
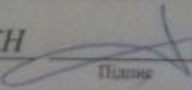
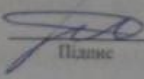


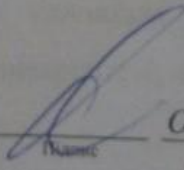
Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Спосіб формування пулу вибіркового навчальних дисциплін на основі генетичного алгоритму

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 4 курсу, група КН-19-1  Н.В. Плахтій
Курс, група виконавця Підпис Ініціали, прізвище
Керівник: к.т.н., доцент кафедри КН  О.А. Пасічник
Науковий ступінь, посада Підпис Ініціали, прізвище
Нормоконтроль: к.т.н., доцент кафедри КН  Р.О. Багрії
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор  О.В. Бармак
Підпис Ініціали, прізвище
1 06 2023 р.

Хмельницький 2023

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 О.В. Бармак
« 6 » 03 2023 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Спосіб формування пулу вибіркового навчальних дисциплін на основі генетичного алгоритму»

2. Завдання видано студенту Плахтію Назарію Володимировичу
(прізвище, ім'я, по батькові)

3. Керівник роботи доцент кафедри КН Пасічник Олександр Анатолійович
(посада, прізвище, ім'я, по батькові)

Затверджено наказом університету від « 1 » 03 2023р. № 5

5. Дата видачі завдання студенту: « 3 » 03 2023р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані. Провести аналіз методів формування пулу вибіркового дисциплін, його мети та завдань; провести аналіз можливостей, переваг та недоліків генетичного алгоритму; реалізувати інформаційну технологію визначення пулу вибіркового навчальних дисциплін на основі генетичного алгоритму; провести експериментальне тестування інформаційної технології. Вихідними даними є навчальні дисципліни, що викладаються в Хмельницькому національному університеті, та їх атрибутивні характеристики.

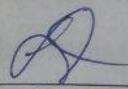
7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником	грудень 2022	Виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	січень 2023	Виконано
3	Робота над розділом 1 – Характеристика предметної області та постановка задачі	січень 2023	Виконано
4	Робота над розділом 2 – Спосіб формування пулу вибіркового навчальних дисциплін на основі генетичного алгоритму	березень 2023	Виконано
5	Робота над розділом 3 – Програмна реалізація способу формування пулу вибіркового навчальних дисциплін на основі генетичного алгоритму	квітень 2023	Виконано
6	Оформлення пояснювальної записки згідно вимог	травень 2023	Виконано
7	Попередній захист кваліфікаційної роботи бакалавра	травень 2023	Виконано
8	Захист кваліфікаційної роботи бакалавра на засіданні Екзаменаційної комісії	червень 2023	Виконано

Виконавець:

студент 4 курсу, група КН-19-1

Курс, група виконавця


Підпис

Н.В. Плахтій

Ініціали, прізвище

Керівник:

к.т.н., доцент кафедри КН

Науковий ступінь, посада


Підпис

О.А. Пасічник

Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: Спосіб формування пулу вибіркового навчальних дисциплін на основі генетичного алгоритму

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-19-1 Плахтій Назарій Володимирович

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

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
64	45	7	31	2

Мета кваліфікаційної роботи полягає в створенні способу формування пулу вибіркового навчальних дисциплін на основі генетичного алгоритму. Для досягнення поставленої мети визначені наступні задачі дослідження: провести аналіз методів формування пулу вибіркового навчальних дисциплін, його мети та завдань; провести аналіз можливостей, переваг та недоліків генетичного алгоритму; реалізувати інформаційну технологію визначення пулу вибіркового навчальних дисциплін на основі генетичного алгоритму; провести експериментальне тестування інформаційної технології.

Результатом виконання кваліфікаційної роботи бакалавра є розробка способу формування пулу вибіркового навчальних дисциплін на основі генетичного алгоритму та його реалізація як інформаційної системи.

Ключові слова: генетичний алгоритм, спосіб формування пулу, вибіркова дисципліна.

Виконавець: студент 4 курсу, група КН-19-1
Курс, група виконавця


Підпис

Н.В. Плахтій
Ініціали, прізвище

Зміст

Перелік скорочень	3
Вступ.....	4
Розділ 1 Характеристика предметної області та постановка задачі	6
1.1 Аналіз предметної області	6
1.2 Аналіз сучасних підходів формування пулу вибірових дисциплін в закладах вищої освіти	9
1.3 Оптимізація за генетичними алгоритмами.....	11
1.4 Аналіз існуючих програмних рішень.....	15
1.5 Мета, задачі та вимоги до реалізації інформаційної системи	17
Розділ 2 Спосіб формування пулу вибірових навчальних дисциплін на основі генетичного алгоритму	18
2.1 Загальні положення щодо складу вибірових дисциплін у закладах вищої освіти	18
2.2 Послідовність формування переліку вибірових дисциплін закладами вищої освіти	20
2.3 Особливості використання генетичного алгоритму для задачі побудови бажаного пулу вибірових навчальних дисциплін для здобувача вищої освіти.....	25
2.3.1. Загальна схема оптимізаційної задачі.....	25
2.3.2. Загальна структура генетичного алгоритму.....	26
2.4 Функціональна структура інформаційної системи.....	34
2.5 Проектування структури інформаційної системи формування пулу вибірових навчальних дисциплін для здобувача вищої освіти	37
2.6 Висновки до розділу 2	41
Розділ 3 Програмна реалізація способу формування пулу вибірових навчальних дисциплін на основі генетичного алгоритму	43
3.1 Структура модулів системи, їх взаємозв'язок	43
3.2 Опис функціональних можливостей інформаційної системи	48
3.3 Висновки до розділу 3	59
Висновки	60
Перелік посилань.....	62
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
КН	Комп'ютерні науки
ІНП	Індивідуальний навчальний план
ОПП	Освітньо-професійна програма
ЗВО	Заклад вищої освіти
ECTS	Європейська система трансферу і накопичення кредитів
CRUD	Створити, переглянути, оновити, видалити

