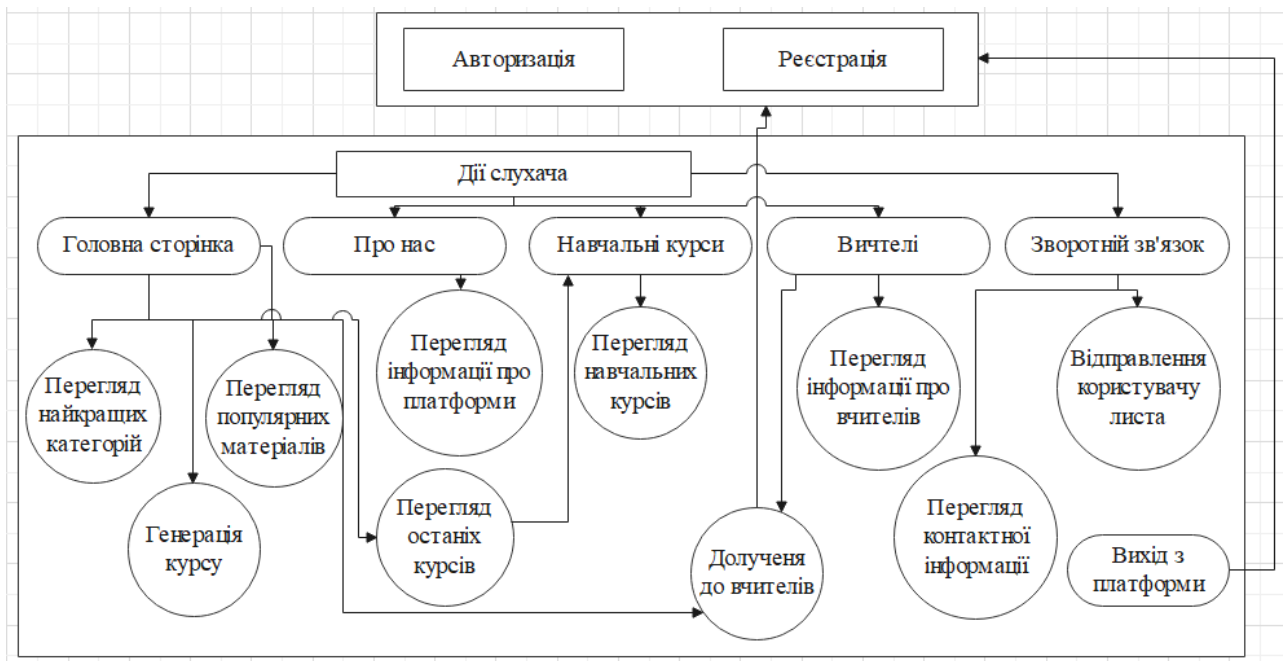


Рисунок 2.8 – Функціональні блоки слухача

Також викладачі можуть виконувати ряд додаткових дій на кожній сторінці. В основному, дії викладача полягають у перегляді, створенні та редагуванні власних навчальних курсів та списків відтворення, у яких вони будуть зберігатись. Потрібно зазначити, що викладачі мають доступ лише для власних курсів і не можуть ніяк впливати на інформацію надану іншими, що дозволяє зберегти курси від зловмисників.

Важливо те, що використання функцій слухача, які також доступні викладачу, можуть бути корисними при перегляді доданого власного навчального матеріалу, перевірки коректності взаємодії курсу разом з генератором курсів і для власного самонавчання при перегляді курсів від інших викладачів. Повний список функціоналу, який доступний викладачам, можна наочно побачити на рисунку далі (рисунок 2.9).

Цей розподіл функціоналу дозволяє забезпечити ефективність користування програмною реалізацією методу для кожної категорії користувачів. Він сприяє покращенню процесу навчання, забезпечуючи відповідність інструментів потребам кожного користувача.

Отже, програмна реалізація методу, яка розробляється, має ряд функцій

які потрібно реалізувати, доступ до яких має бути здійснений через систему ролей. Також потрібно забезпечити нормальну працездатність, з обробкою помилок, які будуть відбуватись під час взаємодії з даною системою

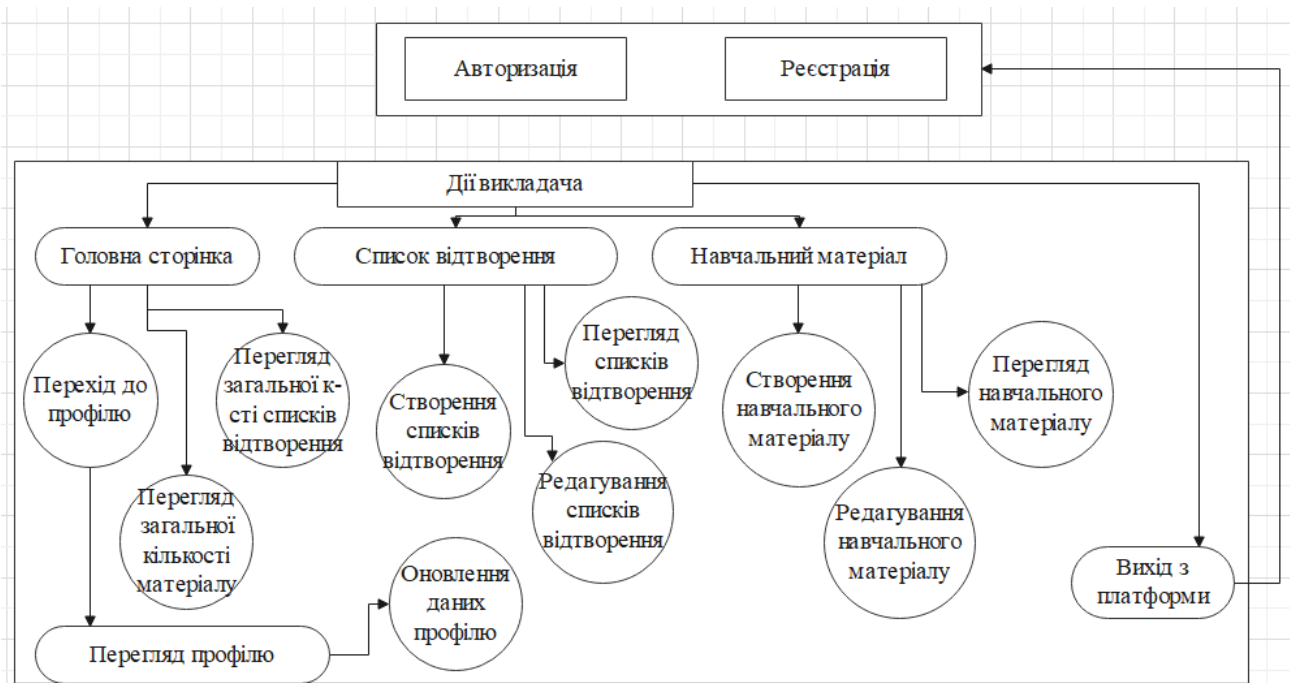


Рисунок 2.9 – Функціональні блоки викладача

2.7 Розробка архітектури навчального вебресурсу з функцією генерації курсів на основі генетичного алгоритму

Відповідно до функціоналу та методу генерації курсів на основі генетичного алгоритму, розроблених у попередніх розділах, можна спроектувати архітектуру цілісної системи, за для подальшої її реалізації.

Почати варто з опису головних компонентів додатку. Узагальнену схему програмного комплексу було представлено на Рисунку 2.10.

Початковим компонентом для всіх користувачів, будь яких ролей, є “Управління доменом додатку”. Дана частина інформаційної системи виділена, як окрема система функцій через специфіку вебдодатків. Зазвичай головна сторінка прив’язана до домену програмної реалізації методу. Починаючи з цього компонента, користувач може перейти до базових функцій. Також, передбачається, що обмеження для генерації курсів, будуть передаватись саме з

цієї сторінки. Якщо в програмній реалізації методу будуть виникати певні помилки, то в першу чергу повідомлення про них буде надаватись саме у цьому блоці, або користувача буде переадресовувати також до даної системи.

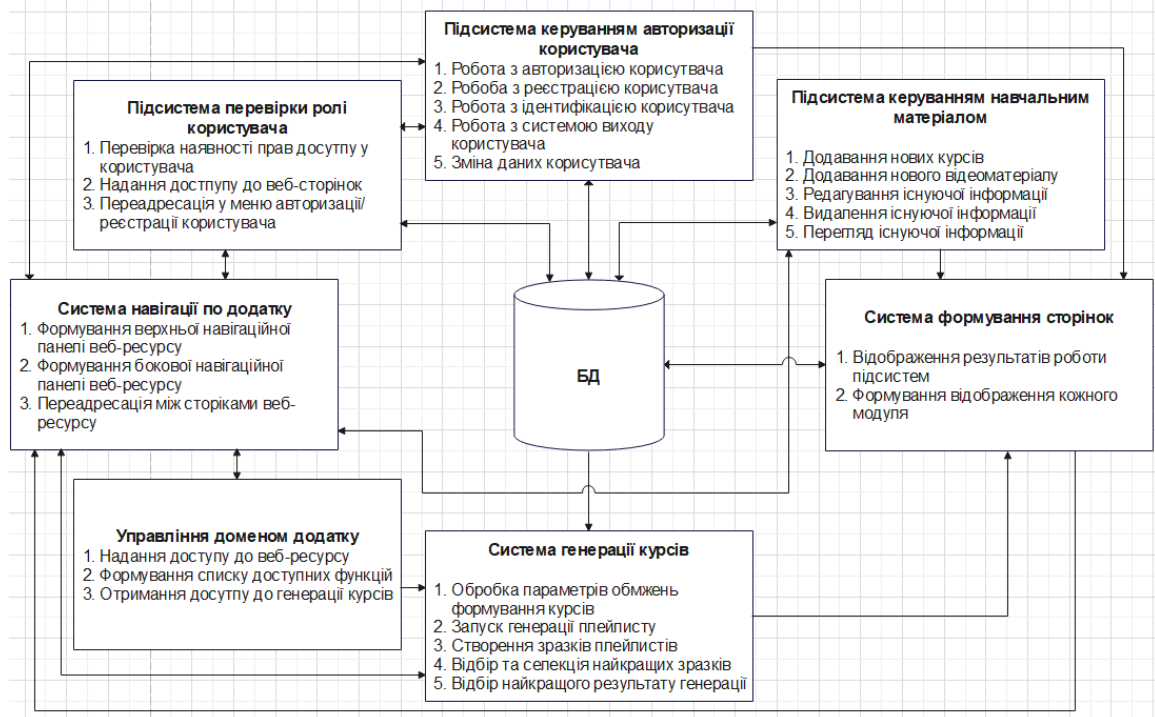


Рисунок 2.10 – Схема навчального вебресурсу з системою генерації курсів на основі генетичного алгоритму

Переходи між сторінками програмної реалізації методу можуть бути виконані за допомогою навігаційної панелі, яка формується, як окремий компонент. Дана частина містить головні функціональні підблоки системи, потрібні для навігації. Для зручності користування, навігацію варто розбити на дві частини, щоб інструменти були розділені на категорії. Бокове меню (Сайд-бар), потрібне для навігації по сторінкам додатку. Верхнє меню (Хедер), забезпечує функціонування налаштувань вигляду програмної реалізації методу, а також функцій авторизації, реєстрації та виходу з акаунту.

Додатковою задачею даної системи є контролювання переходу користувача між сторінками. Якщо користувач спробує ввести некоректне посилання, дана система має сформувати повідомлення про це. Також, функції даної системи мають бути доступними з будь-якої візуальної частини програмної

реалізації методу. Коли користувач виконує перехід до більшості сторінок, доступ його ролі перевіряються за допомогою підсистеми перевірки ролей.

Якщо користувач, який немає достатньо прав, спробує за допомогою навігації перейти до функціоналу, система отримає повідомлення про це, переадресувавши його до компонентів пов'язаних з авторизацією та реєстрацією. Наприклад, слухач, який спробує отримати доступ до дій редагування курсів, відправиться на сторінку авторизації викладачів, для підтвердження власної особистості та отримання відповідних прав.

Система керування авторизацією та реєстрацією має забезпечувати функції, що дозволяються користувачу отримувати та змінювати ролі. Саме до цієї частини ресурсу відбувається переадресування після перевірки прав. Якщо користувач спробує зареєструватись, при цьому введені ним дані коректні, інформація відправляється до бази даних.

Окрім цього, з системи керування авторизацією, користувачу доступні функції зміни даних про свій профіль. Ці функції пов'язані саме з цією системою, через автоматичну перевірку прав, так як при зміні даних про особистість, потрібно замінити і інформацію для перевірки.

Для викладачів доступні функції пов'язані з роботою з навчальним матеріалом. Для зручного управління цими інструментами, вони віднесені до окремого блоку, що дозволяє, при потребі збільшувати інструментарій, при цьому залишаючись в межах одної групи сторінок. Будь-які дії з навчальним матеріалом супроводжуються доступом до бази даних. Для уникнення проблем, які можуть виникнути через некоректні дані, система має перевіряти коректність даних та повідомляти користувача про це.

Якщо програмна реалізація методу уже має дані пов'язані з курсами та відеоматеріалом, користувачі з статусом слухачів можуть або переглянути їх, або спробувати згенерувати власний курс з відео потрібного напрямлення. Підсистема генерації відео доступна з головної сторінки, де знаходяться поля вводу обмежень часу та тематики, після чого запитує доступні для формування курсів відео з бази даних, в результаті чого, починається формування курсу. Усі складові генетичного алгоритму відбуваються саме у цій підсистемі.

Користувачі, окрім як передачі обмежень, не можуть впливати на процеси, що відбуваються у цьому функціональному блоці. Результатом виконання процесів у підсистем завжди є унікальний курс, створений з відео інших курсів, який передається для відображення до системи формування сторінок.

Підсистема формування сторінок, це набір функцій, які динамічно формують візуальний інтерфейс для користувача. Реалізація даної підсистеми на пряму залежить від обраних інструментів розробки та може сильно відрізнятись. Головною задачею даної системи є відображати дані, які формуються під час інших процесів.

Також відділення алгоритмів та процесів від системи формування відображень, дозволяє реалізувати клієнт-серверну архітектуру, яка характерна для більшості вебзастосунків, забезпечуючи доступ до функціоналу на єдиному носії інформаційної системи, забезпечивши доступ багатьом користувачам, формуючи згруповану ієрархію компонентів (Рисунок 2.11).

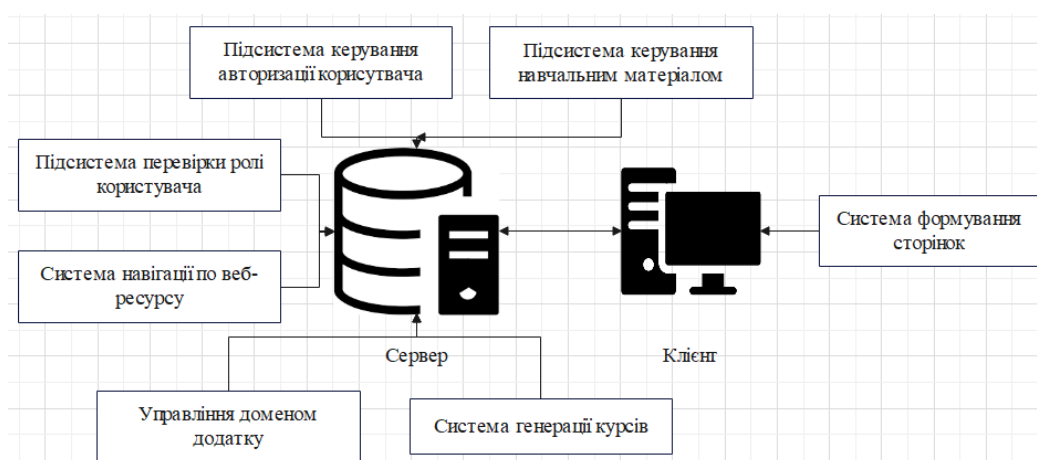


Рисунок 2.11 – Розподіл систем вебресурсу між сторонами клієнт-серверної архітектури

Як висновок, можна сказати, що для успішної реалізації методу потрібно створити 7 основних підсистем, які забезпечать необхідне функціонування даної системи. Окрім цього, потрібно розподілити ці системи у контексті клієнт-серверної архітектури, забезпечивши розподілений доступ до навчального матеріалу, а також віддалений доступ до інструментів.

2.8 Критерії оцінювання покращення вмісту навчальних курсів

Функціональність генетичного алгоритму базується на адаптації курсів до специфічних умов середовища, в якому вони функціонують. Процес селекції в генетичному алгоритмі полягає у відборі найбільш пристосованих курсів за заданими критеріями, що сприяє еволюції загального списку варіантів курсів.

У контексті розробки навчальних курсів, одним із ключових обмежень є час, який користувач готовий витратити на освітній матеріал. Курси, чия тривалість перевищує цей параметр, потребують модифікації. Тому для критерію відбору курсів – найкраще підходить визначення кількості відео у курсі, які максимально закривають бажані компетентності, забезпечуючи максимальне використання встановленого часу. Окрім вищезазнаного, система навчання має бути спрямована на конкретну тематику, що вимагає забезпечення тематичної консистенції та глибини наданих матеріалів. Важливо, щоб навчальні елементи відповідали освітнім цілям та були взаємопов'язані, створюючи цілісний та змістовний навчальний досвід.

Додаткові параметри, які впливають на роботу алгоритму, включають кількість поколінь, які мають бути сформовані, ймовірність мутацій у відеокурсах та загальний розмір списку варіантів курсів. Ефективність генетичного алгоритму визначається його здатністю створювати курси, які не тільки відповідають встановленим обмеженням, але й демонструють здатність до еволюційного розвитку, що підвищує їхню конкурентоспроможність і сприяє формуванню більш адаптованих нащадків.

Для визначення нормативних критеріїв, які використовуються для оцінювання курсів як у початковому поколінні, так і в наступних, важливо зосередитися на аналізі вмісту матеріалів.

У рамках оптимізації навчальних ресурсів у курсі, критично важливо забезпечити унікальність матеріалів. Це обумовлено тим, що повторення одних і тих же навчальних елементів може викликати зниження ефективності навчання та підвищення рівня монотонності, що негативно впливає на залученість

користувачів. Оскільки одним з основних параметрів, на який орієнтується метод, є загальний час курсу, тривалість відео повинна бути адаптована до встановлених часових обмежень. Необхідно також підкреслити, що присутність пустот у списках відтворення відео може перешкоджати генетичному розмаїттю та репродукції унікального потомства, що, в свою чергу, підриває потенціал еволюційного розвитку системи. Також, недопустимі курси, які не мають жодних відео.

Під час вибору найкращого курсу з найкращих, процес відділення включає ідентифікацію всіх елементів з максимальним коефіцієнтом адаптації, який відображає їхню здатність відповідати заданим вимогам. Після цього відбувається сортування цих елементів за загальною тривалістю, відведеного для переліку навчальних відео, і вибирається курс з найменшою тривалістю.

Отже, для ефективної імплементації генетичного алгоритму в метод генерації курсів важливо встановити чіткий критерій для формування контенту конкретної тематики. У процесі селекції найкраще підходить визначення максимальної кількості відео у курсі, яка відповідає вмісту курсу в межах визначеного часу. Ключовим аспектом є також перевірка адаптивності, що включає аналіз нормалізованості та стабільності генетичного матеріалу відносно варіативних умов середовища. Дотримання цих критеріїв приведе до покращення вмісту курсів під час проходження кожної популяції.

2.9 Висновки до розділу 2

Розроблено метод покращення вмісту навчальних курсів за генетичним алгоритмом.

Визначена загальна структура генетичного алгоритму. Виконано імплементацію ГА до вирішуваної задачі, а саме визначення термінів “ген”, “геном”, “особина”, “популяція”, “мутація” в контексті методу. Також, сформовано схему алгоритму для генерації курсів, з використанням принципів нормалізації.

Визначено особливості імплементації генетичного алгоритму для реалізації методу та з'ясовано, що генетичний алгоритм ефективно адаптується

для формування навчального контенту відповідно до варіативних критеріїв.

Описано особливості використання спеціалізованих програмних компонентів, а саме що для управління відеороликами в потоковому режимі для отримання даних, головною використаною утилітою, буде FFmpeg.

Розроблена інформаційна архітектура навчального вебресурсу з функцією генерації курсів, яка потрібна для розміщення інформації про користувачів та відеоролики, які можуть бути використані для генерації курсів. Тому БД для системи була спроектована та нормалізована до третьої нормальної форми. Це зроблено для уникнення аномалій в процесі обробки інформації, яку будуть надавати користувачі.