Вступ

Кваліфікаційна робота бакалавра присвячена розробці способу формування пулу вибірових навчальних дисциплін на основі генетичного алгоритму.

Актуальність теми. Базовою тенденцією розвитку сьогодення є широке, практично всеохоплююче запровадження інформаційних технологій зі стійким трендом до їх поглибленої інтелектуалізації. Гнучким та ефективним кібернетичним інструментом вирішення достатнього широкого кола практичних задач є генетичні алгоритми. Іншою панівною тенденцією є широка модернізація системи освіти, переосмислення її місця та ролі у суспільному житті. Це супроводжується запровадженням нової парадигми та новітніх підходів щодо формування змісту освітнього процесу. Основним принципом сучасної вищої освіти є студентоцентрикований підхід, що передбачає, з одного боку, безпосередню участь здобувача вищої освіти у формуванні власної освітньої траєкторії, а, з іншого боку, надання йому широких можливостей щодо напрямку та рівня отримуваних знань та вмінь. Практична реалізація зазначеного підходу передбачає наявність в освітній програмі обов'язкових та вибірових навчальних дисциплін. Для задоволення потреб здобувачів вищої освіти навчальні заклади пропонують широкий та різноманітний перелік вибірових навчальних дисциплін, до прикладу, в Хмельницькому національному університеті їх кількість перевищує 1000. Остання обставина робить нетривіальною задачу визначення переліку вибірових дисциплін, що додатково ускладняється різноманіттям прагнень здобувачів вищої освіти щодо результатів навчання.

Мета і задачі роботи. Мета кваліфікаційної роботи полягає в створенні способу формування пулу вибірових навчальних дисциплін на основі генетичного алгоритму.

Для досягнення поставленої мети визначені наступні задачі дослідження:

– провести аналіз методів формування пулу вибірових дисциплін, його мети та завдань;

– провести аналіз можливостей, переваг та недоліків генетичного алгоритму;

– реалізувати інформаційну технологію визначення пулу вибірових навчальних дисциплін на основі генетичного алгоритму;

– провести експериментальне тестування інформаційної технології.

Об’єкт дослідження – процес побудови пулу вибірових навчальних дисциплін на основі генетичного алгоритму.

Предмет дослідження – моделі, алгоритми та засоби для створення способу пулу вибірових навчальних дисциплін на основі генетичного алгоритму.

Розділ 1 Характеристика предметної області та постановка задачі

1.1 Аналіз предметної області

Навчання є невід'ємною частиною нашого життя, тому кожному хто проходить цей шлях доводиться мати справу з різноманітними освітніми установами [1]. Вступаючи до університету кожен абітурієнт обирає факультет та спеціальність, що йому до вподоби. Надалі навчання складається з відповідних академічних дисциплін [2], що в сумі формують освітню програму [3].

Академічна дисципліна – це підрозділ знань, який викладають і досліджують на рівні коледжу чи університету. Дисципліни визначаються академічними журналами, в яких публікуються дослідження, а також науковими товариствами та академічними департаментами або факультетами коледжів і університетів, до яких належать їхні практики. Навчальні дисципліни умовно поділяються на гуманітарні, включаючи мову, мистецтво та культурологію, і наукові дисципліни, такі як фізика, хімія та біологія, соціальні науки іноді вважаються третьою категорією.

Формування і реалізація освітніх програм є головним завданням у діяльності закладів вищої освіти. При оцінці освітньої діяльності того чи іншого закладу вищої освіти до уваги насамперед береться якість його освітніх програм. Освітньою програмою називають комплекс освітніх компонентів з відповідною логічною складовою та їх вивчення для досягнення визначених результатів навчання. Якісними освітніми програмами можна вважати ті, що по завершенню можуть гарантувати абітурієнту робоче місце відповідно до опанованого курсу чи продовження навчання на вищому рівні. Приваблювати студента мають саме такі освітні програми, а не зовнішні чинники, як наприклад регіональна близькість закладу, гарантована можливість отримання стипендіальних виплат, уникнення складання додаткових екзаменаційних програм чи низький поріг для вступу. Отже щоб розпочати навчання за обраною освітньою програмою необхідно в першу чергу відповідати встановленим умовам допуску до цього

курсу. Кожна навчальна програма створюється викладачем та затверджується головуючим органом. Також слід зауважити, що однакові навчальні програми не можуть використовуватись для різних освітніх програм, оскільки кожен напрям має свої особливості.

Все ж основною ідеєю формування освітньої програми являється ідея студентоцентрованого навчання [4, 5]. В свою чергу найбільш ефективно організована студентоцентрована навчальна програма потребує індивідуальної траєкторії навчання [6].

Індивідуальна траєкторія навчання передбачає відмову від прив'язування студента до певної академічної групи чи потоку, така програма акцентує увагу саме на усвідомленні студентом власних цілей, потреб суспільства та наочної спрямованості освітніх інтересів й необхідності їх поєднання з суспільними потребами. Отже відповідно до індивідуальної траєкторії навчання студент отримує право вибору між декількома альтернативними навчальними дисциплінами та окремими курсами з певної навчальної дисципліни. Таким чином створюються умови для студентоцентрованого навчання, коли не викладач йде до призначеної йому навчальним відділом чи кафедрою студентської аудиторії, а навпаки студентська група для вивчення певної дисципліни формується навколо викладача. Викладач та його навчальна програма для вивчення відповідної дисципліни є тим, що й має кожного разу об'єднувати студентів у групи [7].

Навчальні дисципліни за вибором студента – це дисципліни, які вводяться вищим навчальним закладом для більш повного задоволення освітніх і кваліфікаційних вимог особи для потреб суспільства, ефективнішого використання можливостей закладу освіти, врахування регіональних особливостей тощо. Вибіркові навчальні дисципліни професійної та практичної підготовки надають можливість здійснення поглибленої підготовки за спеціальностями та спеціалізаціями, що визначають характер майбутньої діяльності; сприяють академічній мобільності студента та його особистим інтересам, дозволяють здійснювати впровадження спеціалізацій у межах базової спеціальності з метою формування компетентності здобувача відповідно до

вимог ринку праці. Також слід зауважити, що освітня програма з будь-якої спеціальності містить нормативну складову та варіативну [8].

Загалом усі вибіркові дисципліни можна поділити на шість категорій: професійно-орієнтовані, гуманітарні, природничі, соціально-економічні, мистецькі та міждисциплінарні.

Професійно-орієнтовані дисципліни – надають спеціалізовану підготовку з певної професії. Ці дисципліни часто включають практичну роботу і навчання на робочому місці, щоб допомогти студентам здобути практичні навички, необхідні для роботи у відповідній галузі.

Гуманітарні дисципліни – вивчають людську поведінку, культуру, історію і мову. Ці дисципліни надають студентам розуміння того, як люди сприймають світ навколо себе і як вони взаємодіють з ним.

Природничі дисципліни – вивчають природу, включаючи біологію, фізику, хімію і геологію. Ці дисципліни допомагають студентам зрозуміти природні процеси, які відбуваються на нашій планеті, і впливають на людське життя.

Соціально-економічні дисципліни – вивчають економіку, соціологію, політику і психологію. Ці дисципліни допомагають студентам зрозуміти та аналізувати соціальні та економічні процеси в суспільстві.

Мистецькі дисципліни, включають музику, мистецтво, театр та інші творчі професії. Ці дисципліни навчають студентів розумінню мистецтва, його історії та різноманітності, а також розвивають творчі навички та здібності.

Міждисциплінарні дисципліни – об'єднують різні галузі знань, такі як інформатика, біоінженерія, когнітивна наука та інші. Ці дисципліни розширюють горизонти знань студентів, допомагаючи їм зрозуміти взаємозв'язки між різними галузями знань та їх взаємодію.

Отже, формування пулу вибірових дисциплін є складним та багатоаспектним процесом, який повинен враховувати потреби студентів, викладачів та ринку праці. Успішність такого методу залежить від добре

продуманої стратегії та ефективної взаємодії між усіма зацікавленими сторонами.

1.2 Аналіз сучасних підходів формування пулу вибіркового дисциплін в закладах вищої освіти

Розпочати можна з того, що перелік та зміст вибіркового дисциплін спрямований на поглиблення знань освітян та набуття додаткових компетентностей. Обсяг дисциплін вільного вибору студентів складає не менше 25 відсотків загальної кількості кредитів ЄКТС [9], передбачених для освітньої програми відповідного рівня вищої освіти. Також невід'ємною частиною складової освітніх програм є майнор та мейджор.

Майнор – це умовна назва блоку з чотирьох взаємопов'язаних непрофільних навчальних дисциплін підготовки освітнього ступеня бакалавр. Сутність майнора полягає у формуванні певних компетентностей у здобувачів, які дозволяють розширити його професійну підготовку в нестандартних напрямках і таким чином підвищити конкурентоздатність випускників на ринку праці [10].

Мейджор – це дисципліна, що обирається здобувачем вищої освіти бакалаврського рівня з пулу спеціальності та сприяє формуванню індивідуальної освітньої траєкторії. Мейджор дозволяє здобувачу вищої освіти поглибити професійні знання та здобути додаткові не фахові навички [11].