Описана підготовка робочих вхідних даних для системи формування курсів, яка передбачає, що частину даних, вводять користувачі і потребує ручного введення у БД. З іншого боку, інформація про генерацію курсів повинна бути введена в систему невід'ємно від коду. Це необхідно для тестування та визначення оптимальності навчального програмної реалізацію методу.

Спроектована функціональна структура програмної реалізації методу, яка повинна мати функціонал, доступ до якого здійснюється через систему ролей користувачів, слухачів та викладачів. Також вона повинна забезпечувати нормальну працездатність і обробку помилок.

На основі функціоналу було сформовано архітектуру програмної реалізації методу. Для успішної реалізації важливо створити 7 основних підсистем, які забезпечать необхідне функціонування системи. Ці підсистеми слід розподілити у контексті описаної клієнт-серверної архітектури.

Визначено критерії оцінювання покращення вмісту навчальних курсів. А саме визначення кількості відео у курсі, які максимально закривають бажані компетентності, забезпечуючи максимальне використання встановленого часу.

Розділ 3 Програмна реалізація методу генерації курсів

3.1 Вибір засобів розробки програмної реалізації методу

Для реалізації будь-якого програмного продукту, потрібно обрати мову програмування, на які буде створена логічна частина і суміжні фреймворки. Також, для запуску програмного продукту, потрібно обрати платформу, яка надаватиме потрібні інструменти. Для роботи з інформацією, потрібно обрати СКБД, яка забезпечить передачу даних в код, для функціонування як системи, так і реалізації методу.

Для розробки логічної частини програмної реалізації методу, обрано мову програмування PHP. Її особливість полягає у збірці частин інтерфейсу прямо перед передачею користувачу. Також, дана мова програмування дозволяє створювати скрипти, які оброблятимуться лише сервером, з можливістю переадресувати користувача, забезпечуючи розділення компонентів і виділення класів у власні файли.

Для формування сторінок, обрано мову розміти HTML5 та мову каскадних стилів CSS3. Даний вибір є найчастішим для формування подібного роду застосунків. Для того, щоб зосередитись на розробці методу і його алгоритмах, для зменшення часу на написання інтерфейсу користувача, буде застосовано бібліотеку стилів Bootstrap. Для обробки даних користувачів і для надсилання запитів до серверу, буде використано JS.

Для розгортання сервера, вирішено використовувати платформу XAMPP, яка пристосована до формування сторінок з використанням PHP. Також, вона надає інструменти для роботи з такими СКБД, як MongoDB та MySQL. Тому, з доступних СКБД, використано MySQL, через надання мінімально необхідних функцій, для реалізації проекту.

Отже, для розробки інформаційної системи з генерацією курсів, було обрано мови: PHP, JS, HTML та CSS. Для фокусування на розробці функцій, обрано бібліотеку стилів Bootstrap. Для розгортання серверу застосовано платформу XAMPP і СКБД MySQL для кращої взаємодії компонентів.

3.2 Структура та функціональне призначення програмних складових системи

Створення програмної реалізації методу розпочиналась з створення загального шаблону сторінки. Він складається з кількох компонентів, які компонуються на усіх сторінках (рисунок 3.1).

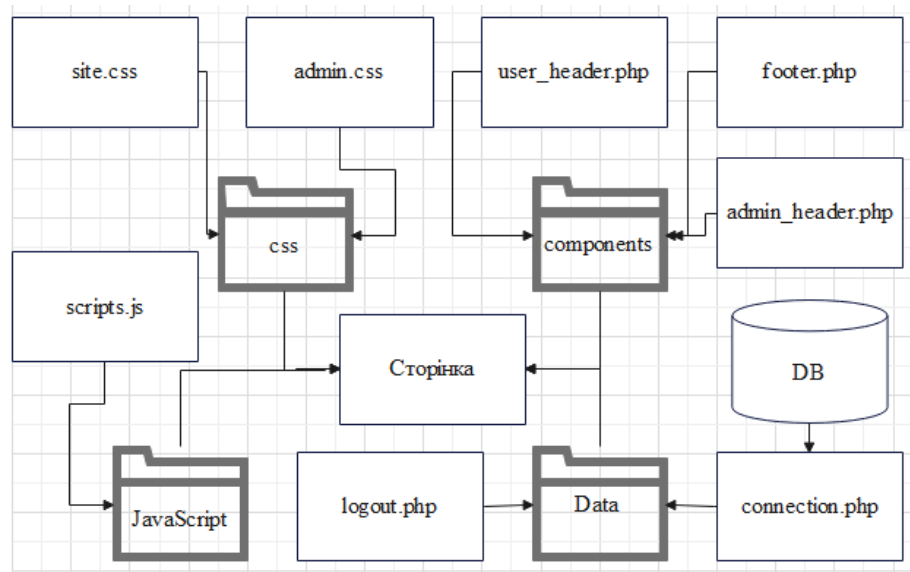


Рисунок 3.1 – Структура компонування сторінок

Таким чином, було сформовано компоненти “admin_header.php”, “user_header.php”, “footer.php”. файли з приміткою “_header”, забезпечують створення панелей інструментів для викладачів та слухачів. Компонент “footer.php” описує на сторінці інформацію про програмну реалізацію методу в нижній частині сторінок користувачів.

Для забезпечення підключення бази даних до сторінок, було створено компонент “connection.php”, який містить шлях до бази даних у файловій системі. Такий підхід дозволяє з будь-якої частини додатку звертатись до централізованого компоненту, створюючи підключення лише в єдиному екземплярі, який при потребі, просто замінити.

Якщо в майбутньому, базу даних потрібно буде перемістити, для підключення доведеться змінити текст лише в одному компоненті. А щоб надати

можливість виходу з акаунту з будь-якої сторінки, було створено компонент “logout.php”, який видаляє дані про користувача з файлів cookie.

Для контролювання динамічних стилів, застосовується JavaScript, який наприклад, дозволяє переключати тему сторінок на світлу та темну, змінює повзунок та дозволяє динамічно міняти адаптацію стилів.

Кожна сторінка представляє окремий PHP-файл, який описує формування відображення для користувачів. ХАМРР дозволив відмовитись від системи класів для більш зручного управління сторінками з серверу, komponуючи логіку та HTML-код. Тому майже усі представлені сторінки можна вважати незалежними компонентами.

Для комфорту розробки сторінок, файли для ролей слухача та викладача були розподілені на дві директорії: user і admin. User містить усі сторінки, доступні усім користувачам, в той час, як admin лише файли викладачів. Тому, наприклад, для реєстрації та авторизації було встановлено файли “registration.php” та “login.php” для обох директорій, кожна з яких потребує введення даних відповідних ролей. Таким чином, процеси для різних типів користувачів не пересікаються.

Викладачі можуть переглядати власні курси, перейшовши через навігаційну панель до “playlists.php”. Для створення курсів, для них створена сторінка “create_playlist.php”, які доступні лише з списку курсів. Перша сторінка призначена для створення курсу. Для того, щоб матеріал можна було додавати в будь-який момент, реалізовано модуль “create_video.php”. Також, для оновлення інформації і видалення створено модулі з префіксами “update_” і “delete_”.

Головна сторінка для викладачів представлена файлом “dashboard.php”. Даний модуль надає статистичну інформацію про акаунт, а також про контент викладений раніше, з функціями швидкого переходу. Таким чином, з даного модуля можна перейти до “contents.php”, який надає альтернативний доступ до викладених відеороликів, в порядку, в якому вони були додані, або оновлені. Повна карта вебзастосунку для викладачів представлена у додатку В.1

Головна сторінка для слухачів, спільна для сторінки гостей і представляється файлом home.php. Дана сторінка має доступ до більшості інших

сторінок і надає навігаційні інструменти для швидкого переходу до перегляду курсів та відеороликів.

З головної сторінки можна перейти до файлу генерації курсів. Даний функціонал забезпечується файлом “generate_playlist.php”. Для модуля генерації немає представлення і він є проміжний. Після того, як сформується курс, користувача перенесе до файлу “quick_playlist.php”, в якому надаватиметься список відео згенерованого курсу.

Ще одним способом знайти відео є перехід на сторінку “courses.php”, яка надає список курсів. Курси розміщені в порядку їх додавання вчителями. З даного модуля можна перейти до “playlist.php”, яка міститиме відеоматеріал, який надається викладачем. Перегляд кожного окремого відеоролика відбувається на сторінці “watch_video.php”.

Також, слухачам надаються додаткові сторінки “contact.php”, яка дозволяє написати відгук про пройдені курси, і “about.php”, що надає коротку інформацію про застосунок. Для того, щоб переглянути відеоролики конкретних вчителів, можна перейти на сторінку “teachers.php”. Також, на даній сторінці слухачу пропонується долучитись до викладачів. Карта дій слухачів представлена у додатку В.2.

Повна карта програмної реалізації методу, з взаємозв'язками сторінок слухачів та вчителів представлено у додатку В.3. Основні сторінки інтерфейсу, з усіма скомпонованими елементами, представлено у додатку Г.

Отож, в процесі реалізації, було спроектовано модулі, які поділено на декілька категорій, при цьому, навколо кожної сторінки, яка представлена даними модулями, розміщені елементи компонування. Розроблена програмна реалізація методу представляє структурований, за правилами логіки користувачів, набір PHP файлів, з можливістю переходу між сторінками, які надані як слухачам, так і викладачам.

3.3 Особливості реалізації генерації курсів

Найважливіший компонент програмної реалізації методу у дослідженні

методу представляється класом “generate_playlist.php”, який містить в собі реалізовані класи для генерації курсів.

Структура методу побудови курсів на основі генетичного алгоритму включає в себе систему етапів, які можуть бути представлені як набір методів, які обов’язкові для функціонування алгоритму. Таким чином можна виділити наступні функціональні блоки:

- initializePopulation – ініціалізація нульового списку варіантів курсів;
- evaluatePopulation – оцінка адаптованості курсів;
- selection – відбір найкращих курсів;
- crossover – схрещування курсів;
- mutate – проведення мутацій;
- array_unique – нормалізація результатів;
- getBestSolution – відбір найкращого курсу.

Враховуючи, що генетичний алгоритм має обмеження, які можуть надходити ззовні і змінюватись в залежності від потреб програмного забезпечення, потрібно визначити і основні змінні, якими буде оперуватись описаний метод. Таким чином можна виділити наступні властивості:

- videos – список доступних відео;
- targetDuration – обмеження часу відео для кожного курсу;
- populationSize – кількість варіантів курсів;
- generations – кількість поколінь, які мають пройти для отримання потрібного результату;
- mutationRate – шанс того, що відео в курсі зміняться.

Такі властивості, як populationSize, generations та mutationRate потрібні для точності результатів і мають роль гіперпараметрів алгоритму.

Відповідно список властивостей реалізовано таким чином:

- id – ідентифікатор відео;
- duration – час відеоматеріалу;

Головним параметром кожного відеоролику є його час, ідентифікатор потрібний для того, щоб алгоритм міг розпізнавати унікальність відеороликів та

видавати в результаті мінімально необхідну інформацію для подальшого відтворення відеороликів на сторінках додатку. Таким чином робота алгоритму абстрагується від зовнішнього середовища і вигляд зберігання вхідних даних стає неважливим, так само як і їх структура, тому що перед початком роботи в будь-якому випадку потрібно модифікувати їх під норми, які потребує метод.

Поєднання методів та властивостей алгоритму і відеороликів, можна виконати за допомогою компонування. При такому підході, вхідними і вихідними даними будуть списки з даними описаного класу. Відповідно, діаграму описаних класів можна зобразити як наступний зв'язок (рисунок 3.2).

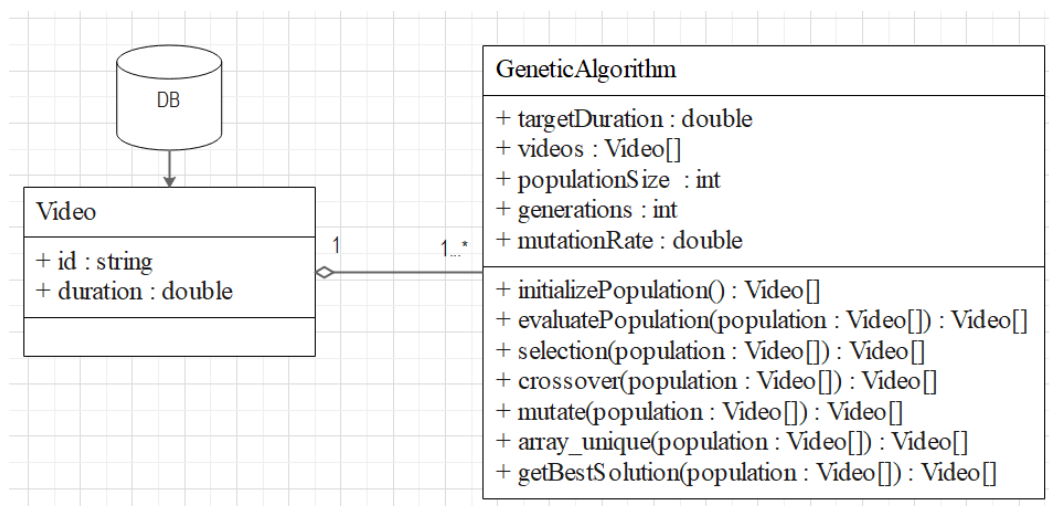


Рисунок 3.2 – Діаграма класів генетичного алгоритму

Ідентифікатор і тривалість, приведена до спільної одиниці вимірювання – секунд, передаються у конструктор класу Video, для формування списку абстрактного представлення відеороликів.

Робота компоненту починається з надсилання форми, яка містить параметри користувача для генерації, які надсилаються до фалу генерації курсів. Передані властивості, отримуються за допомогою POST-методів через PHP і записуються у змінні.

Для того, щоб сформувати список доступних відеороликів, реалізовано SQL-запит, який спочатку обирає курси викладачів, які містять обрану користувачем тематику, після чого, відповідно до ідентифікаторів курсів,

формується список відеороликів.

Інформація про кожний відеоролик приводиться до абстрактної форми. Ідентифікатор використовується такий самий, як ідентифікатор у базі даних. Щоб отримати тривалість відео, застосовується виконання консольного запиту до утиліти FFmpeg, яка надає усю інформацію про кожний відеоролик у текстовому форматі, при цьому час отримується застосуванням пошуку за регулярним виразом.

Усі відео передаються у клас GeneticAlgorithm, який застосовується для налаштування параметрів генетичного алгоритму, а також його запуску. Параметри користувача і гіперпараметри передаються в якості аргументів конструктора, надаючи налаштований екземпляр класу, який запускається методом “run”.

Для формування початкового списку курсів створюється копія переданих відео. Далі, для формування курсів, циклічно обирається випадковий елемент з масиву, який поміщається в якості відео у курс. Відео, яке вже було додано, видаляється з початкового списку навчального матеріалу. Процес додавання відбувається до того моменту, доки додавання нового відео в перспективі не приведе до перевищення доступної тривалості курсу. Дані дії відбуваються для кожного курсу, при цьому копія відеоматеріалу кожного разу поновлюється.

Після формування списку курсів, вони входять в цикл. Першим етапом в цьому процесі є оцінка якості створених осіб, що включає підрахунок відеороликів в масиві. Використовуючи властивість РНР, а саме спорідненість масивів і об'єктів, властивість “fitness”, що позначає пристосованість елемента, записується як параметр даного списку.

На етапі відбору курсів, усі масиви сортуються методом “usort”, якому передається делегат, який ранжує за показником пристосованості кожний елемент, таким чином створюючи список, де найкращі позиції займають елементи з найвищим показником якості. Розмір масиву курсів ділиться на два, а отримане значення округлюється до меншого, після чого гірша половина списку курсів знищується. Також, на даному етапі прибирається властивість “fitness” для коректної роботи усіх наступних методів.

Перед комбінуванням курсів, вони паруються. Якщо останній елемент не має пари, він схрещується з самим з собою. Для поділу списку відео кожного батьківського елемента, застосовується метод “array_slice”, а для комбінування значень в нащадках “array_merge”.

Для заміни випадкового відео, спочатку перевіряється кожний курс, чи отримає вона новий відеоролик. Для цього, генерується випадкове число від 0 до 1 і порівнюється з гіперпараметром шансу мутації. У випадку, якщо елемент мутує, у нього замінюється випадкове відео на випадковий елемент з початкового списку, або додається новий. Динамічні масиви PHP не мають статичних розмірів, що дозволяє додавати нові відео без втрати швидкості.

Останній етап в ітерації представлений нормалізацією. Спочатку, перевіряється, чи в результаті метаморфоз загальна тривалість не перевищує допустиму. У випадку, коли тривалість недопустима, обирається випадковий відеоролик, який видаляється. Перевірка і встановлення унікальності курсів забезпечується функцією “array_unique”.

Коли пройде кількість ітерацій, яка рівна кількості допустимих поколінь, відбувається вихід з циклу, після чого усі значення ранжуються по кількості відео, а другим параметром ранжування є загальна тривалість курсу. В результаті на першому місці в масиві встає елемент з найбільшою кількістю відео, але з найменшою кількістю загальної тривалості відеороликів.

Отриманий набір відео встановлюється в cookie, після чого відбувається переадресація на сторінку перегляду відео. Для формування списку відео, створюється SQL-запит, який шукає усі відеоролики з заданими ідентифікаторами, після чого відбувається формування відображення згенерованого курсу.

Отже, реалізований метод використовує багато аспектів мови PHP, що дозволило спростити реалізацію деяких функцій. Основна складова методу містить в собі два основних класи, Video та GeneticAlgorithm, які використовуються дані отримані з POST-форм, при цьому останній з них записує результати виконання у файли cookie для подальшого використання.

3.4 Тестування працездатності та ефективності генерації курсів

Тестування розробленого методу був важливим етапом, для визначення його роботи при різних умовах. Для перевірки загальної дієздатності, для початку було створено ряд Test Case. Для усіх тестувань, гіперпараметри визначено, як сталі параметри: кількість варіантів курсів – 100, кількість поколінь – 100, шанс мутації 10%.

Поведінка при нормальних умовах передбачає введення параметрів часу, який більший за мінімальний час будь-якого відеоролику з списку, при цьому менший за загальний час відеороликів у базі даних. Список в результаті має бути не пустим. В результаті, за описаними умовами, було створено перший Test Case, зображений нижче (таблиця 3.1).

Таблиця 3.1 – Тестування при нормальних умовах

Дія	Очікуваний результат	Дійсний результат
ПЕРЕДУМОВИ		
Ввести час в розмірі 15 хвилин	Система дозволить ввести дані	Passed
Ввести параметр теми “Backend”	Система дозволить ввести дані	Passed
ОПИС ТЕСТОВОГО ВИПАДКУ		
Запуск генерації	Почнеться завантаження сторінки	Passed
Очікування генерації	Генерація займе менше секунди	Passed
Відображення списку курсів	Виведеться список з лише унікальних відео з теми “Backend” і часом меншим за 15 хв.	Passed
ПІСЛЯУМОВИ		
Натискається кнопка “Головна”	Перенесе на головну сторінку	Passed

В результаті, тестування при нормальних умовах, надало коректні результати. Час генерації відео займав приблизно одну мікросекунду, що є доволі хорошим результатом. Всі відео відповідали тематиці. Проміжний етап, з відображенням даних, можна побачити на наступному рисунку (рисунок 3.3).