У кожного навчального закладу є власні особливості формування пулу вибіркового дисциплін тому розглянемо одразу декілька для кращого розуміння, серед них: метод автоматизованого формування пулу на основі аналізу даних про успішність студентів, метод формування пулу на основі експертної оцінки, метод формування пулу на основі аналізу потреб ринку праці, метод формування пулу на основі стандартів освіти [12].

Метод на основі аналізу результатів тестів або інших форм контролю знань студентів. Цей метод передбачає аналіз результатів, які були отримані студентами на попередніх етапах навчання, з метою визначення їхнього рівня знань з окремих дисциплін.

На основі результатів аналізу викладач може сформувати список дисциплін, які відповідають рівню знань студентів. Ці дисципліни можуть бути зібрані у пул вибіркового дисциплін, з якого студенти можуть обирати ті, які їм цікаві або необхідні для отримання більш глибоких знань в конкретній області.

Однією з переваг цього методу є те, що він базується на об'єктивних даних про рівень знань студентів, а не на суб'єктивних оцінках викладача. Крім того, використання результатів тестів або інших форм контролю знань може допомогти викладачу виявити проблемні моменти в навчанні і зробити певні корективи до процесу навчання.

Однак, цей метод має певні недоліки. Наприклад, він не забезпечує гнучкість індивідуального навчання, оскільки студентам запропоновано лише пул вибіркового дисциплін, що відповідає їхньому рівню знань. Крім того, даний метод не враховує інтересів та побажань студентів, які можуть бути важливими для формування індивідуальної освітньої траєкторії.

Метод формування пулу на основі аналізу потреб ринку праці та вимог до фахівців полягає в тому, що в процесі формування пулу вибіркового дисциплін здійснюється аналіз потреб ринку праці та вимог до фахівців, що дозволяє враховувати сучасні тенденції в розвитку ринку праці та підготовки кваліфікованих кадрів.

Для реалізації цього методу необхідно провести дослідження ринку праці та вивчити вимоги до фахівців у конкретній галузі або професії. На основі отриманих даних формується перелік дисциплін, які повинні бути вивчені студентами для забезпечення їхньої конкурентоспроможності на ринку праці.

Перевагами даного методу є відповідність підготовки фахівців до потреб ринку праці, що дозволяє студентам отримувати актуальні знання та навички, а також збільшує їх шанси на успішне працевлаштування після завершення навчання. Однак, недоліком даного методу є те, що він може бути залежний від конкретного регіону або ринку праці, тому результати дослідження не завжди можуть бути універсальними для всіх закладів вищої освіти.

Метод формування пулу на основі стандартів освіти використовується для забезпечення відповідності підбору дисциплін вимогам освітньої програми.

Освітні стандарти визначають компетентності, які повинен мати випускник, та навички, які потрібно набути для досягнення цих компетентностей.

Метод передбачає складання пулу з тих дисциплін, які відповідають компетентностям і навичкам, визначеним стандартами. Цей метод використовується для забезпечення відповідності навчального плану та програмі навчання відповідному стандарту.

Для застосування методу потрібно провести аналіз освітніх стандартів та визначити вимоги до компетентностей та навичок студентів. Далі необхідно відібрати дисципліни, які відповідають цим вимогам та скласти з них пул.

Одним з переваг цього методу є забезпечення відповідності вимогам освітньої програми та освітнього стандарту. Проте, недоліком може бути обмеженість вибору дисциплін, які відповідають вимогам стандарту, а також відсутність гнучкості в побудові навчального плану.

У результаті аналізу сучасних підходів до формування пулу вибіркового дисциплін в закладах вищої освіти було виявлено, що вони мають свої переваги та недоліки. Метод формування пулу на основі стандартів освіти обмежений тим, що не враховує індивідуальні потреби студентів. Метод формування пулу на основі аналізу потреб ринку праці та вимог до фахівців обмежений тим, що вимоги до фахівців можуть змінюватись з часом.

Саме тому розроблення автоматизованої системи для формування пулу вибіркового дисциплін методом генетичного алгоритму значно зменшить витрати часу та зусиль та буде орієнтуватись на врахування індивідуальних потреб здобувачів освіти.

1.3 Оптимізація за генетичними алгоритмами

Зараз життя людей влаштовано так, що при вирішенні поставлених задач, майже завжди мета полягає в тому, щоб знайти нове оптимальне рішення або оптимізувати вже існуюче.

Оптимізація – це процес покращення чогось. Оптимізація передбачає пошук значень, серед вхідних даних, таким чином, щоб отримати найкращі

вихідні значення [13]. Визначення «найкращого» варіюється від проблеми до проблеми, але в математичних термінах означає максимізацію або мінімізацію однієї або кількох цільових функцій шляхом зміни вхідних параметрів. Набір усіх можливих рішень або значень, які можуть прийняти вхідні дані, становить простір пошуку. У цьому просторі пошуку лежить точка або набір точок, які дають оптимальне рішення. Мета оптимізації – знайти цю точку або набір точок у просторі пошуку.

Природа завжди була великим джерелом натхнення для всього людства. Генетичний алгоритм (ГА) – це метод оптимізації на основі пошуку, заснований на принципах генетики та природного відбору [14]. Він часто використовується для пошуку оптимальних або близьких до оптимальних рішень складних проблем, на вирішення яких в іншому випадку знадобилося б усе життя. Часто використовується для вирішення проблем оптимізації, у дослідженнях і машинному навчанні. ГА є підмножиною набагато більшої галузі обчислень, відомої як еволюційні обчислення [15]. ГА були розроблені Джоном Холландом та його студентами та колегами з Мічиганського університету, особливо Девідом Голдбергом, з того часу ГА випробували на різноманітних проблемах оптимізації, з досить високою результативністю.

Для реалізації ГА необхідно виконати такі кроки:

- 1) випадкове створення початкової популяції;
- 2) визначення функції пристосованості;
- 3) обчислення пристосованості кожного окремого індивідуума та середньої пристосованості популяції;
- 4) вибір двох осіб з поточної популяції для реалізації функції схрещування;
- 5) реалізація мутації із заданими ймовірностями та формування генотипу нащадків;
- 6) визначення кількості індивідуумів для виключення їх з популяції, щоб розмір популяції залишався постійним;
- 7) перевизначення індивідуальної та середньої пристосованості популяції;

8) продовження розвитку протягом поколінь, доки критерій зупинки не буде досягнутий.

Генетичні алгоритми достатньо рандомізовані за своєю природою, але працюють набагато краще, оскільки використовують історичну інформацію [16]. Також можна згадати і про інші переваги ГА, що роблять вказаний алгоритм таким популярним, до них належать:

- використовує цільову функцію і не вимагає жодної похідної чи додаткової інформації;
- працює із закодованою формою значень;
- швидший і ефективніший в порівнянні з традиційними методами;
- чудово працює в паралельному режимі;
- ймовірнісні правила відбору;
- оптимізує як безперервні, так і дискретні функції, а також багатоцільові проблеми;
- надає список «хороших» рішень, а не лише одне рішення;
- працює, коли область пошуку нескінчена, змінюється протягом часу, має безліч локальних оптимумів;
- низька ймовірність знаходження неправильного розв'язку;
- чітка обмеженість точності за часом в залежності від необхідної точності обчислень.

Серед обмежень та недоліків ГА можна відзначити:

- не є оптимальним рішенням для всіх проблем, особливо для задач з похідною інформацією;
- значення оптимізації функції обчислюється неодноразово, що може збільшити загальну складність алгоритму;
- будучи стохастичним, немає гарантій щодо оптимальності чи якості рішення;
- неможливо створити універсальний код, що описує функцію та критерій оптимізації, оскільки початкові умови завжди різні.

Все ж основною перевагою ГА залишається здатність швидко знаходити «достатньо хороше» рішення. Це робить генетичні алгоритми привабливими для використання, при вирішенні багатьох задач оптимізації. Існує великий набір NP-складних задач [17], це означає, що навіть найпотужнішим обчислювальним системам потрібна значна кількість часу, щоб вирішити подібного роду проблему. У такому сценарії ГА виявляється ефективним інструментом для надання оптимальних рішень за короткий проміжок часу.

Гرادієнтні методи – чисельні методи рішення з допомогою градієнта задач, що зводяться до знаходження екстремумів функції [18]. Такого роду методи працюють починаючи з випадкової точки та рухаються у напрямку градієнта. Цей метод ефективний і дуже добре працює для цільових функцій з одним піком. Проте в більшості прикладних задач виникає проблема, яку називають ландшафтами, ландшафти складаються з багатьох вершин і долин, саме тому, такого роду чисельні методи страждають від властивої тенденції застрягання на локальних оптимумах як показано на рисунку 1.1.



Рисунок 1.1 – Застрягання значення цільової функції на локальному оптимумі

Використання ГА сьогодні надзвичайно різноманітне, їх використовують при вирішенні багатьох наукових та технічних проблемах, ось лише незначний перелік галузей в яких використовується ГА: проектування нейронних мереж, вирішення фінансово-економічних задач, управління роботами, створення різних обчислювальних структур, моделювання «штучного життя», моделювання управління виробництвом, вирішення технічно-конструкторських завдань.

Отже, в багатьох випадках ГА виявляються більш ефективними, ніж традиційні алгоритми та методи. З часом зі збільшенням обчислювальної потужності комп'ютерів ГА стане ще більш ефективним алгоритмом пошуку.

1.4 Аналіз існуючих програмних рішень

Сьогодні сфера освіти перебуває у стані реформування. Саме тому багато навчальних закладів почали впроваджувати різноманітне програмне забезпечення, що могло б допомогти, як здобувачам освіти, так і викладачам.

Такого роду процеси називають цифровою трансформацією, тобто переведення будь-якого різновиду інформації в цифрову форму з використанням цифрових технологій, задля спрощення доступу до необхідних інформаційних ресурсів.

Зараз цифрова трансформація є одним з найпопулярніших трендів сучасності та радикальним чином перетворює сутнісні характеристики процесів усіх сфер суспільного життя. Сьогодні вже неможливо для закладу освіти залишатися осторонь цифрової трансформації. Цифрові технології стали невід'ємною частиною нашого повсякденного життя, тому спроба уникати їх використання означає «втечу» від реальності.