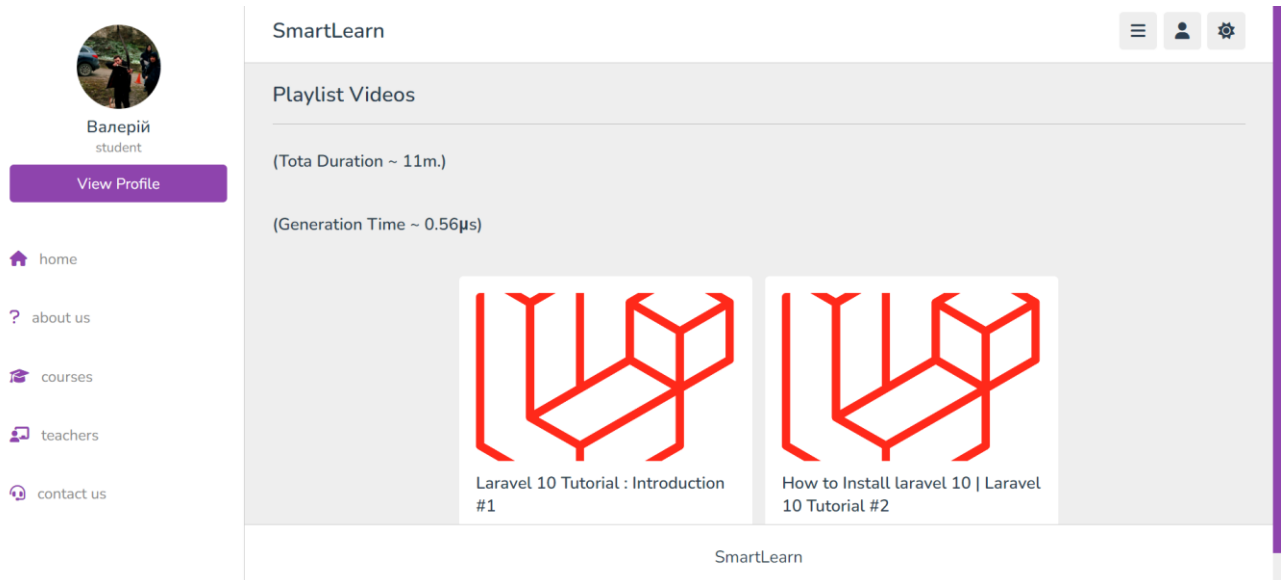


Рисунок 3.3 – Результати генерації курсів при нормальних умовах

Далі було проведено тестування в стресових умовах. Враховуючи, що вибір тегів обмежений і не може бути змінений користувачами, було протестовано введення чисел, які можуть здатись некоректними для системи. Результати випробувань зображені в наступній таблиці (таблиця 3.2).

Відповідно до таблиці 3.2, можна зробити висновки, що введення некоректних значень присікаються системою, повідомляючи користувача, про потребу змінити дані (рисунок 3.4). При веденні часу, більшого за максимальний можливий, буде згенеровано список з усіх можливих відеороликів.

Спроби відправити дані на сервер через інші інструменти клієнтської частини, мали такі ж самі результати, як відправка через форму.

Також, натискання кнопки переходу на сторінку власного курсу, відкриває сторінку з уже раніше згенерованими даними, при цьому процесу регенерації не відбувається, а використовується інформація лише з cookie. При цьому швидкість відновлення даних відбувається за коротший час, чим

генерація, що зумовлено лише процесом звернення до БД.

Далі було проведено тестування ефективності алгоритму. Для цього було створено файл, який має десять тисяч абстракцій відеороликів, в якому перший стовпець відповідає ідентифікатору, який в свою чергу дорівнює порядковому номеру рядку. Другий стовпець позначає випадкову тривалість відеороликів довжиною від 60 до 600 хвилин. В якості довжини абстрактного курсу, який формується цільова функція була 1000 хвилин.

Таблиця 3.2 – Тестування системи в стресових умовах

Case Taste генерації за нормальних умов. Має пройти повний цикл генерації курсів.		
Дія	Очікуваний результат	Дійсний результат
ПЕРЕДУМОВИ		
Ввести параметр теми “FullStuck”	Система дозволить ввести дані	Passed
ОПИС ТЕСТОВОГО ВИПАДКУ		
Запуск генерації з часом в -1 хв.	Система повідомить про некоректність	Passed
Запуск генерації з часом в 0 хв.	Система почне генерацію	Passed
Очікування генерації	Час генерації займе менше секунди	Passed
Відображення курсу	Буде відображено курс без відео	Passed
Повернення на головну сторінку	Відобразиться головна сторінка	Passed
Запуск генерації з часом в 10000 хв.	Система почне генерацію	Passed
Очікування генерації	Час генерації займе менше 1 с	Passed
Відображення курсу	Буде відображено курс з всіма доступними відео тематики “FullStuck”	Passed

Quick Playlist

your free time (minutes) *

-1

your interest *

Full Stuck

Значення має бути більшим або дорівнювати 0.

Create Playlist

Have You Created A Playlist Before?

Рисунок 3.4 – Результати генерації курсів при некоректних даних

Для проведення тестування, використовувався цикл, який спочатку зчитує усі рядки в створеному файлі, після чого, виконувалась операція створення курсу за генетичним алгоритмом. Для одної і тієї ж кількості файлів, операція генерації відбувалась десять разів, після чого підраховувалась середньоарифметична кількість відео, які були додані до курсу, а також середній час виконання операцій з такою кількістю файлів. При виконанні усіх операцій, кількість матеріалу з файлів скорочувався в десять разів і тестування повторювалось знову. Варто зазначити, що дані в файлі не змінювались для кожного окремого тестування.

Щоб задокументувати інформацію, використовувалось логування даних у браузерному вікні (рисунок 3.5).

Усі середньостатистичні дані тестування ефективності для різної кількості вхідних даних, представлено нижче (таблиця 3.3).

Таблиця 3.3 – Середні значення тестування для різної кількості даних

Кількість файлів	Середня кількість відеороликів	Середній час виконання
10000	9.6	~45.13 секунд
1000	8.6	~11.27 секунд
100	8.4	~138.24 мілісекунд
10	6	~24.92 мілісекунд
1	1	~12.27 мілісекунд

```

Кількість відео: 6, час виконання ~ 22.672176361084 мс
Кількість відео: 6, час виконання ~ 31.615018844604 мс
Кількість відео: 6, час виконання ~ 22.730112075806 мс
-----

Середній результат для 10 файлів:
-Середній час виконання: 24.926805496216 мс
-Середній кількість відео: 6
-----

Кількість відео: 1, час виконання ~ 13.58699798584 мс
Кількість відео: 1, час виконання ~ 11.788845062256 мс
Кількість відео: 1, час виконання ~ 10.707855224609 мс
Кількість відео: 1, час виконання ~ 13.720989227295 мс
Кількість відео: 1, час виконання ~ 13.411998748779 мс
Кількість відео: 1, час виконання ~ 13.818979263306 мс
Кількість відео: 1, час виконання ~ 15.558004379272 мс
Кількість відео: 1, час виконання ~ 10.06817817688 мс
Кількість відео: 1, час виконання ~ 10.759115219116 мс
Кількість відео: 1, час виконання ~ 9.3648433685303 мс
-----

Середній результат для 1 файлів:
-Середній час виконання: 12.278580665588 мс
-Середній кількість відео: 1

```

Рисунок 3.5 – Логи з інформацією про тестування швидкодії

Як видно з таблиці 3.3, при сталих гіперпараметрах, швидкість виконання прямо залежить від кількості наданої інформації для обробки.

Для тестування залежності швидкодії програмного продукту від гіперпараметрів, алгоритм перевірки було запущено з параметрами кількості курсів та кількості поколінь, зменшивши до значення 10, відповідно. Результати даного тестування зображені нижче (таблиця 3.4).

Швидкість роботи алгоритму значно виросла, але при цьому точність результатів для великої кількості файлів значно впала, що свідчить про залежність ефективності від правильно підібраних параметрів.

Отже, протестувавши метод генерації курсів, можна зробити висновки, що розроблена реалізація, як і алгоритм, стійкі до несподіваних вхідних даних і

будуть працювати з будь-якою інформацією. Генетичний алгоритм добре працює з малою кількістю інформації, що може бути спричиненою недосконалою роботи з масивами даних, через що в залежності від кількості опрацьованих файлів, швидкодія може падати.

Таблиця 3.4 – Середні значення тестування при менших гіперпараметрах

Кількість файлів	Середня кількість відеороликів	Середній час виконання
10000	7.3	~3.01 секунди
1000	6.3	~43.56 мілісекунд
100	6.2	~5.49 мілісекунд
10	5	~0.3 мілісекунд
1	1	~0.1 мілісекунд

3.4 Оптимізація гіперпараметрів

Для оптимізації роботи генерації курсів, потрібно визначити, які значення гіперпараметрів дозволять знайти співвідношення часу до точності обчислень, щоб збільшити ефективність роботи з великим об'ємом інформації.

Для початку було створено графік залежності параметрів кількості курсів та кількості поколінь. Для цього було використано інструмент Plotly.js, який надає можливість створення 3D-відображень для різних точок.

В якості вісі абсцис були застосовані точки, які відповідають різній кількості курсів, а для всі ординат було застосовано кількість поколінь. В якості вісі аплікат використовуються результати запуску генетичного алгоритму з параметрами, які динамічно оновлюються.

Для того, щоб сформувати матрицю результатів для параметрів, використовувались вкладені цикли. Цикл першого рівня відповідав за зміну кількості варіантів курсів від 1 до 40. Цикл другого рівня, передбачав зміну кількості поколінь. При цьому, шанс мутацій завжди був сталим з параметром, який відповідає значенню 0.1.

Щоб результати створення графіків були точними, використовувався цикл третього рівня, який запускав виконання алгоритму 20 разів. Це пов'язано з неоднозначністю результатів генетичного алгоритму, що потребує пошук середнього арифметичного кінцевих значень.

Для створення графіку, використовувались реальні значення з бази даних, при цьому тегом для формування курсів був "Full Stuck", а обмеження в тривалості визначалось, як 25 хвилин.

Результати формування графіку на основі кількості поколінь та популяцій, зображено на рисунку 3.6.

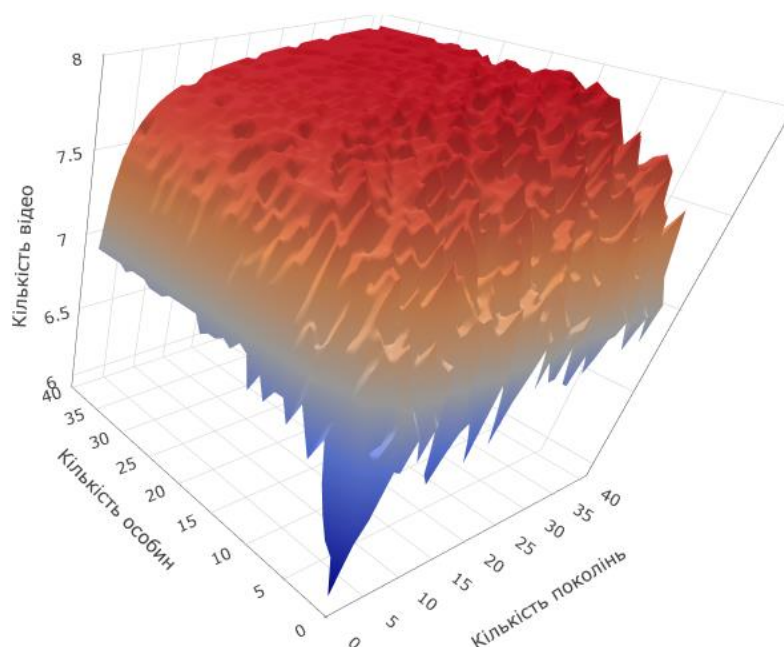


Рисунок 3.6 – Графік точності створення курсу на основі кількості варіантів курсів і кількості поколінь

Як видно з рисунку 3.6, поверхнева площина утворена логарифмічною залежністю. При мінімальних значення кількості курсів та поколінь, результати тестування видають результати, які далекі від очікуваних. Збільшення кількості поколінь з сталою мінімальною кількістю курсів покращує результат, але при цьому, як видно з графіку, приблизно до 20 курсів, середня кількість відео дуже не стабільна, що проявляється в сильних перепадах між результатами.

Рух в сторону збільшення кількості курсів характеризується більш чітким

і плавним переходом кількості згенерованих відео, але при цьому низький показник поколінь сприяє незадовільним результатам.

Плавний перехід може бути пов'язаний з можливістю генерації оптимальних, або близьких до цього, курсів уже на перших поколіннях, після чого вони отримують домінуючі позиції, даючи схоже на себе потомство. Також, це показує пріоритетність кількості відео в списку курсів перед кількістю поколінь, які мають пройти.

Нехтування кількістю поколінь, може спричинити зменшення швидкості виконання алгоритму, через велике навантаження на пристрої в результаті формування більшої кількості масивів відео в курсах, тому, оптимальна точка даних параметрів, може бути досягнута при діагональному русі по графіку, при цьому визначатиметься стабільністю результатів і високою швидкодією.

Для визначення залежності шансу мутацій, було спочатку створено графік, в якому комбінацією параметрів виступали кількість поколінь і ймовірність мутацій, при цьому параметр кількості курсів був сталим на протязі всього дослідження залежності.

Результат комбінування кількості поколінь й ймовірності випадкових змін, представлено на наступному графіку (рисунок 3.7)

Графік (рисунок 3.7) демонструє, що випадкові зміни мають не сильний вплив на кінцевий результат, тому що результативність генерації курсів варіюється від 8 до, приблизно 7.8. Але при цьому можна побачити, що значення в діапазоні ймовірностей від 0.1 до 0.2, мають найкращий показник генерації.

Для визначення того, які комбінації параметрів можуть бути ефективними при виконанні алгоритму, було запущено процес тестування з трьома рівня вкладеності циклів, де кожний з циклів відповідає зміні параметрів. Відповідно до інформації отриманої з графіків, цикл, що відповідає за зміну кількості курсів, змінюватиметься від 15 до 40. Це саме стосується і зміни кількості поколінь. Ймовірність мутацій змінюється від 0.1 до 0.2.

Для коректності вимірів, алгоритм з кожною комбінацією гіперпараметрів виконувався десять разів, після чого використовувались середньоарифметичні значення результатів. Тестування відбувалось з використанням файлу з

абстрактними відеороликами у розмірі в 400 позицій.

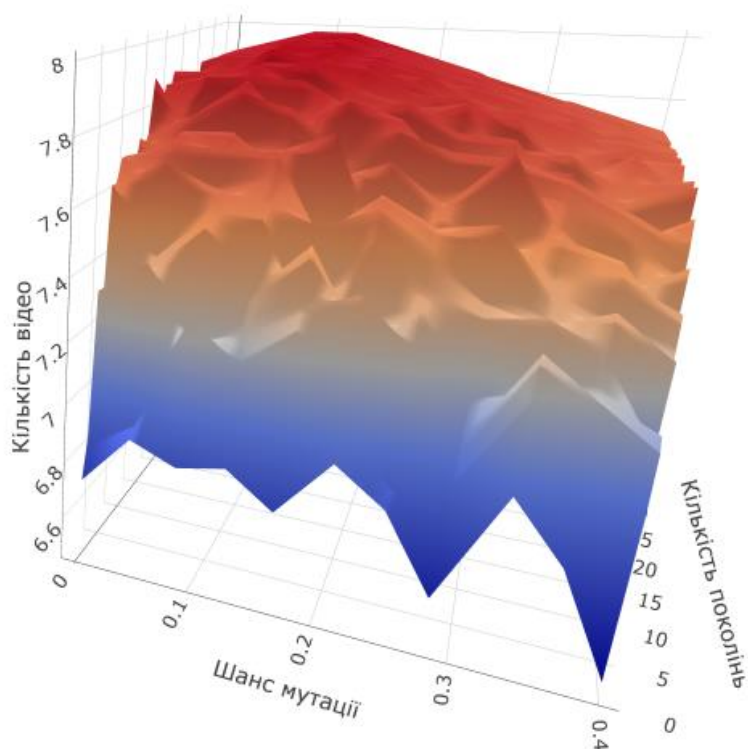


Рисунок 3.7 – Графік точності створення курсу на основі кількості поколінь і ймовірності випадкових змін в курсах

Результати перебору оптимальних рішень зображено на наступному рисунку в якості логів (рисунок 3.8).

Кількість особин: 40, кількість поколінь: 39, шанс мутації: 0.24 || Середня к-сть відео: 22.031470681923, середній час (мс): 53.496026992798
 Кількість особин: 40, кількість поколінь: 39, шанс мутації: 0.26 || Середня к-сть відео: 20.812975139879, середній час (мс): 52.206802368164
 Кількість особин: 40, кількість поколінь: 39, шанс мутації: 0.28 || Середня к-сть відео: 21.192427686401, середній час (мс): 55.343008041382
 Кількість особин: 40, кількість поколінь: 39, шанс мутації: 0.3 || Середня к-сть відео: 20.972741195512, середній час (мс): 54.515790939331,
 Кількість особин: 40, кількість поколінь: 40, шанс мутації: 0.1 || Середня к-сть відео: 21.274375839391, середній час (мс): 50.580406188965,
 Кількість особин: 40, кількість поколінь: 40, шанс мутації: 0.12 || Середня к-сть відео: 21.094616296963, середній час (мс): 48.250198364258
 Кількість особин: 40, кількість поколінь: 40, шанс мутації: 0.14 || Середня к-сть відео: 21.11459245518, середній час (мс): 48.837184906006,
 Кількість особин: 40, кількість поколінь: 40, шанс мутації: 0.16 || Середня к-сть відео: 21.514104488956, середній час (мс): 55.828619003296
 Кількість особин: 40, кількість поколінь: 40, шанс мутації: 0.18 || Середня к-сть відео: 22.093419732621, середній час (мс): 58.238792419434
 Кількість особин: 40, кількість поколінь: 40, шанс мутації: 0.2 || Середня к-сть відео: 20.874842111271, середній час (мс): 53.47056388855, c1
 Кількість особин: 40, кількість поколінь: 40, шанс мутації: 0.22 || Середня к-сть відео: 21.054614149523, середній час (мс): 52.738618850708
 Кількість особин: 40, кількість поколінь: 40, шанс мутації: 0.24 || Середня к-сть відео: 20.435368899866, середній час (мс): 53.16481590271,
 Кількість особин: 40, кількість поколінь: 40, шанс мутації: 0.26 || Середня к-сть відео: 21.214364493431, середній час (мс): 51.677417755127
 Кількість особин: 40, кількість поколінь: 40, шанс мутації: 0.28 || Середня к-сть відео: 21.593918047229, середній час (мс): 52.676820755005
 Кількість особин: 40, кількість поколінь: 40, шанс мутації: 0.3 || Середня к-сть відео: 20.97459245518, середній час (мс): 53.534364700317, c1

Найкращий результат:

Кількість особин: 30, кількість поколінь: 30, шанс мутації: 0.1 || Середня к-сть відео: 20.450916631525, середній час (мс): 25.881624221802,

Рисунок 3.8 – Логи знаходження найкращих комбінацій гіперпараметрів

Дане тестування було виконано кілька разів, для перевірки точності найкращих результатів. Результати даних тестувань представлено у таблиці 3.5. Як видно з таблиці 3.5, кількість курсів та кількість поколінь, в результаті тестів, мають оптимальний показник співвідношення ефективності генерації до часу виконання, при параметрах в межах від 29 до 31. Імовірність випадкових змін найкраще себе проявляє за умови, що показник знаходиться в межах від 0,1 до 0,18.

Після встановлення нових гіперпараметрів, було проведено повторне тестування роботи ефективності з різними обсягами даних, результат якого наведено в таблиці 3.6.

Таблиця 3.5 – Найкращі варіанти комбінацій гіперпараметрів

Номер тесту	Кількість курсів	Кількість поколінь	Ймовірність змін	Середня к-сть відеороликів	Середній час виконання (мс)
1	30	30	0.1	20.451	25.882
2	32	29	0.14	19.524	20.322
3	31	31	0.1	20.152	23.547
4	31	30	0.18	20.536	25.912
5	30	30	0.14	20.612	24.732

Таблиця 3.6 – Середні значення тестування при оптимальних параметрах

Кількість файлів	Середня кількість відеороликів	Середній час виконання
10000	8.1	~10.88 секунд
1000	7.8	~232.4 мілісекунд
100	7.3	~13.17 мілісекунд
10	5	~2.7 мілісекунд
1	1	~1.2 мілісекунд

Отже, можна зробити висновки, що оптимізація гіперпараметрів,

дозволила знайти комбінацію значень, при якій співвідношення точності результатів до часу виконання алгоритмів може задовільнити більшість потреб, які поставлені до розробленої інформаційної системи. Тестування ефективності з отриманими значеннями показує, що алгоритм значно пришвидшився, відносно початкових параметрів, при цьому отримані результати можуть задовільнити потреби користувачів, які генеруватимуть курси.

3.5 Покращення якості вмісту навчальних курсів

Для виконання процесу закриття компетенцій, було сформовано новий файл, який містить абстракції відеороликів різних категорій, згенеровані випадковим чином. Щоб тестування було точнішим, кількість абстракцій складається з десяти тисяч елементів. Перевірка відбувалась для двадцяти різних категорій, які формували віртуальну базу даних, при цьому деякі відеоролики відносились до кількох категорій. Час відеороликів випадковим чином формувався у межах від 10 до 30 хвилин, при цьому обмеження алгоритму встановлювалось різне по кілька разів. Для отримання оптимальних результатів, використано оптимізовані значення гіперпараметрів.