Розглянемо програмний застосунок для формування освітньої траєкторії здобувачів вищої освіти [19]. Зазначена інформаційна система передбачає застосування методу повного перебору для оптимізації індивідуальної освітньої траєкторії. Основна мета полягає в тому, щоб включити до траєкторії тільки ті дисципліни, компетентності яких є унікальними, і збільшити їх кількість на

максимально можливе значення. Головна проблема полягає в тому, що при збільшенні кількості дисциплін, які потрібно розглянути, кількість можливих комбінацій експоненційно зростає, що робить метод неефективним для реалізації в більш складних випадках. Таким чином, для використання методу повного перебору потрібно значно обмежувати кількість дисциплін або використовувати його тільки в досить простих випадках (рисунок 1.2).

Про ресурс Побудова траєкторії

Як будується індивідуальна освітня траєкторія?

Індивідуальна освітня траєкторія будується за допомогою списку обов'язкових дисциплін, а також за допомогою списку вибіркових дисциплін. Обидва списки визначені довільним шляхом заради демонстрації роботи методу побудови ІОТ, який полягає у формуванні набору дисциплін таким чином, щоб в кінці були охоплені всі необхідні компетентності.

Перед натисканням на кнопку "Сформувати траєкторію", оберть параметер цільової функції.

Max ЗК Max СК

Сформувати траєкторію

Дисципліни індивідуальної освітньої траєкторії			
Назва Предмету	Кредити	Спеціальні Компетентності	Загальні Компетентності
Вища математика	15	СК1 СК2	ЗК1 ЗК3
Фізика	5	СК3	ЗК2
Об'єктно-орієнтоване програмування	6	СК4	ЗК4 ЗК5
Теорія алгоритмів	5	СК5 СК6	ЗК6
Нові об'єктно-орієнтованого програмування	5	СК7	ЗК7 ЗК8
Операційні системи	5	СК14	ЗК10 ЗК11
Технічна механіка	5	СК13 СК15	ЗК13 ЗК14

Рисунок 1.2 – Сформована індивідуальна освітня траєкторія [19]

Отже формування пулу вибіркового дисциплін за допомогою методу генетичного алгоритму є більш ефективним підходом порівняно з іншими методами, оскільки дозволяє знайти оптимальний набір дисциплін з урахуванням багатьох параметрів і обмежень. Крім того, генетичний алгоритм забезпечує швидке та точне рішення, що є особливо важливим для великих навчальних закладів з великою кількістю студентів і різноманітними вимогами до навчання. Саме тому, розробка застосунку на основі генетичного алгоритму, для підвищення ефективності процесу формування пулу вибіркового дисциплін в закладах вищої освіти, є надзвичайно актуальною.

1.5 Мета, задачі та вимоги до реалізації інформаційної системи

Метою кваліфікаційної роботи бакалавра є створення методу формування пулу вибірових навчальних дисциплін на основі генетичного алгоритму, що в свою чергу дозволить оптимізувати навчання студентам вищих навчальних закладів. Тому для виконання поставленого завдання необхідно:

- провести аналіз методів формування пулу вибірових дисциплін у вищих навчальних закладах;
- провести аналіз генетичних алгоритмів, їх різновидів та можливості імплементації у інформаційну систему;
- побудувати інформаційну модель методу формування пулу вибірових дисциплін за критеріями зацікавленості у співпраці здобувача освіти з конкретним викладачем та рівнем особистого інтересу до дисципліни.
- виконати алгоритмічну та програмну реалізацію методу формування пулу вибірових дисциплін у вищих навчальних закладах.

Розділ 2 Спосіб формування пулу вибірових навчальних дисциплін на основі генетичного алгоритму

2.1 Загальні положення щодо складу вибірових дисциплін у закладах вищої освіти

Склад вибірових дисциплін у закладах вищої освіти визначається навчальним планом спеціальності та вимогами до її здобуття. Вибіркові дисципліни включають різноманітні курси, що допоможуть студентам поглибити знання та навички у відповідних галузях знань.

Вибіркові дисципліни мають бути пов'язані з основними напрямками та спеціалізаціями навчання, а також відповідати профілю та вимогам ринку праці. Так, у закладах, які надають освіту в сфері технічних наук, вибіркові дисципліни включають курси з програмування, інформаційних технологій, електроніки тощо. У закладах, які надають освіту в сфері гуманітарних наук, вибіркові дисципліни включають курси з літератури, мовознавства, історії тощо.

У більшості випадків, заклади вищої освіти включають в програму вибіркові дисципліни з різних галузей знань, щоб забезпечити студентам можливість отримати різноманітну освіту (рисунок 2.1). Зокрема, вибіркові дисципліни можуть включати:

- Дисципліни, що спрямовані на розвиток загальних компетенцій, наприклад, лідерства, комунікаційних навичок, креативності тощо.
- Дисципліни, що спрямовані на розвиток певних професійних компетенцій, які можуть бути корисними для майбутньої кар'єри студента.
- Дисципліни, що допомагають студентам зрозуміти та аналізувати глобальні проблеми та виклики, наприклад, екологічні проблеми, глобалізацію, розвиток технологій тощо.
- Дисципліни, що допомагають студентам зрозуміти та аналізувати різноманітні культурні аспекти.



Рисунок 2.1 – Схема різновидів вибірових дисциплін

У загальному розумінні, склад вибірових дисциплін може бути доволі різноманітним, але важливо дотримуватись прагнень здобувача вищої освіти що до результатів навчання, які визначаються спеціалізованими дисциплінами, що доповнюють обов'язкові. Загальноосвітні дисципліни допомагають студентам отримати широке загальне освітнє підґрунтя, вивчити загальні принципи та закономірності, розвинути критичне мислення та аналітичні навички. Спеціалізовані дисципліни допомагають студентам засвоїти конкретні знання та навички, необхідні для роботи в певних професійних галузях.

Важливо зазначити, що вибірові дисципліни не повинні ставити студентів у незручне становище або перевантажувати їх навчання. Заклади вищої освіти повинні забезпечувати належну підтримку та консультації, щодо вибору вибірових дисциплін, а також забезпечувати рівномірне розподілення навантаження на студентів.

Окрім того, важливо забезпечити можливість студентам самостійно скласти програму навчання з вибірових дисциплін, які їм цікаві та відповідають їхнім професійним цілям та інтересам.

Також можуть бути запроваджені певні обмеження на формування пулу вибірових навчальних дисциплін. Наприклад, деякі дисципліни можуть мати

передумови в знаннях, тому студент повинен спочатку пройти певний набір обов'язкових дисциплін.

Крім того, важливо враховувати потреби ринку праці та забезпечити вибір таких дисциплін, які допоможуть студентам здобути необхідні знання та навички для успішної кар'єри в майбутньому.

Зазвичай, склад вибірових дисциплін у закладах вищої освіти визначається у співпраці з представниками різних галузей промисловості та бізнесу, які надають рекомендації щодо того, які дисципліни повинні бути включені до програми навчання.

Загалом, склад вибірових дисциплін у закладах вищої освіти має бути добре продуманим та забезпечувати студентам можливість отримання необхідних знань та навичок для подальшого розвитку в обраній професійній галузі, а також розвиток загального рівня освіти та критичного мислення.

2.2 Послідовність формування переліку вибірових дисциплін закладами вищої освіти

Послідовність формування переліку вибірових дисциплін варіюється в залежності від освітньої програми та вимог, які ставляться до студентів. Однак, в загальному випадку, така послідовність виглядатиме наступним чином.

Заклад вищої освіти розробляє освітню програму для кожної спеціальності, яка визначає загальний перелік обов'язкових і вибірових дисциплін, що необхідні для отримання кваліфікації. Освітня програма повинна відповідати вимогам державних стандартів вищої освіти та реальним потребам ринку праці.

Формування загального переліку вибірових дисциплін здійснюється на рівні закладу вищої освіти та містить пропозиції всіх факультетів, кафедр та викладачів. Основну роль у формуванні набору вибірових дисциплін відіграють саме факультети (рисунок 2.2).

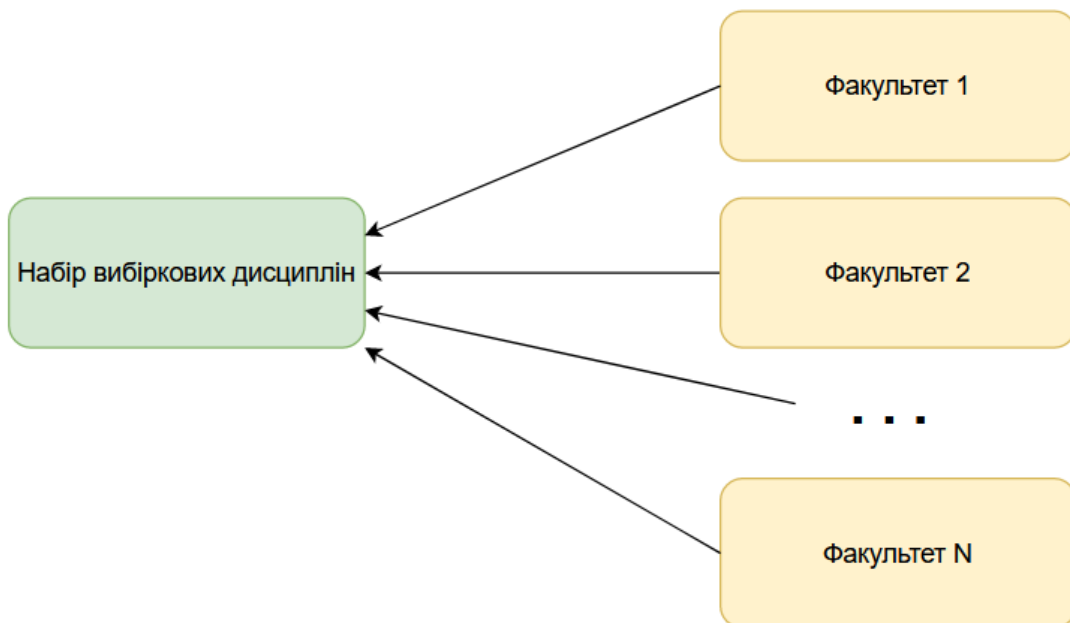


Рисунок 2.2 – Схема формування набору вибірових дисциплін

Наступною ланкою після факультетів слідують кафедри, що належать до того чи іншого факультету (рисунок 2.3).