Сформована база даних, мала список атрибутів відеоуроків, які відносяться до двох, або трьох категорій. Таким чином, була сформована таблиця, яка показує належність кожного створеного елемента до різних наявних компетенцій та час кожного відеоролика (таблиця 3.7).

Для початку, була перевірена робота алгоритму для відносно невеликого вказаного часу (додатки Д.1 – Д.3). Тематика сформованого курсу обиралась випадковим чином. Після виконання програмного коду, підраховувалась кількість відеороликів, які підходять під обрану тематику. Також, підраховувалось, скільки часу, з доступного, займає сформований курс.

Результати показали, що при перевірці будь-якої компетенції, сформовані курси містять лише відеоролики, одна з компетенцій якої завжди відповідає обраній. При цьому, покриття заданого обмеження часу дуже близьке до очікуваних результатів, що вказує на коректну максимізацію результатів, навіть

при використанні великого масиву даних. Кількість відібраних відеороликів у курсі, лише на кілька елементів різнилось від максимально можливої кількості.

Таблиця 3.7 – Множина сформованих відеороликів

Номер відеоролику	Тривалість відеоролику	K1	K2	K3	K4	...	K20
1	19:11		X		X	.	
2	10:32	X				.	
3	28:59		X			.	X
4	12:52	X		X		.	
...
10000	17:52		X			.	X

Наступне тестування компетенцій проводилось для вдвічі більшого проміжку часу (додатки Д.4 – Д.6). Результати показали, що збільшення обмеження в часі ніяк не вплинуло на коректність відбору тематики відеороликів, що означає, що кожна компетентність виконується. Але, при цьому стає помітним, що при використанні одного і того ж набору відеороликів, кількість матеріалу у курсі стає ще меншою, від максимально можливої. Проте тривалість сформованих курсів залишається близькою до заданих обмежень.

Збільшивши обмеження на тривалість ще у два рази (додатки Д.7 – Д.9), привело до очікуваних регресійних результатів роботи. По при те, що усі компетенції залишаються коректно закритими, відбір відео показав, що їх кількість стала ще трохи меншою від максимально можливої. При цьому, співвідношення кількості матеріалу в курсах, залишається високим, що свідчить про малу похибку, під час обрахування результатів. Як і очікувалось, покриття доступного часу, займає майже увесь вільний наданий простір.

Отже, отримані результати свідчать про те, що поставлені критерії до тематики відеороликів та тривалості курсів виконуються. З збільшенням

параметру обмежень в часі, які впливають на розмір курсів, які формуються, результати стають все менш подібними до ідеальних, хоча похибка має відносно малий ефект. Проте, алгоритм показав власну можливість формування курсів з великої вибірки матеріалу, які максимально покривають задані обмеження, оптимально обираючи близьку кількість відеороликів. Робота з такою кількістю даних та параметрів, займе у людини багато часу, при цьому результат ймовірно буде далекий від досконалого, або не буде досягнутий, що вказує на ефективність розробленого алгоритму.

Найважливішим етапом у дослідженні було тестування, яке було націлене на закриття компетенцій якості створених курсів, а також ефективності формування курсів. Якість сформованих курсів базувалась на співвідношенні очікуваних результатів до отриманих. Отриманні значення були переведені до відсотків, на основі яких пораховані середні відсотки збігів для протестованих категорій (таблиця 3.8). Отримані результати свідчать про те, що алгоритм повністю закриває встановлені компетенції.

Дану таблицю можна представити у вигляді діаграми, яка відображає залежність результатів перевірки компетенцій від обмеження часу (рисунок 3.9)

Таблиця 3.8 – Відсотковий показник якості формування курсів

Обмеження в часі	Заповненість тривалості курсів	Відсоток охоплених компетенцій
180	99.1%	100%
360	99.9%	100%
720	99.4%	100%
Сер. значення	99.46%	100%

Тестування з використанням тестового набору навчальних відеоуроків виявило, що заповнення наданого простору часу становить майже 100% й у прикладах становило 99, 1-99,9% за умови 100%-го охоплення компетенцій.

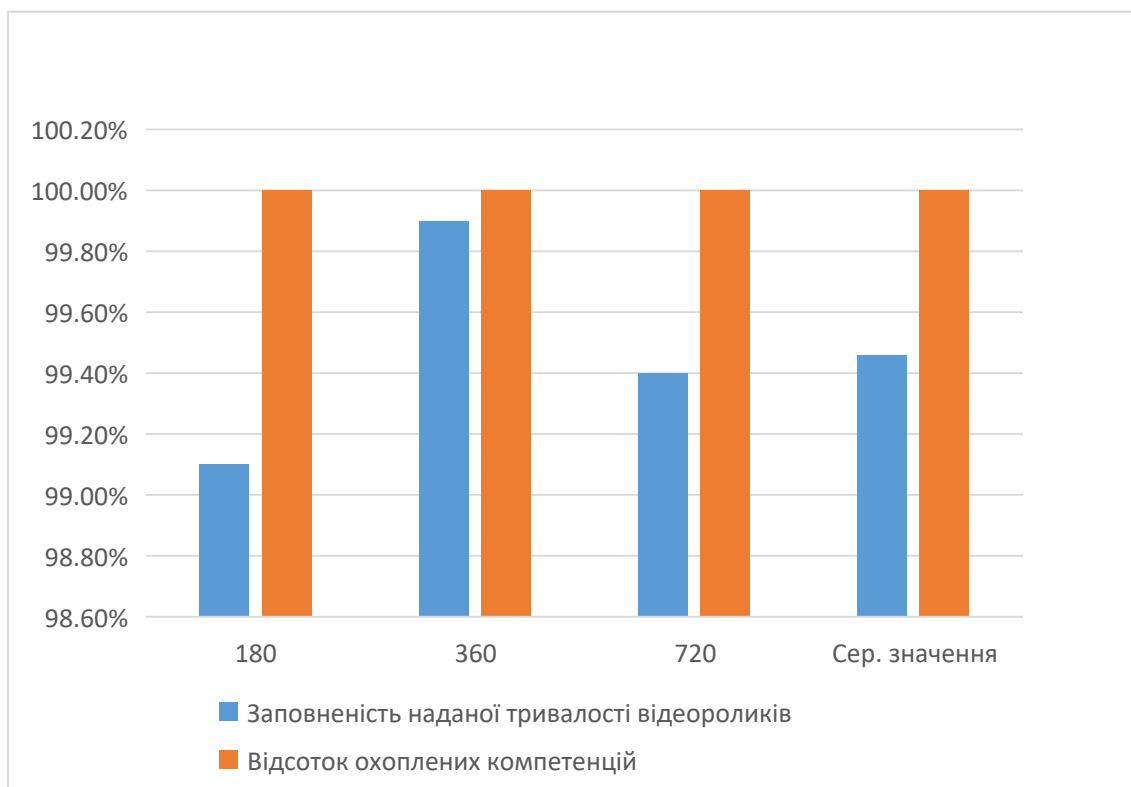


Рисунок 3.9 – Використання виділеного часу та охоплення компетенцій

3.6 Висновки до розділу 3

Виконано програмну реалізацію методу покращення вмісту навчальних курсів за генетичним алгоритмом. Реалізація виконана як вебсистема із визначенням його структури та функціонального призначення його складових.

Програмна реалізація методу виконана із використанням технологій PHP, JS, HTML, CSS та стилістичної бібліотеки Bootstrap. Інтеграція та взаємодія компонентів системи забезпечена через використання платформи XAMPP та системи керування базами даних MySQL.

Структура програмної реалізації методу складається з взаємопов'язаних компонентів, розділених на дві основні підгрупи: компоненти викладачів та компоненти слухачів, розподілених по вебсторінкам. До другого було віднесено механізм генерації курсів, який поділений на дві послідовні підсистеми: динамічної і статичної.

Для реалізації процесу генерації курсів, було застосовано особливості

мови РНР, що дозволило звертатись до масивів, як до динамічних списків, що в свою чергу, надало можливість зменшити обсяг коду. Реалізація передбачала створення двох класів, які можуть працювати з різними даними, якщо вони перед передачею формуються до відповідного вигляду подачі.

Тестування методу у контексті розробленої інформаційної системи, показало, що усі функції працюють коректно, а дані що передаються до генетичного алгоритму, валідуються, що свідчить про стійкість системи. Але, тестування ефективності показало, що система довго опрацьовує велику кількість даних, через що довелося проводити оптимізацію гіперпараметрів для збільшення швидкості алгоритму.

В результаті оптимізації гіперпараметрів, було визначено, що алгоритм чутливий до різних їх комбінацій, при цьому початкові параметри працювали повільно. Тому, після перебору різних комбінацій, для збільшення швидкодії, довелося знехтувати точністю кінцевих результатів.

Результати оцінювання покращення якості вмісту навчальних курсів показали, що результати генерації відповідають поставленим критеріям. Таким чином, усі відеоролики у сформованих курсах завжди відповідають обраній тематиці. При цьому заповнення доступної тривалості, якщо це можливо, завжди близьке до встановленого обмеження, ефективно використовуючи час.

Отримані результати засвідчують що даний метод коректно опрацьовує інформацію, адаптуючи кожне покоління до очікуваних результатів, швидко та якісно формуючи курси для користувачів за персоналізованими параметрами. Швидкість збігу корелює з розмірами гіперпараметрів і, відповідно, швидкістю виконання. В результаті, з кожним поколінням розподіл доступних курсів рухається до більшої пристосованості.

Тестування з використанням тестового набору навчальних відеоуроків виявило, що заповнення наданого простору часу становить майже 100% й у тестових прикладах становило 99, 1-99,9% за умови 100%-го охоплення бажаних компетенцій.

Загальні висновки

В результаті виконання кваліфікаційної роботи бакалавра, було розроблено та реалізовано метод покращення освітніх вебресурсів на основі генетичного алгоритму.

При виконанні кваліфікаційної роботи бакалавра було вирішено такі задачі дослідження:

- проведено аналіз методів формування та покращення вмісту навчальних дисциплін;
- проведено аналіз інтелектуальних методів для визначення найкращого вмісту навчальних курсів;
- реалізовано метод покращення вмісту навчальних курсів за генетичним алгоритмом
- виконано програмну реалізацію методу покращення вмісту навчальних курсів з адаптацією генетичного алгоритму;
- проведено експериментальне тестування застосунку;
- виконано оцінювання покращення вмісту навчальних курсів.

Під час етапу проектування визначено ключові поняття та розроблено схему алгоритму для генерації курсів з використанням процесів нормалізації. Імплементация генетичного алгоритму показала ефективність в адаптації контенту до різних критеріїв.

Використання FFmpeg для управління потоковими відео та нормалізація бази даних до третьої нормальної форми допомогли уникнути аномалій при обробці інформації. Для системи було передбачено ручне та автоматичне введення даних користувачами для генерацію курсів для коректного тестування.

Створено функціональну структуру з ролями користувачів, слухачів та викладачів, яка забезпечує працездатність і обробку помилок. В архітектурі програмної реалізації методу визначено 7 підсистем для оптимального функціонування клієнт-серверної системи.

Встановлено критерії оцінки покращення курсів, включаючи кількість відео, що закривають необхідні компетентності та ефективно використовують

наданий час.

Далі було створено вебсистему для покращення змісту навчальних курсів за допомогою генетичного алгоритму. Реалізація виконана з використанням PHP, JS, HTML, CSS та Bootstrap, а компоненти системи взаємодіяли через XAMPP.

Система включала дві підгрупи компонентів: для викладачів і для слухачів, з механізмом генерації курсів, поділеним на динамічну і статичну підсистеми. Використання PHP дозволило оптимізувати код і створити два класи для роботи з різними даними.

Тестування показало, що всі функції працюють коректно, а дані валідовані. Однак оптимізація показала перспективи підвищення якості та швидкості розробленого методу. Результати генерації курсів відповідали заданим критеріям, ефективно використовуючи час.

Система продемонструвала коректну обробку інформації та швидке формування персоналізованих курсів. Тестування показало 99,1 - 99,9% заповнення доступного часу при 100% охопленні бажаних компетенцій.

Результати дослідження показують, що розроблений метод ефективний для створення персоналізованих курсів і може використовуватись в різних освітніх системах. Програмна реалізація методу може ефективно працювати з великими об'ємами даних, що робить його корисним для вдосконалення освітніх вебресурсів.

Перелік посилань

1. Crutcher D. P. Essential Computer Science: A Programmer's Guide to Foundational Concepts / D. P. Crutcher, K. S. Singh, P. Tiegs. – New York City : Apress, 2021. – 316 p.
2. Cowan L. Build Your Own Website for Beginners / L. Cowan. – London : Usborne Publishing, 2020. – 96 p.
3. Computer Science and Design Vs Computer Engineering: What's the Difference? [Електронний ресурс] // DPU : [сайт]. – Режим доступу: <https://dypsst.dpu.edu.in/blogs/computer-science-and-design-vs-computer-engineering-what-s-the-difference>. – Назва з екрану
4. Dangerfield B. System Dynamics: Theory and Applications / B. Dangerfield. – New York City : Springer US, 2020. – 532 p.
5. Aggarwal S. Machine Learning Algorithms, Perspectives, and Real-World Application: Empirical Evidence from United States Trade Data / S. Aggarwal // International Journal of Science and Research (IJSR). – 2023. – № 12. – P. 292-313.
6. Wharton J. The Website Handbook: Plan and Create a Website That Attracts Leads and Makes Sales / J. Wharton, C. Wharton. – Southampton : 2nd Floor Designs, 2022. – 112 p.
7. Doctorow C. The Internet Con: How to Seize the Means of Computation / C. Doctorow. – New York City : Verso, 2023. – 138 p.
8. 12 Popular Types of Websites You Can Create [Електронний ресурс] // HostGator : [сайт]. – Режим доступу: <https://www.hostgator.com/blog/popular-types-websites-create>. – Назва з екрану.
9. Whatman S. Researching Practices Across and Within Diverse Educational Sites / S. Whatman [and etc.]. – Leeds : Emerald Publishing, 2023. – 200 p.
10. Impact of the COVID-19 pandemic on online learning in higher education: a bibliometric analysis [Електронний ресурс] // Frontiers : [сайт]. – Режим доступу: <https://www.frontiersin.org/articles/10.3389/feduc.2023.1225834/full>. – Назва з екрану.

11. 27 common types of websites (with templates to get you started) [Электронный ресурс] // WIXBlog : [сайт]. – Режим доступа: <https://www.wix.com/blog/types-of-websites>. – Назва з екрану.

12. Martin F. R. Digital Platform Regulation: Global Perspectives on Internet Governance / F. R. Martin, T. Flew. – New York City : Springer, 2022. – 319 p.

13. What are the Biggest Challenges of Online Education Today? [Электронный ресурс] // Hurixdigital : [сайт]. – Режим доступа: <https://www.hurix.com/what-are-the-biggest-challenges-facing-online-education-today>.

14. Maximizing student success with intelligent course recommendations. – Назва з екрану [Электронный ресурс] // Ellucian : [сайт]. – Режим доступа: <https://www.ellucian.com/blog/maximizing-student-success-intelligent-course-recommendations>. – Назва з екрану.

15. The importance of discussion forums for online learning [Электронный ресурс] // Educational Design And Engagement : [сайт]. – Режим доступа: <https://blogs.ed.ac.uk/ede/2021/03/03/the-importance-of-discussion-forums-for-online-learning>. – Назва з екрану.

16. Weina G. E-Learning, E-Education, and Online Training / G. Weina, W. Fu. – New York City : Springer, 2023. – 781 p.

17. Watts S. Online Teaching Survival Guide: The Best Teaching Strategies and Tools for Your Online Classroom / S. Watts. – Hampshire : Wryting Ltd, 2020. – 135 p.

18. Lakhmi C. J. Smart Education and E-Learning / C. J. Lakhmi, J. R. Howlett, V. L. Uskov. – New York City : Springer, 2021. – 506 p.

19. The Benefits of E-books in Online Education [Электронный ресурс] // Dan Institute : [сайт]. – Режим доступа: <https://daninstitute.com/blog/the-benefits-of-ebooks-in-online-education/>. – Назва з екрану.

20. Advantages and disadvantages of webinars [Электронный ресурс] // TellOnline : [сайт]. – Режим доступа: <https://www.tellonline.org/articles/advantages-and-disadvantages-of-webinars/>. – Назва з екрану.

21. Kahn J. Principles and Methods in Commercial Education: A Textbook for Teachers, Students, and Business Men / J. Kahn. – Riga : Creative Media Partners,

2023. – 458 p.

22. Whiteside A. L. Social Presence in Online Learning: Multiple Perspectives on Practice and Research / A. L. Whiteside, A. Dikkers, K. Swan. – Oxfordshire : Taylor & Francis, 2023. – 242 p.

23. Milman N. B. Designing and Teaching Online Courses During Uncertain Times: A Special Issue of Distance Learning Ends and Means Columns, Distance Learning / N. B. Milman. – Charlotte : Information Age Publishing, 2020. – 117 p.

24. What are Course Creators? How to Become One in 2024 [Электронный ресурс] // DISCO : [сайт]. – Режим доступа: <https://www.disco.co/blog/what-are-course-creators>. – Назва з екрану.

25. Recommended process for course creation (and beyond) [Электронный ресурс] // UdeMy : [сайт]. – Режим доступа: <https://teach.udemy.com/course-creation/recommended-course-creation-process/>. – Назва з екрану.

26. Nash S. S. Moodle 4 E-Learning Course Development - Fifth Edition: The definitive guide to creating great courses in Moodle 4.0 using instructional design principles / S. S. Nash. – Birmingham : Packt Publishing, 2022. – 436 p.

27. What Is a Site Administrator? (With Job Requirements and FAQs) [Электронный ресурс] // Indeed : [сайт]. – Режим доступа: <https://ca.indeed.com/career-advice/finding-a-job/what-is-a-site-administrator>. – Назва з екрану.

28. Chong H. N. Create E-learning Courses People Want: How to Create Engaging E-learning Courses the ADDIE Way / H. N. Chong. – London : Barnes & Noble Press, 2021. – 154 p.

29. Slade T. The ELearning Designer's Handbook: A Practical Guide to the ELearning Development Process for New ELearning Designers / T. Slade. – Chicago : Independently Published, 2020. – 234 p.

30. The Power Of Feedback In eLearning [Электронный ресурс] // eLearning Industry : [сайт]. – Режим доступа: <https://elearningindustry.com/power-of-feedback-in-elearning>. – Назва з екрану.

31. Seo K. The impact of artificial intelligence on learner–instructor interaction in online learning / K. Seo, J. Tang, I. Roll, S. Fels // International Journal

of Educational Technology in Higher Education : [сайт]. – 2021. – № 18. – 23 p. – Режим доступа: <https://link.springer.com/article/10.1186/s41239-021-00292-9>.

32. The Rise Of AI-Enhanced Learning: Education For The Digital Age [Электронный ресурс] // Forbes : [сайт]. – Режим доступа: <https://www.forbes.com/sites/forbesbusinesscouncil/2024/02/13/the-rise-of-ai-enhanced-learning-education-for-the-digital-age/?sh=71c1add82b9c>. – Назва з екрану.

33. The Role Of Artificial Intelligence In Learning [Электронный ресурс] // eLearning Industry : [сайт]. – Режим доступа: <https://elearningindustry.com/artificial-intelligence-in-learning-role>. – Назва з екрану.

34. The Pros and Cons of AI in Education and How it Will Impact Teachers in 2024 [Электронный ресурс] // ClassPoint : [сайт]. – Режим доступа: <https://www.classpoint.io/blog/the-pros-and-cons-of-ai-in-education>. – Назва з екрану.

35. AI in Learning and Development: 4 Practical Use Cases That Will Set You Apart [Электронный ресурс] // 360 Learning : [сайт]. – Режим доступа: <https://360learning.com/blog/ai-in-learning-and-development-uk/>. – Назва з екрану.

36. Heung I. Artificial intelligence (AI) learning tools in K-12 education: A scoping review / I. Heung, J. Su // Journal of Computers in Education : [сайт]. – 2024. – № 12. – 39 с. – Режим доступа: <https://link.springer.com/article/10.1007/s40692-023-00304-9>.

37. How AI in education can enhance the learning process [Электронный ресурс] // FastCompany : [сайт]. – Режим доступа: <https://www.fastcompany.com/90946322/how-ai-in-education-can-enhance-the-learning-process>. – Назва з екрану.

38. Khan R. Artificial Intelligence and Natural Algorithms / R. Khan. – Sharjah : Bentham Science Publishers, 2022. – 383 p.

39. The Types Of Artificial Intelligence And Their Role In E-Learning Platforms [Электронный ресурс] // Thkee : [сайт]. – Режим доступа: <https://thkee.com/en/the-types-of-artificial-intelligence-and-their-role-in-e-learning-platforms/>. – Назва з екрану.

40. Elmeawad S. The Use of AI in Libraries to Predict the Users' Needs in Different Communities / S. Elmeawad. – Eldon : Eldonusa Incorporated, 2023.– 410 p.

41. Lee D. D. AI Educator: Upgrading the Education System with Artificial Intelligence / D. D. Lee. – New York City : Amazon Digital Services LLC - Kdp, 2023. – 102 p.

42. Tao J. DNA Computing Based Genetic Algorithm: Applications in Industrial Process Modeling and Control / J. Tao, R. Zhang, Y. Zhu. – New York City : Springer, 2020. – 274 p.

43. Chambers L. D. The Practical Handbook of Genetic Algorithms: New Frontiers, Volume II / L. D. Chambers. – Boca Raton : CRC Press, 2019. – 464 p.

44. SkillShare [Электронный ресурс] // SkillShare. – Режим доступа: <https://www.skillshare.com/>. – Назва з екрану.

45. Udacity [Электронный ресурс] // Udacity. – Режим доступа: <https://www.udacity.com/>. – Назва з екрану.

46. LinkedIn Learning [Электронный ресурс] // LinkedIn Learning. – Режим доступа: <https://www.linkedin.com/learning/>. – Назва з екрану.

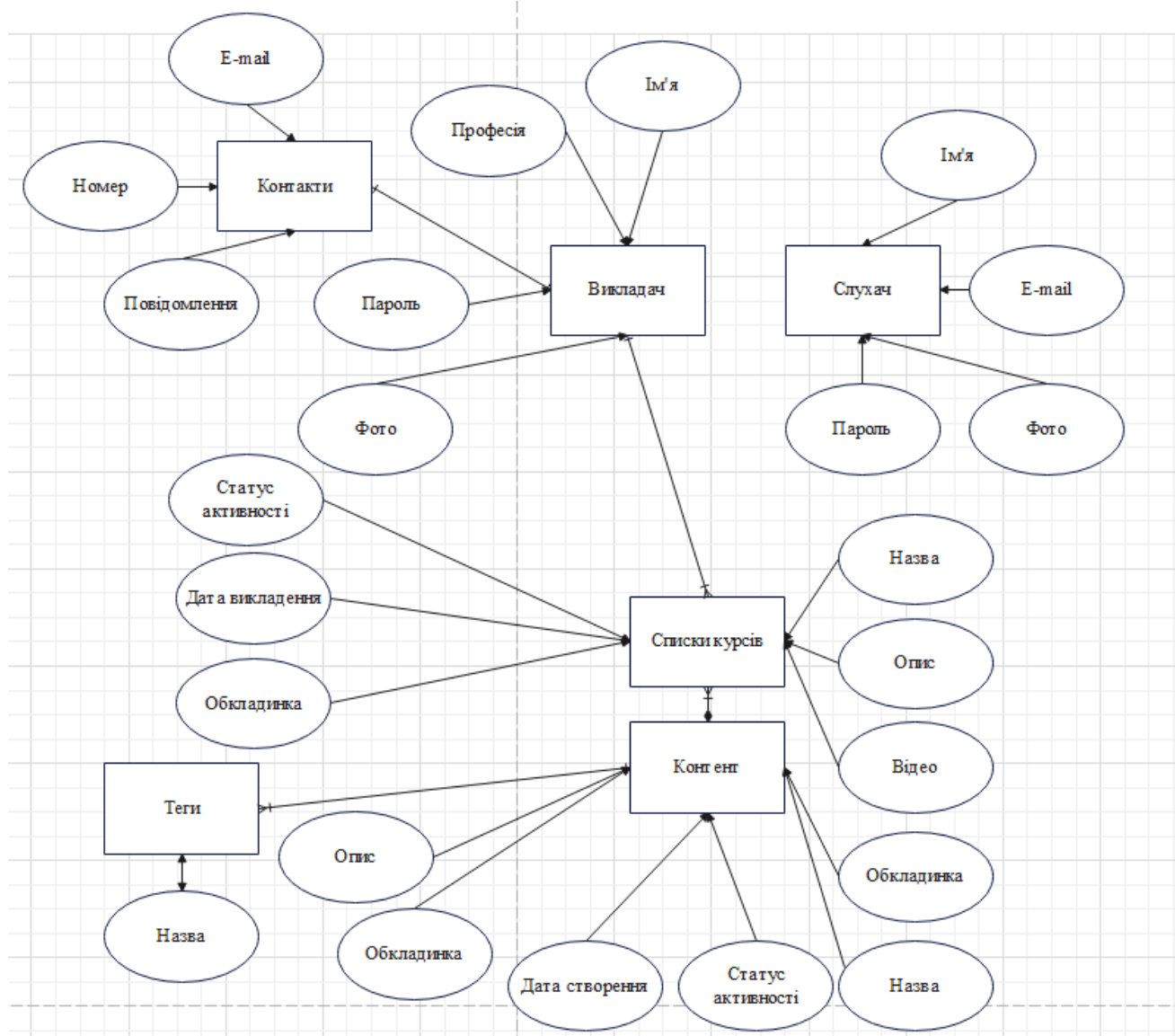
47. Esteban A. Helping university students to choose elective courses by using a hybrid multi-criteria recommendation system with genetic optimization / A. Esteban, A. Zafra, C. Romero // Knowledge-Based Systems : [сайт]. – 2020. – № 194(4). – 43 с. – Режим доступа: <https://www.sciencedirect.com/science/article/abs/pii/S0950705119306306?via%3Dihub>.

ДОДАТКИ

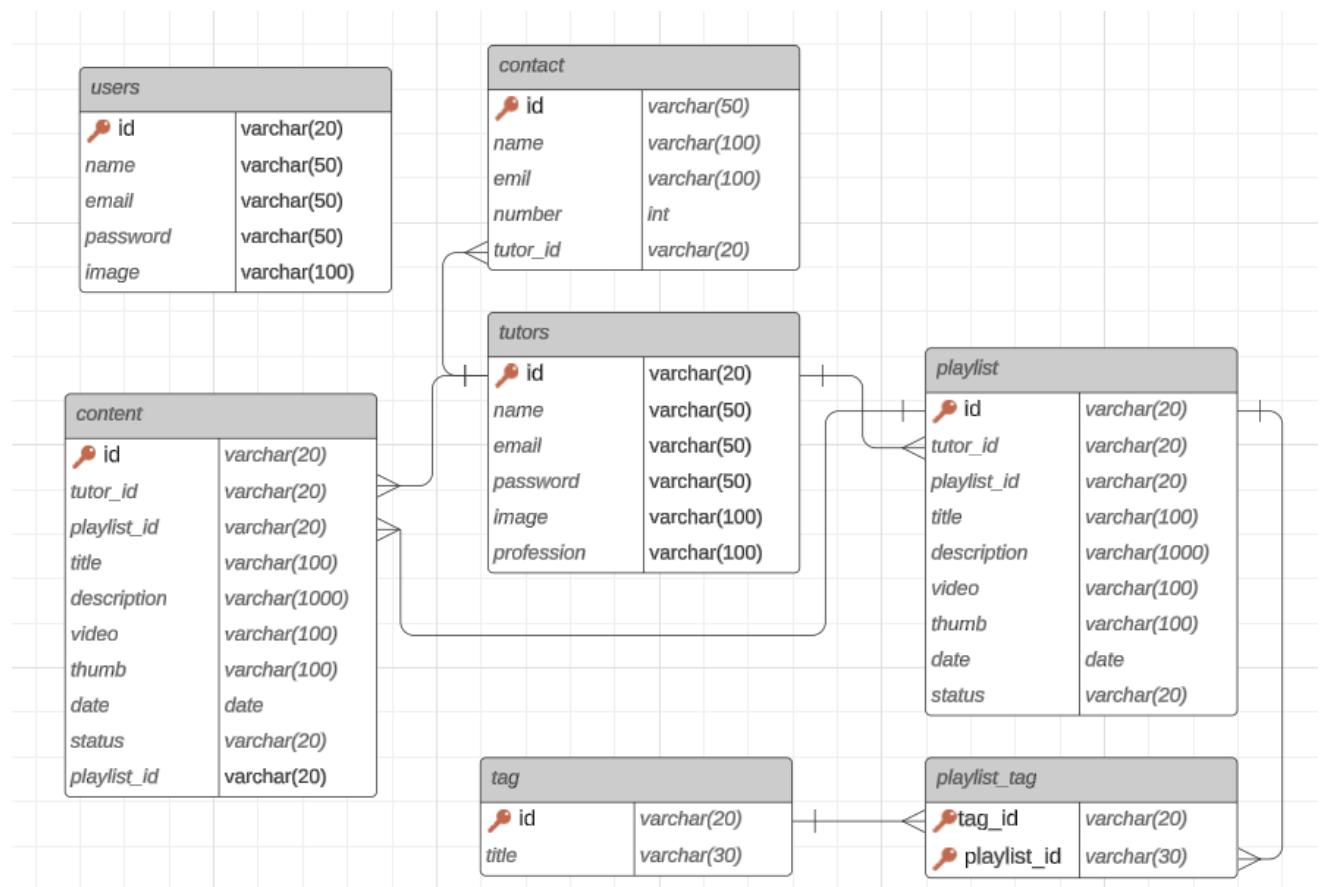
Додаток А

Моделі бази даних

Додаток А.1 – ER-діаграма даних навчального вебресурсу



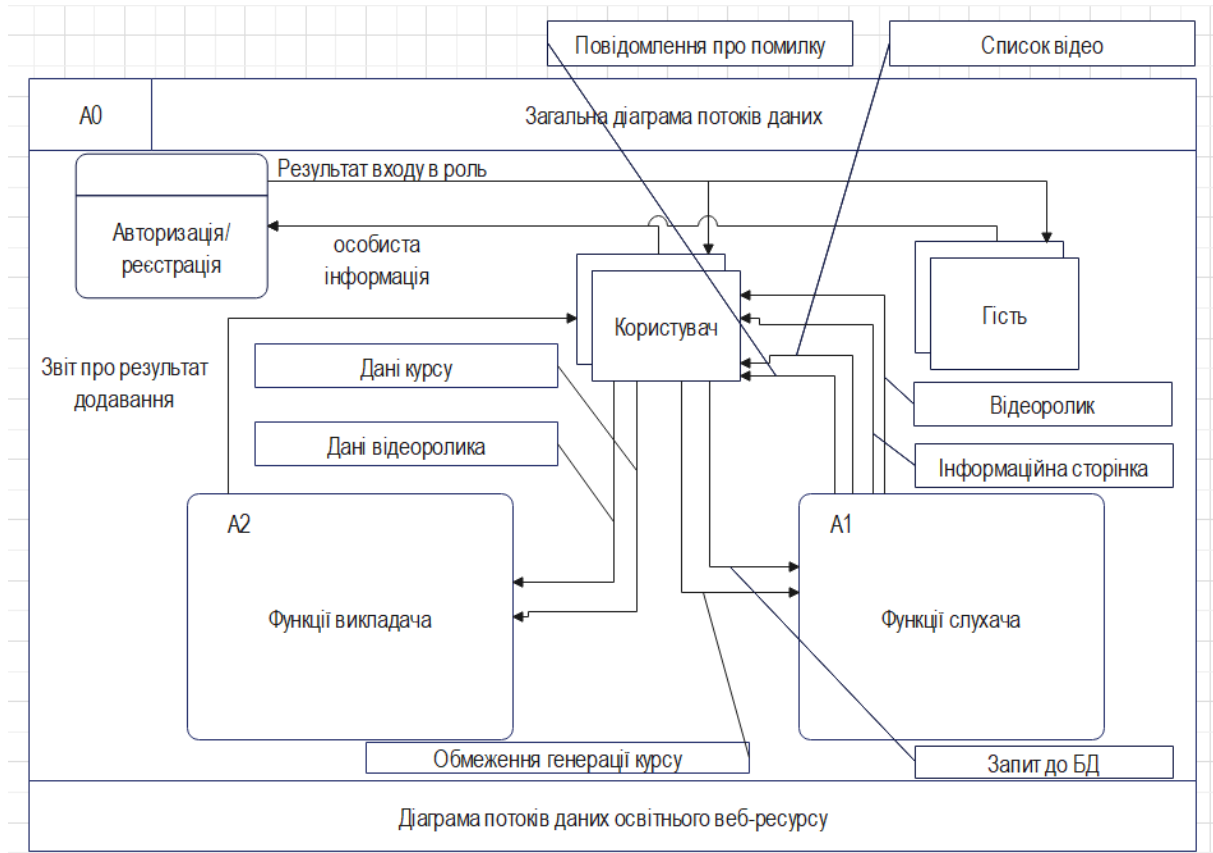
Додаток А.2 – Даталогічна модель бази даних для навчального вебресурсу



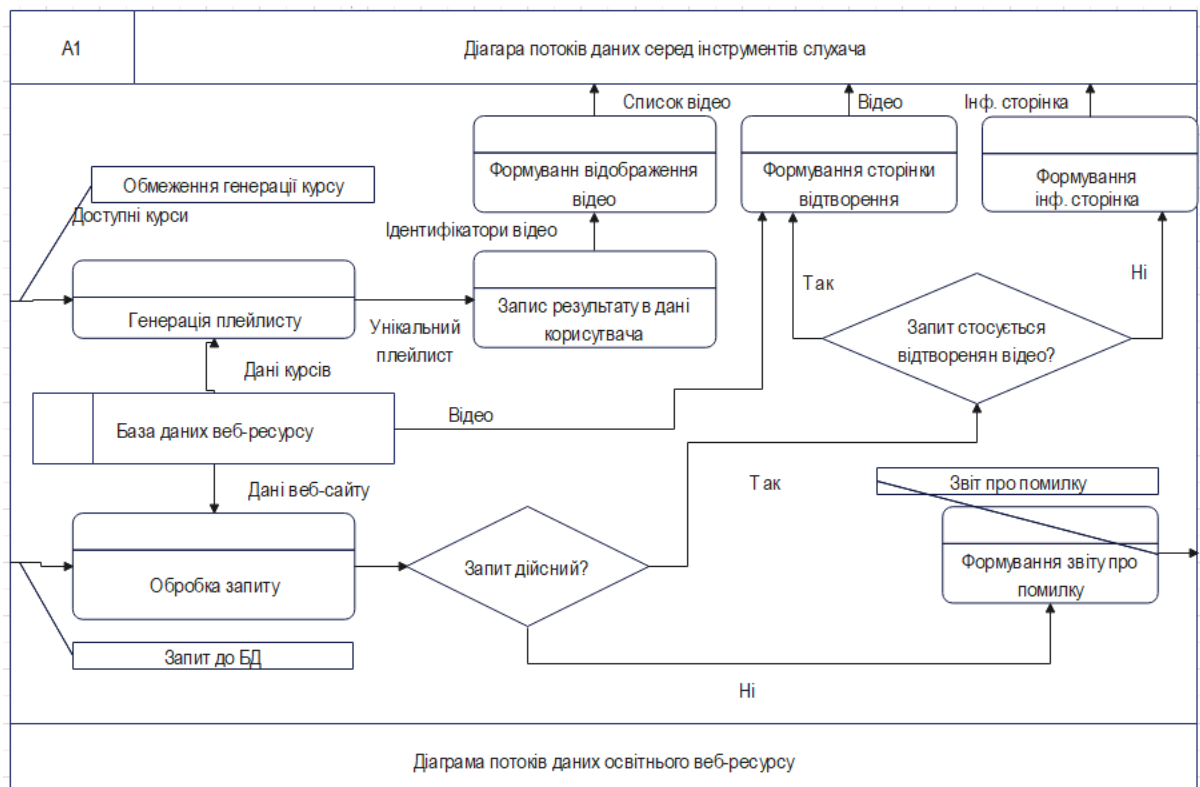
Додаток Б

Декомпонована діаграми потоків даних

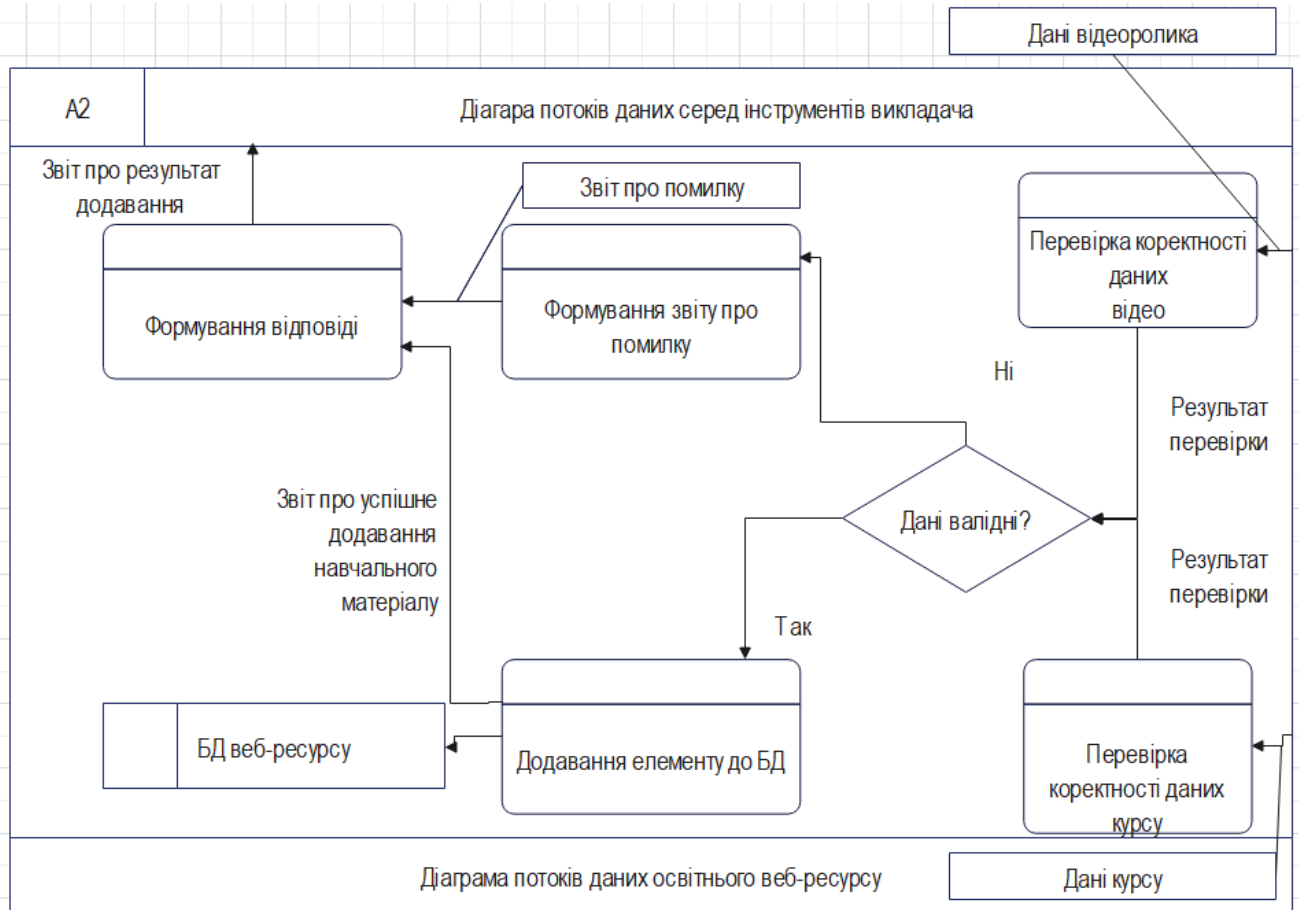
Додаток Б.1 – Загальна діаграма потоків даних



Додаток Б.2 – Діаграма потоків даних у функціях слухача



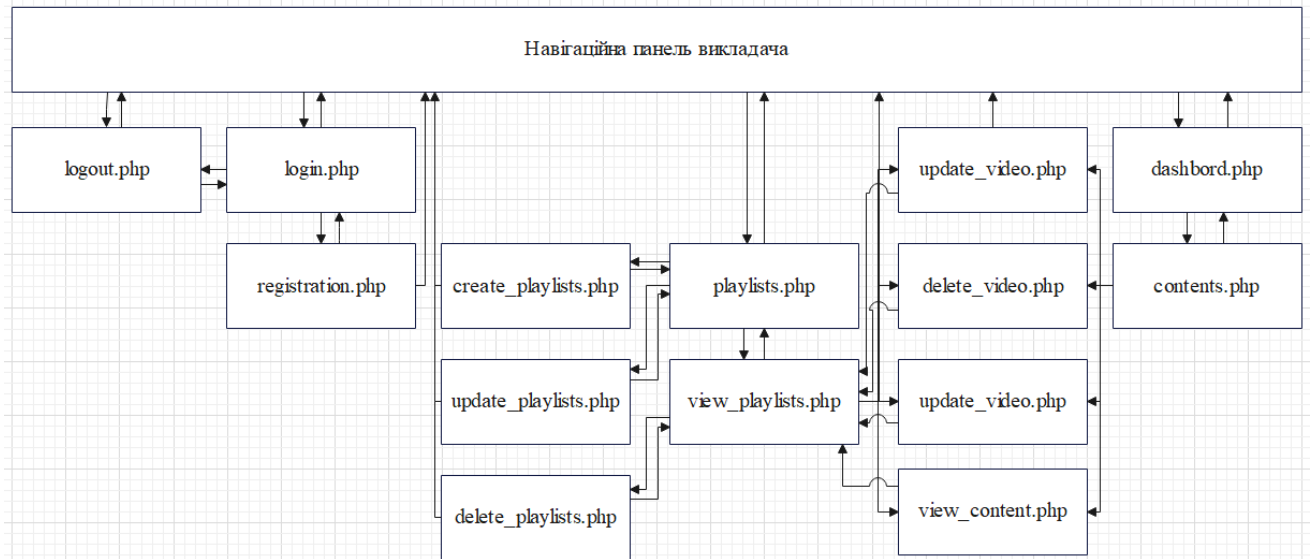
Додаток Б.3 – Діаграма потоків даних у функціях викладача



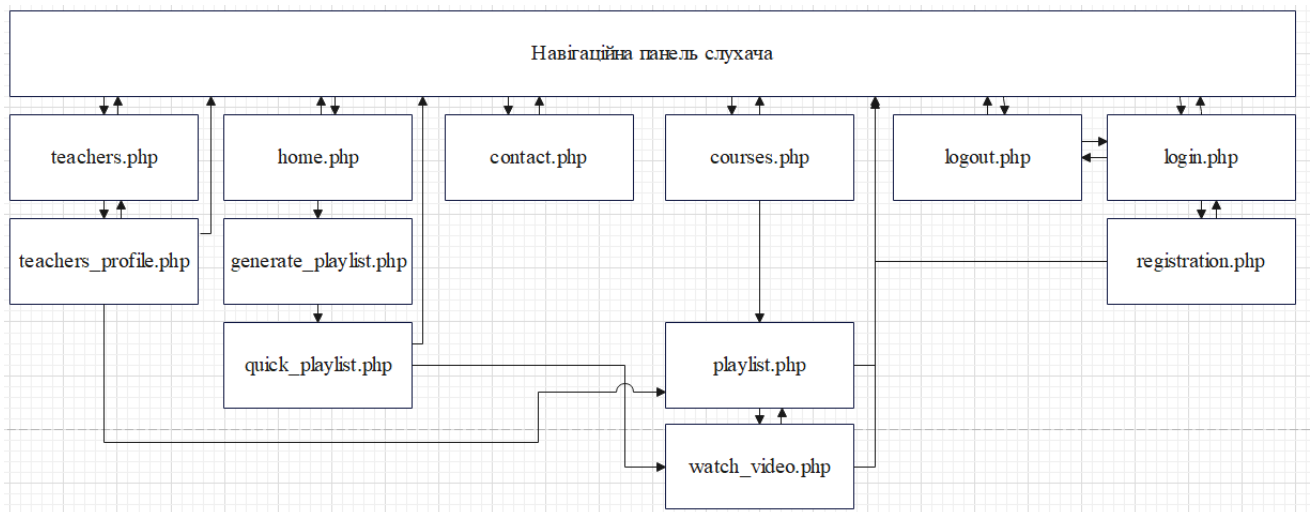
Додаток В

Карта сайту переходу між сторінками користувачів

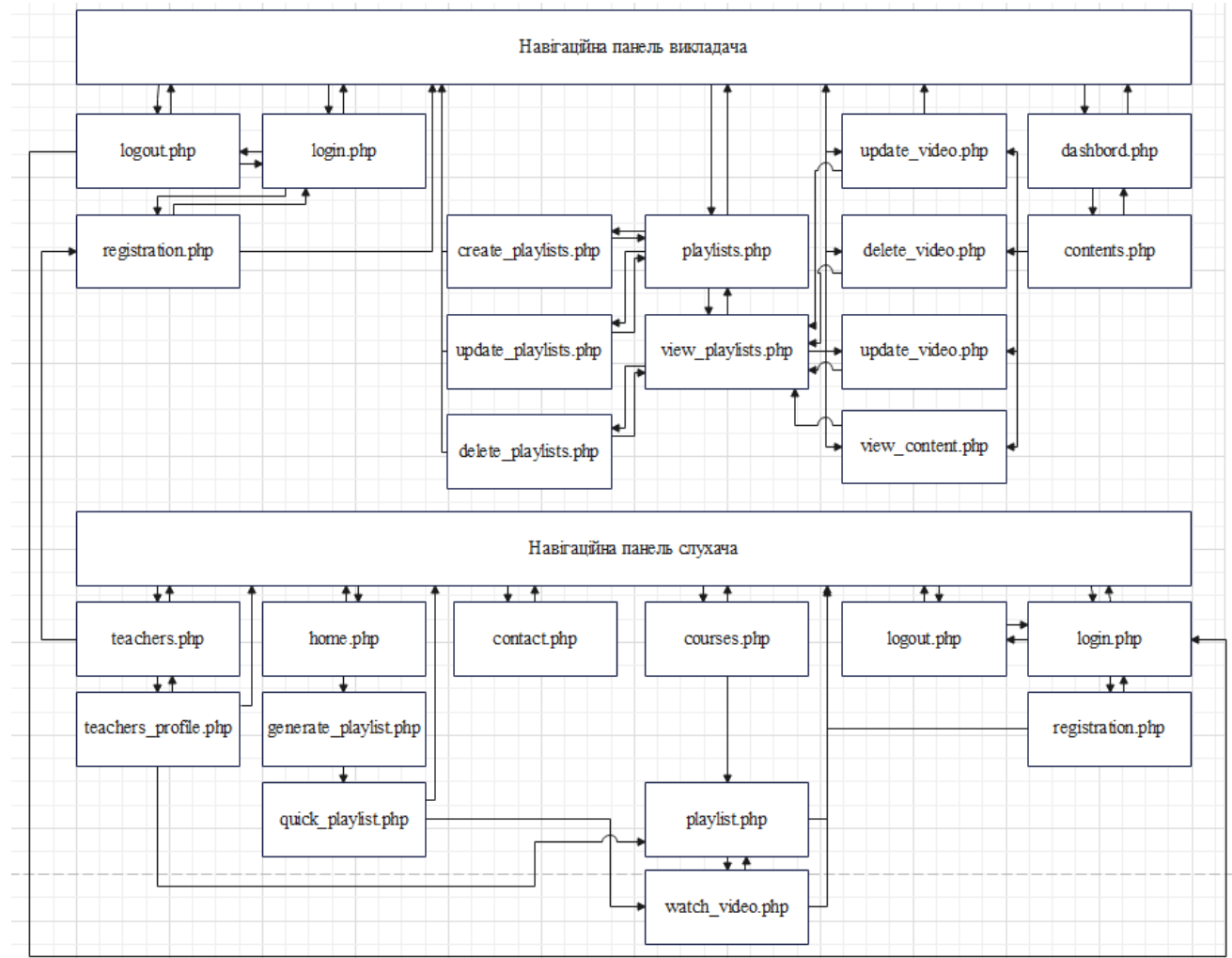
Додаток В.1 – Карта сайту для функцій викладача



Додаток В.2 – Карта сайту для функцій слухача



Додаток В.3 – Загальна карта сторінок застосунку



Додаток Г

Зображення сторінок освітньої вебплатформи для навчальних курсів

Рисунок Г.1 – Головна сторінка

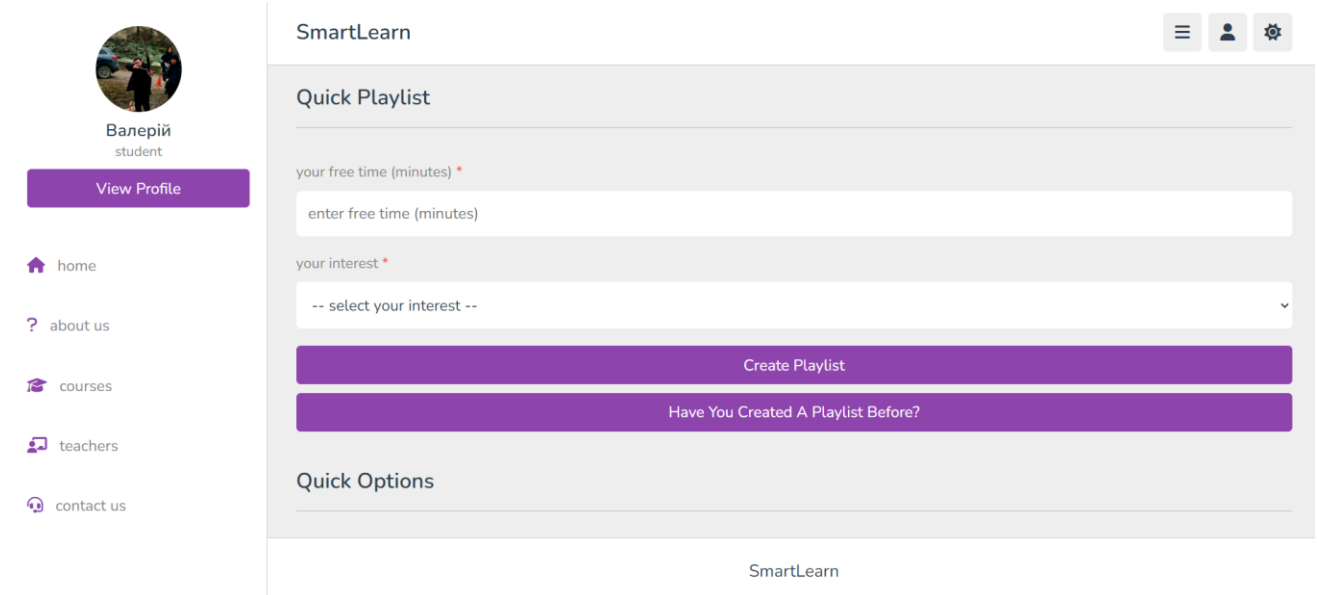


Рисунок Г.2 – Сторінка про платформу

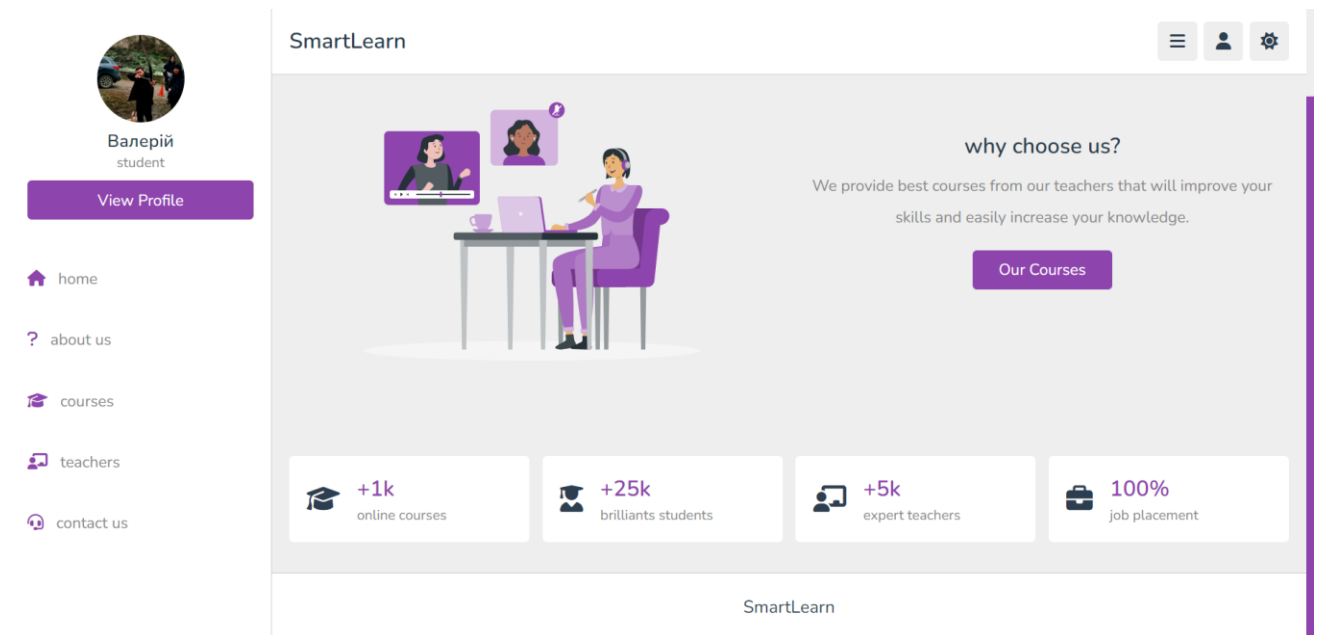


Рисунок Г.3 – Сторінка перегляду навчальних курсів

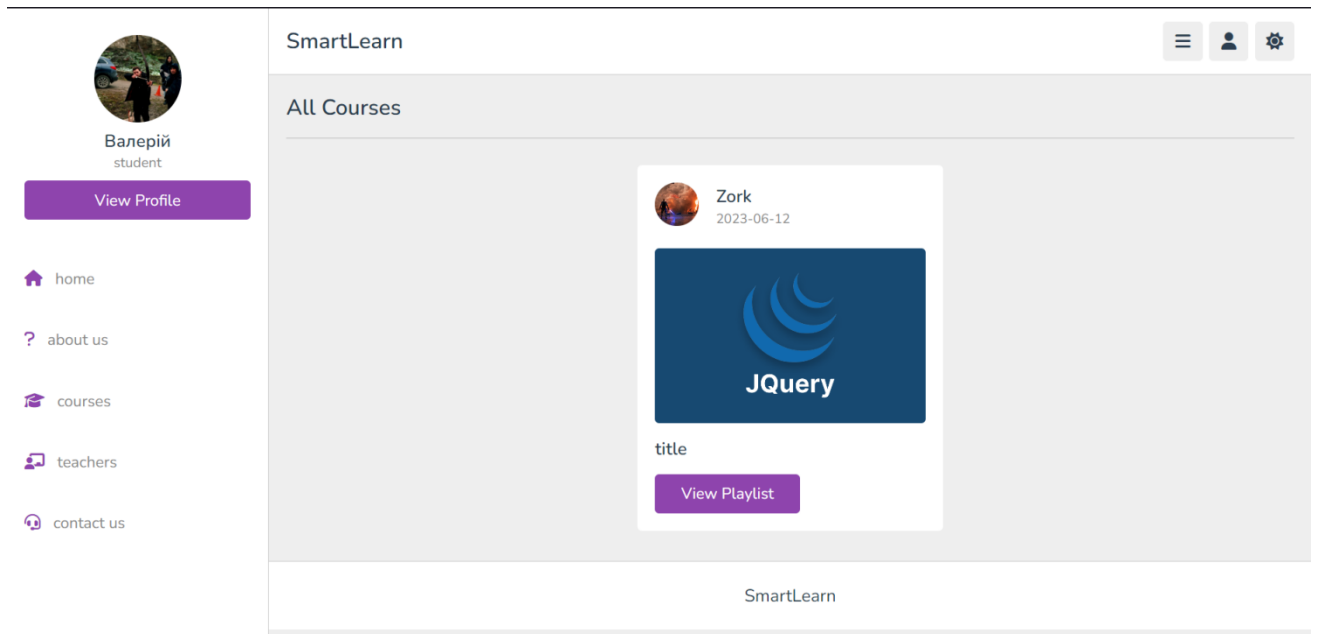


Рисунок Г.4 – Сторінка про вчителів

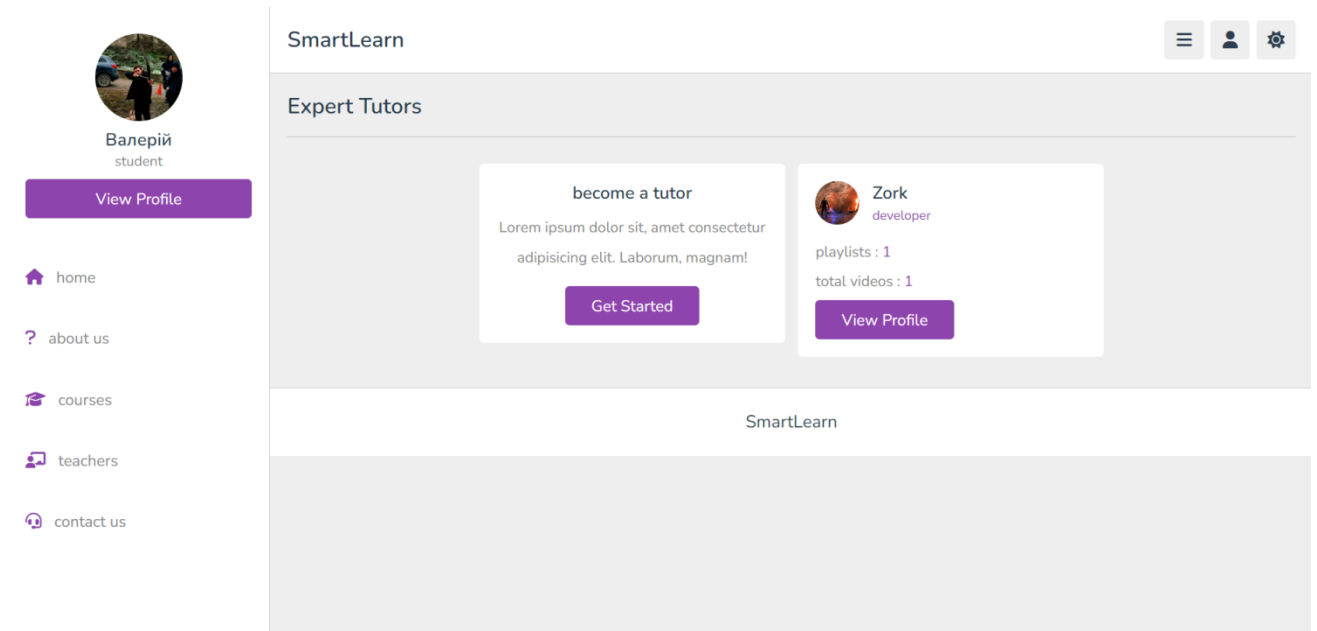


Рисунок Г.4 – Сторінка зворотнього зв'язку

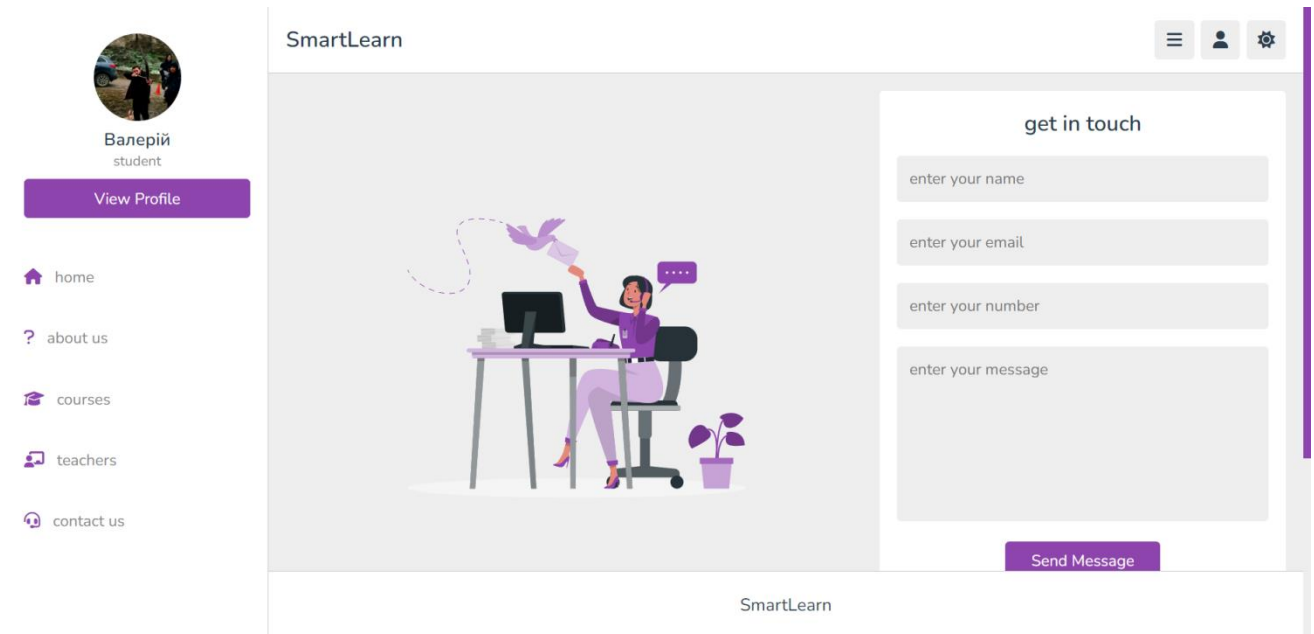
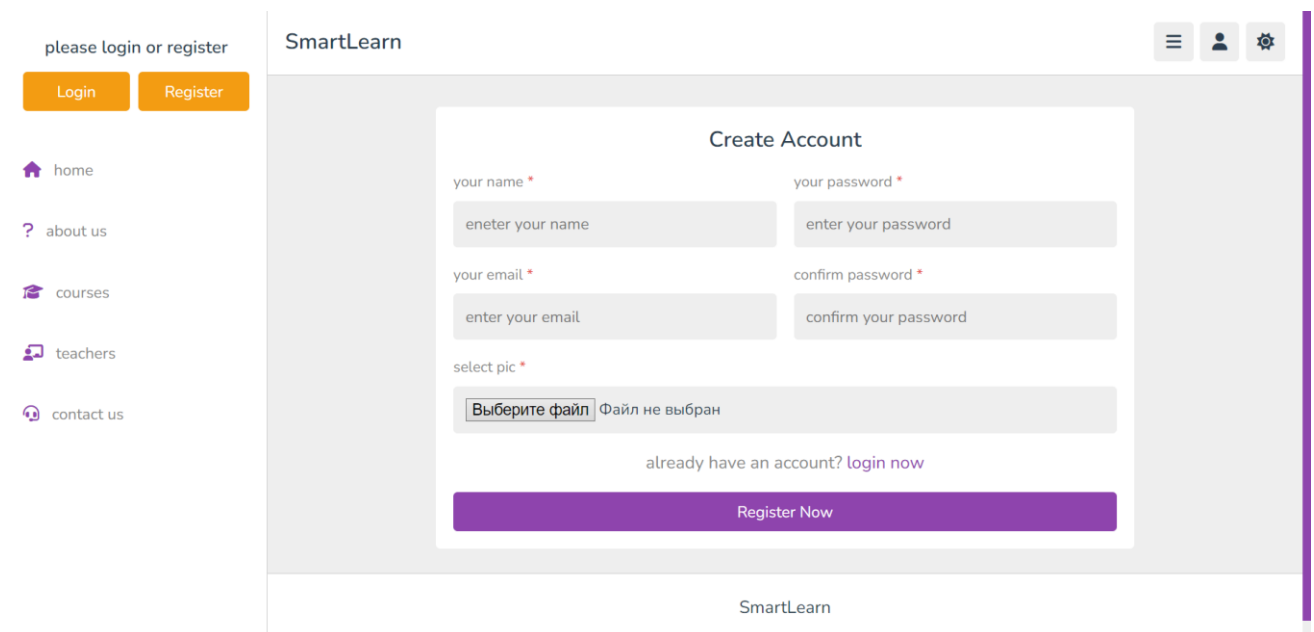


Рисунок Г.5 – Сторінка реєстрації в якості викладача



Додаток Д
Результати закриття компетенцій

Таблиця Д.1 – Звіт тестування №1 для формування 180-хвилинних курсів

Тестування для компетенції №4							
Номер відеоролику	Тривалість	K1	K2	K3	K4	...	K20
9	11:09	X		X		.	
278	13:07					.	
596	10:32		X		X	.	
...
9689	10:28	X				.	X
Сумарний час	177:39				Кількість відео		16

Таблиця Д.2 – Звіт тестування №2 для формування 180-хвилинних курсів

Тестування для компетенції №14							
Номер відеоролику	Тривалість	K1	K2	K3	K14	...	K20
36	10:12				X	.	X
436	10:51				X	.	X
468	10:23		X		X	.	
...
9379	10:47		X		X	.	
Сумарний час	176:14				Кількість відео		16

Таблиця Д.3 – Звіт тестування №3 для формування 180-хвилинних курсів

Тестування для компетенції №12							
Номер відеоролику	Тривалість	K1	K2	K9	K12	...	K20
12	10:54		X		X	.	
526	10:14			X		.	X
636	10:00		X		X	.	
...	
9761	10:02				X	.	X
Сумарний час	179:39				Кількість відео		17

Таблиця Д.4 – Звіт тестування №1 для формування 360-хвилинних курсів

Тестування для компетенції №17							
Номер відеоролику	Тривалість	K1	K2	K7	K17	...	K20
7	11:12				X	.	
15	10:17		X		X	.	X
162	11:34	X		X		.	
...	
9783	10:28				X	.	X
Сумарний час	356:11				Кількість відео		32

Таблиця Д.5 – Звіт тестування №2 для формування 360-хвилинних курсів

Тестування для компетенції №19							
Номер відеоролику	Тривалість	K1	K2	K3	...	K19	K20
36	11:12	X			.		
436	11:51			X	.	X	X
468	14:23				.	X	
...	
9379	10:47	X		X	.	X	
Сумарний час	354:52				Кількість відео		34

Таблиця Д.6 – Звіт тестування №3 для формування 360-хвилинних курсів

Тестування для компетенції №1							
Номер відеоролику	Тривалість	K1	K2	K3	K4	...	K20
2	10:32	X				.	
152	11:14	X		X		.	
157	15:01		X		X	.	
...	
9631	12:02	X				.	X
Сумарний час	357:57				Кількість відео		33

Таблиця Д.7 – Звіт тестування №1 для формування 720-хвилинних курсів

Тестування для компетенції №11							
Номер відеоролику	Тривалість	K1	K2	K10	K11	...	K20
73	12:27		X		X	.	
125	11:27	X		X		.	
236	12:56				X	.	X
...	
9245	14:15		X	X		.	
Сумарний час	709:56				Кількість відео		67

Таблиця Д.8 – Звіт тестування №2 для формування 720-хвилинних курсів

Тестування для компетенції №16							
Номер відеоролику	Тривалість	K1	K2	K14	K16	...	K20
46	10:25		X			.	X
251	11:59	X			X	.	
290	11:36		X	X		.	
...	
9427	13:22		X		X	.	
Сумарний час	711:28				Кількість відео		69

Таблиця Д.9 – Звіт тестування №3 для формування 720-хвилинних курсів

Тестування для компетенції №13							
Номер відеоролику	Тривалість	K1	K2	K8	K13	...	K20
112	11:32		X			.	X
257	12:14			X	X	.	
751	12:01		X			.	
...	
9927	13:02			X	X	.	
Сумарний час	716:37				Кількість відео		68

Додаток Е

Програмні коди

Лістинг generate_playlist.php:

```
class Video
{
    public $id;
    public $duration;
    public function __construct($id, $duration)
    {
        $this->id = $id;
        $this->duration = $duration;
    }

    public function getDetails() {
        return "Details: {$this->id} {$this->duration}";
    }
}

function duration($solution) {
    $dur = 0;
    foreach ($solution as $video) {
        $dur += $video->duration;
    }
    return $dur;
}

class GeneticAlgorithm
{
    private $videos = array();
    private $targetDuration;
    private $populationSize;
    private $generations;
    private $mutationRate;

    public function __construct($videos, $targetDuration, $populationSize, $generations, $mutationRate)
    {
        $this->videos = $videos;
        $this->targetDuration = $targetDuration;
        $this->populationSize = $populationSize;
        $this->generations = $generations;
        $this->mutationRate = $mutationRate;
    }

    public function run()
    {
        global $x;
        global $y;
        // Ініціалізуємо популяцію випадковими рішеннями
        $population = $this->initializePopulation();

        $k = 0;
        for ($generation = 0; $generation < $this->generations; $generation++) {
            $x[] = $generation;
            $y[] = count($population[0]) - 1;

            // Оцінюємо пристосованість (fitness) кожного рішення
            $population = $this->evaluatePopulation($population);

            // Відбираємо найкращі рішення для наступного покоління
```

```

    $selectedPopulation = $this->selection($population);

    // Створюємо нащадки (потомство) з обраних рішень
    $offspringPopulation = $this->crossover($selectedPopulation);

    // Мутуємо нащадків
    $offspringPopulation = $this->mutate($offspringPopulation);

    // Об'єднуємо обрану популяцію і нащадків
    $population = array_merge($selectedPopulation, $offspringPopulation);

    // Перевіряю, чи нащадки мають несумісні гени (повтори)
    foreach ($population as &$solution) {
        $solution = array_unique($solution);
    }
}

return $this->getBestSolution($population);
}

private function initializePopulation()
{
    $population = array();
    for ($i = 0; $i < $this->populationSize; $i++) {
        $solution = array();
        $durationSum = 0;
        $videosCopy = $this->videos;
        while ($durationSum < $this->targetDuration && !empty($videosCopy)) {
            $randomIndex = array_rand($videosCopy);
            $randomVideo = $videosCopy[$randomIndex];

            if ($durationSum + $randomVideo->duration <= $this->targetDuration) {
                $solution[] = $randomVideo->id;
                $durationSum += $randomVideo->duration;
            }

            unset($videosCopy[$randomIndex]);
            $videosCopy = array_values($videosCopy);
        }

        $population[] = $solution;
    }
    return $population;
}

private function evaluatePopulation($population)
{
    foreach ($population as &$solution) {
        $solution['fitness'] = count($solution);
    }
    unset($solution);
    return $population;
}

private function selection($population)
{
    usort($population, function ($a, $b) {
        return $b['fitness'] - $a['fitness'];
    });

    foreach ($population as &$solution) {
        unset($solution['fitness']);
    }
    unset($solution);
}

```

```

    return array_slice($population, 0, $this->populationSize / 2);
}

private function crossover($population)
{
    $offspringPopulation = array();
    for ($i = 0; $i < count($population); $i += 2) {
        $parent1 = $population[$i];

        if (array_key_exists($i + 1, $population)) {
            $parent2 = $population[$i + 1];
        } else {
            $parent2 = $parent1;
        }

        $crossoverPoint1 = rand(1, count($parent1) - 1);
        $crossoverPoint2 = rand(1, count($parent2) - 1);

        $offspring1 = array_merge(array_slice($parent1, 0, $crossoverPoint1), array_slice($parent2, $crossoverPoint2));
        $offspring2 = array_merge(array_slice($parent2, 0, $crossoverPoint2), array_slice($parent1, $crossoverPoint1));

        $this->fixOffspringDuration($offspring1);
        $this->fixOffspringDuration($offspring2);

        $offspringPopulation[] = $offspring1;
        $offspringPopulation[] = $offspring2;
    }
    return $offspringPopulation;
}

//корекція нащадків
private function fixOffspringDuration(&$offspring)
{
    $durationSum = 0;
    foreach ($offspring as $videoId) {
        foreach ($this->videos as $video) {
            if ($video->id === $videoId) {
                $durationSum += $video->duration;
                break;
            }
        }
    }
    if ($durationSum > $this->targetDuration) {
        while ($durationSum > $this->targetDuration) {
            $randomIndex = array_rand($offspring);
            $videoIdToRemove = $offspring[$randomIndex];
            foreach ($this->videos as $key => $video) {
                if ($video->id === $videoIdToRemove) {
                    $durationSum -= $video->duration;
                    unset($offspring[$randomIndex]);

                    $offspring = array_values($offspring);
                    break;
                }
            }
        }
    }
}

private function mutate($population)
{
    foreach ($population as &$solution) {
        if (mt_rand() / mt_getrandmax() < $this->mutationRate) {

```

```

if(empty($solution)) return $population;

$randomIndex = array_rand($solution);
// Замінюємо випадковий елемент на інший випадковий елемент
$solution[$randomIndex+1] = $this->videos[array_rand($this->videos)]->id;

// Перевіряємо, чи не перевищує "fitness" максимальне значення
$durationSum = $this->calculateDurationSum($solution);
if ($durationSum > $this->targetDuration) {
    // Видаляємо випадковий елемент, поки "fitness" не буде менше або дорівнювати максимальному
значенню
    while ($durationSum > $this->targetDuration) {
        $maxVideoId = 0;
        $maxDuration = 0;
        for($j = 0; $j < count($solution); $j++) {
            $duration = $this->getVideoDurationById($solution[$j]);
            if($duration < $maxDuration) {
                $maxVideoId = $j;
                $duration = $maxDuration;
            }
        }

        $videoIdToRemove = $solution[$maxVideoId];
        $durationSum -= $this->getVideoDurationById($videoIdToRemove);

        unset($solution[$maxVideoId]);
        $solution = array_values($solution);
    }
}

unset($solution);
return $population;
}

private function calculateDurationSum($solution)
{
    $durationSum = 0;
    foreach ($solution as $videoId) {
        $durationSum += $this->getVideoDurationById($videoId);
    }
    return $durationSum;
}

private function getVideoDurationById($videoId)
{
    foreach ($this->videos as $video) {
        if ($video->id === $videoId) {
            return $video->duration;
        }
    }
    return 0;
}

private function getBestSolution($population)
{
    $bestSolution = null;
    $bestFitness = PHP_INT_MIN;
    /*global $x;
    global $y;
    $countValues = [];

```

```

$к = 0;
foreach ($population as $item) {
    if($к % 3 == 0) {
        if (!isset($countValues[count($item)]) {
            $countValues[count($item)] = 0;
        }
        $countValues[count($item)]++;
    }
    $к++;
}

// Відсортовані ключі для збереження порядку
$x = array_keys($countValues);
sort($x);

// Побудова масиву у на основі відсортованих ключів x
foreach ($x as $key) {
    $y[] = $countValues[$key];
}*/

foreach ($population as &$solution) {
    $solution['fitness'] = count($solution);
}
unset($solution);

foreach ($population as &$solution) {
    if ($solution['fitness'] > $bestFitness) {
        $bestSolution = $solution;
        $bestFitness = $solution['fitness'];
    }
}
unset($solution);

$population = array_filter($population, function($solution) use ($bestFitness) {
    return $solution['fitness'] == $bestFitness;
});

usort($population, function($a, $b) {
    return $this->calculateDurationSum($a) - $this->calculateDurationSum($b);
});

$bestElement = reset($population);
$bestElement["time"] = $this->calculateDurationSum($bestElement);

return $bestElement;
}
?>

```

<?php

```

include 'components/connect.php';

function getVideoDuration($name) {
    $videoPath = 'uploaded_files/' . $name;
    $ffmpegOutput = shell_exec("ffmpeg -i $videoPath 2>&1");

    preg_match('/Duration: (.*)/, $ffmpegOutput, $durationMatches);
    preg_match('/(\d+) fps/', $ffmpegOutput, $fpsMatches);

    $duration = isset($durationMatches[1]) ? $durationMatches[1] : '00:00:00';
    $fps = isset($fpsMatches[1]) ? intval($fpsMatches[1]) : 30;

    $durationParts = explode(':', $duration);

```

```

$totalSeconds = ($durationParts[0] * 3600) + ($durationParts[1] * 60) + $durationParts[2];

return $totalSeconds;
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $start = microtime(true);
    $sql = "SELECT c.* FROM content c
    JOIN playlist p ON c.playlist_id = p.id
    JOIN playlist_tag pt ON p.id = pt.playlist_id
    JOIN tag t ON pt.tag_id = t.id
    WHERE t.id = " . $_POST['category'] . " AND c.status = 'active'";

    $result = $conn->query($sql);

    $videos = [];
    if ($result->rowCount() > 0) {
        while($row = $result->fetch(PDO::FETCH_ASSOC)) {
            $videos[] = new Video($row["id"], getVideoDuration($row["video"]));
        }
    }

    $targetDuration = 0;
    $targetDuration = $_POST['time'] * 60;

    $ga = new GeneticAlgorithm($videos, $targetDuration, 100, 100, 0.15);
    $bestSolution = $ga->run();
    $end = microtime(true);
    $bestSolution["workTime"] = ($end - $start);

    setcookie("my_playlist", json_encode($bestSolution), time() + 3600, "");

    header("Location: quick_playlist.php");
    exit;
}
?>

```

Лістинг quick_playlist.php:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>playlist</title>

    <!-- font awesome cdn link -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.0/css/all.min.css">

    <!-- custom css file link -->
    <link rel="stylesheet" href="css/style.css">

</head>
<body>

<?php include 'components/user_header.php'; ?>

<!-- videos container section starts -->
<section class="videos-container">

    <h1 class="heading">playlist videos</h1>

```

```

<?php
if ($cookieTime !== null) {
    echo "<h2 class='time-heading'>(total duration ~ ". round($cookieTime/60) ."m.)</h2>";
    echo "<h2 class='time-heading'>(generation time ~ ". round($cookieWorkTime, 2) ."µs)</h2>";

}
?>

<div class="box-container">

<?php
if($videoIds != null) {

    $placeholders = str_repeat('?', count($videoIds) - 1) . '?';

    $sql = "SELECT * FROM `content`
    WHERE id IN ($placeholders) AND status = 'active'
    ORDER BY date DESC, id DESC";
    $select_content = $conn->prepare($sql);

    $select_content->execute($videoIds);
    if($select_content->rowCount() > 0){
        while($fetch_content = $select_content->fetch(PDO::FETCH_ASSOC)){
            ?>

<a target="_blank" href="watch_video.php?get_id=<?= $fetch_content['id']; ?>" class="box">
    <i class="fas fa-play"></i>
    
    <h3><?= $fetch_content['title']; ?></h3>
</a>

<?php
    }
    } else {
        echo '<p class="empty">no videos added yet!</p>';
    }
} else {
    echo '<p class="empty">no videos added yet!</p>';
}
?>

</div>

</section>

<!-- videos container section ends -->

<?php include 'components/footer.php'; ?>

<!-- custom js file link -->
<script src="js/script.js"></script>

</body>
</html>

```

Лістинг tests.php:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

<title>playlist</title>

<!-- font awesome cdn link -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@6.2.0/css/all.min.css">
<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
<!-- custom css file link -->
<link rel="stylesheet" href="css/style.css">

</head>
<body style="padding:0;">
<!--
<div id="myDiv" style="width:100%; height:700px;"></div>
<?php /*
    $x = [];
    $y = [];
    $z = [];

    for ($i = 1; $i <= 40; $i += 1) {
        $i = round($i, 2);
        $x[] = $i;
    }

    for ($i = 1; $i <= 40; $i += 1) {
        $i = round($i, 2);
        $y[] = $i;
    }

    for ($i = 2; $i <= 40; $i += 1) {
        $i = round($i, 2);
        $sinnerArray = [];
        for ($j = 1; $j <= 40; $j += 1) {
            $middle = 0;
            for($a = 1; $a <= 30; $a += 1) {
                $ga = new GeneticAlgorithm($videos, $targetDuration, $i, $j, 0.1);
                $bestSolution = $ga->run();
                $middle += (count($bestSolution)-1) - pow(count($bestSolution)-1, 1/($j+2));
            }
            $sinnerArray[] = $middle/30;
        }
        $z[] = $sinnerArray;
    }
*/
?>

<script>
var data = [{
    x: <?php //echo json_encode($x); ?>,
    y: <?php //echo json_encode($y); ?>,
    z: <?php //echo json_encode($z); ?>,
    type: 'surface'
}];

console.log(data);

var layout = {
    title: '3D Surface Plot',
    scene: {
        xaxis: {title: 'Кількість поколінь'},
        yaxis: {title: 'Кількість особин'},
        zaxis: {title: 'Кількість відео'}
    }
};

Plotly.newPlot('myDiv', data, layout);

```

```

</script>

<?php
/*
try {

    $fullCount = 0;
    $file = fopen('file2.txt', 'r');
    $videos = [];
    $rowCount = 0;
    if ($file) {
        while (($line = fgets($file)) !== false && $rowCount < 1000) {
            list($id, $time) = explode("\t", trim($line));
            $video = new Video($id, $time);
            $videos[] = $video;
            $rowCount += 1;
        }
        fclose($file);
    } else {
        echo "Помилка при відкритті файлу.";
    }

    $ga = new GeneticAlgorithm($videos, 1000, 31, 31, 0.15);
    $bestSolution = $ga->run();

} catch (Exception $e) {
    echo 'PHP перехватил исключение: ', $e->getMessage(), "\n";
} */
?>

```

```

<div id="myDiv" style="width:100%; height:700px;"></div>

```

```

<script>
var data = [{
    x: <?php echo json_encode($x); ?>,
    y: <?php echo json_encode($y); ?>,
    type: 'bar',
    marker: {
        color: '#C8A2C8',
        line: {
            width: 2.5
        }
    }
}
];

```

```

console.log(data);

```

```

var layout = {
    title: 'Scatter Plot',
    scene: {
        xaxis: {title: 'Кількість відео'},
        yaxis: {title: 'Покоління'},
    }
};

```

```

Plotly.newPlot('myDiv', data, layout);

```

```

</script>

```

```

<div id="myDiv" style="width:100%; height:700px;"></div>

```

```

<script>
var trace1 = {
    x: <?php echo json_encode($x); ?>,

```

```

y: <?php echo json_encode($y); ?>,
type: 'scatter',
name: 'Оптимізована',
line: {
  width: 5
}
};

var trace2 = {
x: <?php echo json_encode($x2); ?>, y: <?php echo json_encode($y2); ?>,
type: 'scatter',
name: 'Неоптимізована',
line: {
  width: 5
}
};

var data = [trace1, trace2];

console.log(data);

var layout = {
  title: 'Scatter Plot',
};

Plotly.newPlot('myDiv', data, layout);

</script>
-->
<?php
/*
try {
  set_time_limit(0);
  $file = fopen('file.txt', 'r');
  $videos = [];

  if ($file) {
    while (($line = fgets($file)) !== false) {
      list($id, $time) = explode("\t", trim($line));
      $video = new Video($id, $time);
      $videos[] = $video;
    }
    fclose($file);
  } else {
    echo "Помилка при відкритті файлу.";
  }

  $solution = [];
  for ($i = 15; $i <= 40; $i += 1) {
    $i = round($i, 2);
    $innerArray = [];

    for ($j = 15; $j <= 40; $j += 1) {
      for ($k = 0.1; $k <= 0.3; $k += 0.02) {
        $fullCount = 0;
        $middleTime = 0;
        $start = microtime(true);

        for ($a = 1; $a <= 5; $a += 1) {
          $ga = new GeneticAlgorithm($videos, 100, $i, $j, $k);
          $bestSolution = $ga->run();
          $fullCount += (count($bestSolution)-1) - pow(count($bestSolution)-2, 1/($j+2)) - abs(0.15-$k);
        }
      }
    }
  }
}

```

```

        $send = microtime(true);
        $middleTime = ($send - $start) * 1000 / 5;
        $middleCount = $fullCount/5;
        $corr = $middleCount / $middleTime;
        $res = [];
        $res["text"] = "Кількість особин: $i, кількість поколінь: $j, шанс мутації: $k || Середня к-сть відео:
$middleCount, середній час (мс): $middleTime, співвідношення: $corr <br/>";
        $res["corr"] = $corr;
        $solution[] = $res;
        echo $res["text"];
    }
}

$bestText = "";
$bestCorr = -1;
foreach ($solution as $el) {
    if($el["corr"] > $bestCorr) {
        $bestCorr = $el["corr"];
        $bestText = $el["text"];
    }
}
echo "<br/><br/>Найкращий результат:<br/><br/>";
echo $bestText;

} catch (Exception $e) {
    echo 'PHP перехватил исключение: ', $e->getMessage(), "\n";
}*/
?>

<?php
/*
try {
    set_time_limit(0);

    for($i = 10000; $i > 0.11; $i = $i / 10) {
        $middleTime = 0;
        $middleCount = 0;

        for($a = 0; $a < 10; $a += 1) {
            $fullCount = 0;
            $file = fopen('file2.txt', 'r');
            $videos = [];
            $rowCount = 0;
            if ($file) {
                while (($line = fgets($file)) !== false && $rowCount < $i) {
                    list($id, $time) = explode("\t", trim($line));
                    $video = new Video($id, $time);
                    $videos[] = $video;
                    $rowCount += 1;
                }
                fclose($file);
            } else {
                echo "Помилка при відкритті файлу.";
            }

            $start = microtime(true);

            $ga = new GeneticAlgorithm($videos, 1000, 31, 31, 0.15);
            $bestSolution = $ga->run();

            $fullCount += count($bestSolution) - 2;

            $send = microtime(true);

```

```

$time = ($end - $start) * 1000;

$middleTime += $time;
$middleCount += $fullCount;

echo "Кількість відео: $fullCount, час виконання ~ $time мс <br/>";
}

$middleTime = $middleTime / 10;
$middleCount = $middleCount / 10;
echo "<br/>-----<br/>";
echo "<br/>Середній результат для $i файлів:<br/>";
echo "-Середній час виконання: $middleTime мс<br/>";
echo "-Середній кількість відео: $middleCount <br/>";
echo "<br/>-----<br/><br/>";
}

} catch (Exception $e) {
    echo 'PHP перехватил исключение: ', $e->getMessage(), "\n";
}
*/
?>
<?php
/*
try {

    $fullCount = 0;
    $file = fopen('file2.txt', 'r');
    $videos = [];
    $rowCount = 0;
    if ($file) {
        while (($line = fgets($file)) !== false && $rowCount < 100) {
            list($id, $time) = explode("\t", trim($line));
            $video = new Video($id, $time);
            $videos[] = $video;
            $rowCount += 1;
        }
        fclose($file);
    } else {
        echo "Помилка при відкритті файлу.";
    }

    $ga = new GeneticAlgorithm($videos, 1000, 100, 100, 0.12);
    $bestSolution = $ga->run();

} catch (Exception $e) {
    echo 'PHP перехватил исключение: ', $e->getMessage(), "\n";
}
*/
?>

</body>

```

Додаток Є

Презентаційний матеріал

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Кафедра комп'ютерних наук

Метод покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми

ВИКОНАВ: ВАЛЕРІЙ ЧИСТЯКОВ
СТУДЕНТ: КН-20-2

КЕРІВНИК: ОЛЕКСАНДР ПАСІЧНИК

Актуальність

В час швидкого розвитку технологій, людям потрібно навчитись так само швидко адаптуватись до них, що зазвичай може проявлятись у потребі освоювати нові сфери знань.

Розвиток сфери онлайн-платформ для навчання є важливим у цьому завданні, оскільки це сприяє доступності освіти для широкого кола людей незалежно від їх місця проживання або графіку. Крім того, розвиток таких систем сприяє інноваціям у методах навчання та стимулює викладачів та експертів до пошуку нових підходів до навчання та педагогіки.

Розвиток онлайн-курсів також сприяє розширенню доступності освіти для людей з обмеженими можливостями, які можуть знайти у навчанні в інтернеті більш комфортну та доступну форму ніж традиційні навчальні заклади.



Об'єкт дослідження

Об'єкт дослідження - процес покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми.



Предмет дослідження - методи та засоби кібернетичного аналізу для покращення вмісту навчальних курсів.



Мета та задачі

Метою кваліфікаційної роботи бакалавра є покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми.

Для досягнення мети визначено такі задачі дослідження:

- провести аналіз методів формування та покращення вмісту навчальних дисциплін,
 - реалізувати метод покращення вмісту навчальних курсів за генетичним алгоритмом,
 - виконати програмну реалізацію методу покращення вмісту навчальних курсів за генетичним алгоритмом як освітньої вебсистеми,
 - провести експериментальне методу та його програмної реалізації,
 - виконати оцінювання покращення вмісту навчальних курсів.
- 
- 

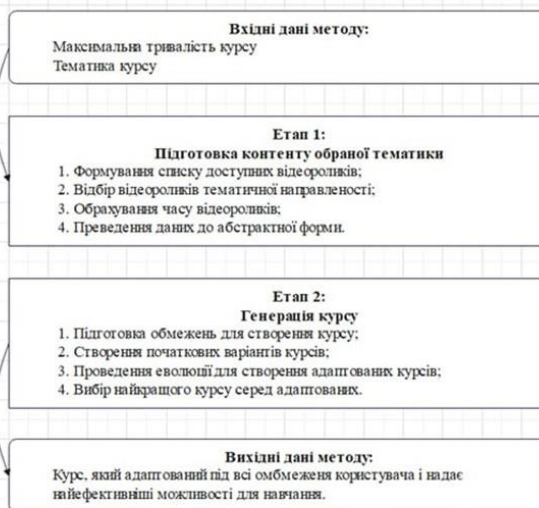
Аналіз предметної області

Сучасні методи комп'ютерних наук значно сприяють розвитку інтернет-технологій, особливо у створенні вебзастосунків.

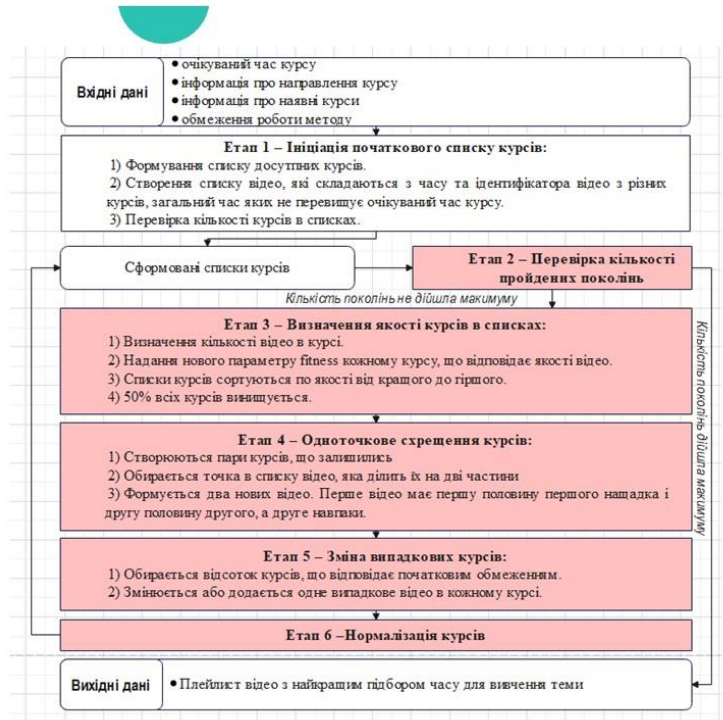
Вебсайти стали основою цифрової взаємодії, забезпечуючи функції і в сфері освіти, що сприяє гнучкому та доступному навчанню.

Для покращення освіти, активно використовують алгоритми машинного навчання, включно з генетичними алгоритмами.

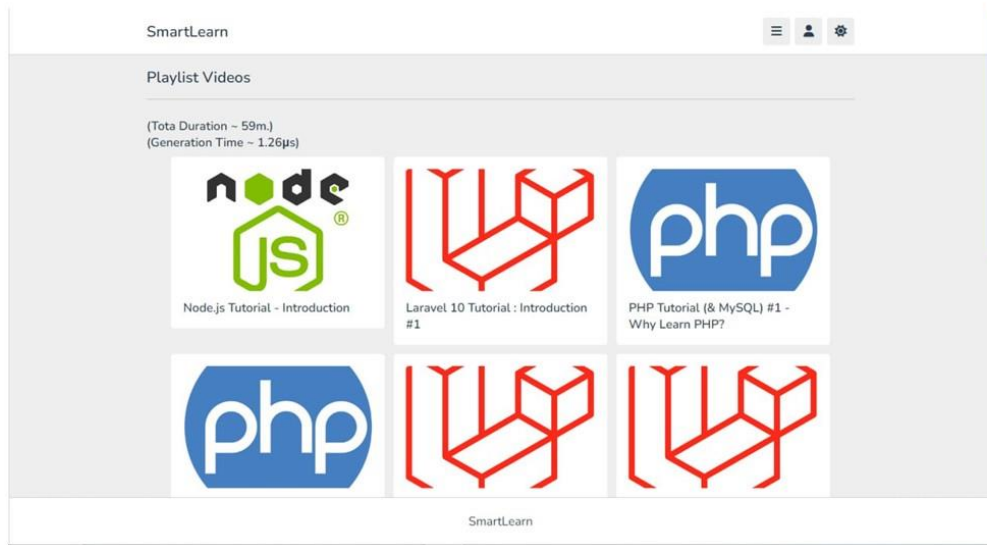
Запропонований метод



Імплементация генетического алгоритма



Результати роботи



Тестування

| Тестування для компетенції №4 | | | | | | | |
|-------------------------------|------------|----|----|----|-----------------|-----|-----|
| Номер відеоролику | Тривалість | К1 | К2 | К3 | К4 | ... | К20 |
| 9 | 11:09 | X | | X | | . | |
| 278 | 13:07 | | | | | . | |
| 596 | 10:32 | | X | | X | . | |
| ... | . | . | . | . | . | . | . |
| 9689 | 10:28 | X | | | | . | X |
| Сумарний час | 177:39 | | | | Кількість відео | | 16 |

| Кількість файлів | Середня кількість відеороликів | Середній час виконання |
|------------------|--------------------------------|------------------------|
| 10000 | 8.1 | ~10.88 секунд |
| 1000 | 7.8 | ~232.4 мілісекунд |
| 100 | 7.3 | ~13.17 мілісекунд |
| 10 | 5 | ~2.7 мілісекунд |
| 1 | 1 | ~1.2 мілісекунд |

При виконанні кваліфікаційної роботи бакалавра було вирішено такі задачі дослідження:

- проведено аналіз методів формування та покращення вмісту навчальних дисциплін;
- проведено аналіз інтелектуальних методів для визначення найкращого вмісту навчальних курсів;
- реалізовано метод покращення вмісту навчальних курсів за генетичним алгоритмом
- виконано програмну реалізацію методу покращення вмісту навчальних курсів з адаптацією генетичного алгоритму;
- проведено експериментальне тестування застосунку;
- виконано оцінювання покращення вмісту навчальних курсів.

Висновки



Дякую за увагу!

Anti-Plagiarism v-15.257**Максимальне співпадіння з одним документом 2.0%**

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 8%

| | | | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------|-----------------------------|---------|
| ID: 130304
Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми
Додано в БД: 2024-06-13
Автора: Валерій ЧИСТЯКОВ
Керівники: Олександр ПАСІЧНИК
Консультанти:
Опоненти: | Документ | | Сумарний збіг по Базі Даних | |
| | Символи | Лексеми | Символи | Лексеми |
| | 93005 | 1435 | 2707 (3%) | 43 (3%) |

Джерело плагіату

| ID | Опис | Наявність плагіату в документі | |
|----|------|--------------------------------|---------|
| | | Символи | Лексеми |
| | | | |

Ім'я користувача:
Кафедра КН

ID перевірки:
1016357778

Дата перевірки:
13.06.2024 18:45:23 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
13.06.2024 18:46:02 EEST

ID користувача:
100005671

Назва документа: КН-20-2 Чистяков_ЗАПИСКА

Кількість сторінок: 73 Кількість слів: 14666 Кількість символів: 115970 Розмір файлу: 1.91 MB ID файлу: 1016162162

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

6.85% Схожість

Найбільша схожість: 2.52% з джерелом з Бібліотеки (ID файлу: 1016133152)

4.99% Джерела з Інтернету

837

Сторінка 75

3.7% Джерела з Бібліотеки

159

Сторінка 79

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування

11
сторінок

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми

Автор: студент групи КН-20-2 Чистяков Валерій

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.т.н., доцент Олександр Пасічник

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок | Позначка про відповідність |
|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту. | відповідає |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи | |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |

Підтвердження:

Запозичення, виявлені в роботі Валерія Чистякова, не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти програмного коду, що не мають авторства і містять поширені конструкції; серед запозичень знаходяться загальновідомі терміни, скорочення та визначення.

Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості, складає:

- за системою Anti-Plagiarism: 2%;

- за системою Unichек: 6,85 %,

що є допустимими запозиченнями.

Керівник роботи



Олександр ПАСІЧНИК

Гарант ОП



Олександр МАЗУРЕЦЬ

Завідувач кафедри КН



Олександр БАРМАК



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук



ВІДГУК НАУКОВОГО КЕРІВНИКА на кваліфікаційну роботу бакалавра

студента гр. КН-20-2 Чистякова Валерія

за темою Метод покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми

1. Актуальність теми

В час швидкого розвитку технологій, людям потрібно навчитись так само швидко адаптуватись до них, що зазвичай може проявлятися у потребі освоювати нові сфери знань. Розвиток сфери онлайн-платформ для навчання є важливим у цьому завданні, оскільки це сприяє доступності освіти для широкого кола людей незалежно від їх місця проживання або графіку. Крім того, розвиток таких систем сприяє інноваціям у методах навчання та стимулює викладачів та експертів до пошуку нових підходів до навчання та педагогіки. Такий підхід дозволяє впроваджувати нові технології та методи навчання, що може покращити якість освіти в загальному. Окрім цього, розвиток онлайн-курсів також сприяє розширенню доступності освіти для людей з обмеженими можливостями, які можуть знайти у навчанні в інтернеті більш комфортну та доступну форму ніж традиційні навчальні заклади.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

За стандартом, а саме описом предметної області, об'єктами вивчення та діяльності є математичні, інформаційні, імітаційні моделі реальних явищ, об'єктів, систем і процесів та методи і технології отримання, зберігання, обробки, передачі та використання інформації. Метою роботи саме є розробка методу покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми. При вирішенні поставленої задачі використано відповідні математичні моделі, методи та алгоритми розв'язання теоретичних і прикладних задач. Тому результати виконання кваліфікаційної роботи бакалавра відповідають стандарту бакалавра спеціальності 122 – Комп'ютерні науки.

3. Професійні та особистісні якості бакалавра

При роботі над кваліфікаційною роботою бакалавра Чистяков Валерій проявив себе кваліфікованим фахівцем та дисциплінованим студентом, вчасно виконуючи

поставлені етапи дослідження. Як в процесі написання пояснювальної записки, так і при розробці Методу покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми та його програмній реалізації, тестування та експериментальній перевірці проявив достатні для одержання успішного результату компетентності та результати навчання. Опанував професійні скіли за напрямком «Комп'ютерні науки» та достатньо значний софт скіл.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Одержані в роботі результати є наслідком особистої діяльності студента, який самостійно виконував всі поставлені задачі.

5. Ступінь оволодіння методами дослідження

При виконанні кваліфікаційної роботи показав достатній рівень компетентностей та володіння необхідними інструментами та обладнанням, методами, методиками та технологіями предметної області комп'ютерних наук.

6. Повнота та якість розкриття теми роботи

Тема роботи в повній мірі обґрунтована й розкрита, проведено аналіз актуальності та відомих досліджень в межах обраної теми, поставлені завдання, які у роботі виконані, та виконано програмну реалізацію для валідації та верифікації запропонованого методу.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне.

8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Розроблений у роботі метод та його програмна реалізація може бути використана для імплементації системи загального та професійного вдосконалення особистості протягом життя в освітній простір.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

Враховуючи високий рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник



к.т.н., доцент каф. КН Олександр ПАСІЧНИК



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ



Кафедра комп'ютерних наук

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента *гр. КН-20-2 Чистякова Валерія*

за темою: Метод покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми

1. Актуальність обраної теми

В час швидкого розвитку технологій, людям потрібно навчитись так само швидко адаптуватись до них, що зазвичай може проявлятися у потребі освоювати нові сфери знань. Розвиток сфери онлайн-платформ для навчання є важливим у цьому завданні, оскільки це сприяє доступності освіти для широкого кола людей незалежно від їх місця проживання або графіку. Такий підхід дозволяє впроваджувати нові технології та методи навчання, що може покращити якість освіти в загальному. Це сприяє зменшенню витрат на навчання для студентів.

2. Повнота розкриття мети та завдань роботи

Мета роботи розкрита повністю, всі завдання виконані.

3. Зміст кожного розділу роботи

В першому розділі виконано аналіз сучасних підходів до вирішення завдань предметної області та сучасного стан. Визначено мету роботи та виконано постановку завдань. В другому розділі реалізовано метод покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми. Визначено критерії оцінки покращення. В третьому розділі виконано програмну реалізацію методу та її тестування. Виконано оцінку покращення вмісту навчальних курсів.

4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблений метод покращення вмісту навчальних курсів за генетичним алгоритмом для освітньої вебсистеми може бути використаний у сфері освіти протягом життя.

5. Якість оформлення кваліфікаційної роботи бакалавра

Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне.

6. Недоліки кваліфікаційної роботи бакалавра

В роботі використано два критерії для оцінки покращення, але не зазначено чи є цей перелік вичерпним.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Рецензент

*В. Мартинюк, д.т.н., проф.,
зав. каф. АІІТ та Р ХНУ*