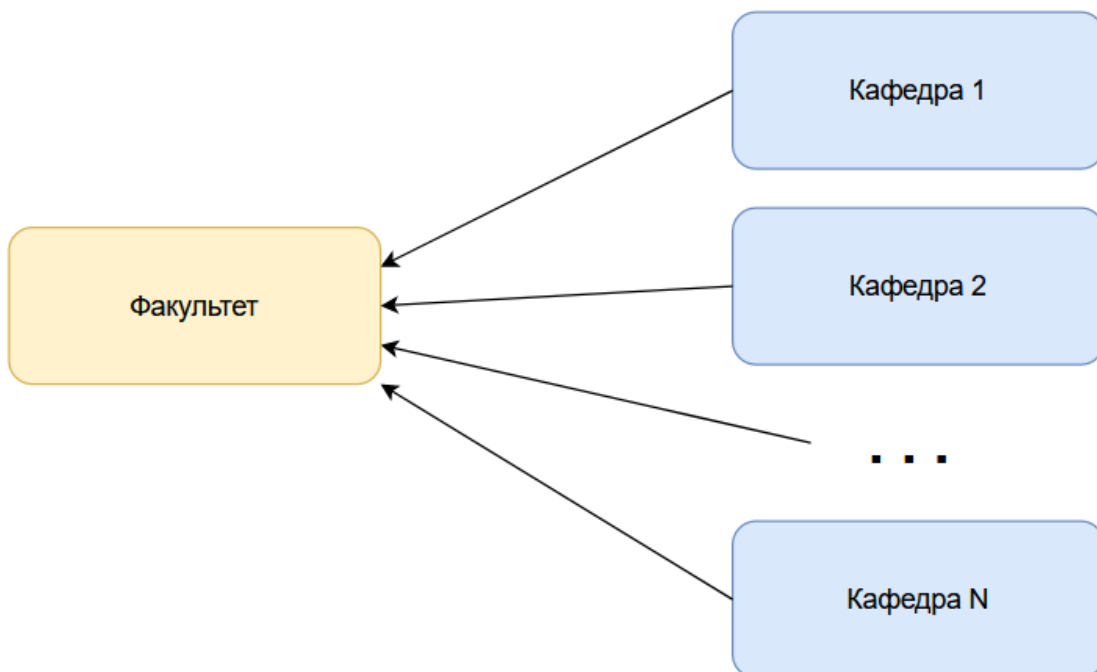


Рисунок 2.3 – Схема формування набору вибірових дисциплін на рівні факультету

У складі кафедр працюють викладачі, які й розробляють програми навчальних дисциплін, що в результаті стають складовими пулу навчальних вибірових дисциплін (рисунок 2.4).

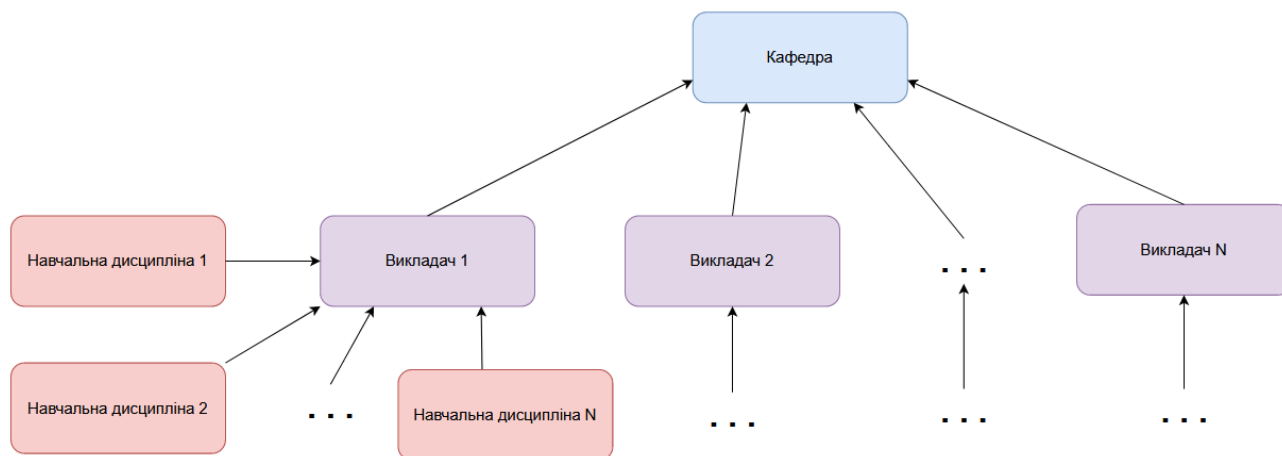


Рисунок 2.4 – Схема формування набору вибірових дисциплін на рівні кафедри

Загальний перелік вибірових дисциплін формується із дотриманням Державних стандартів та інших нормативних актів у сфері освіти та враховує рекомендації всіх груп стейкхолдерів, а саме: здобувачів освіти, роботодавців, викладачів та закладів вищої освіти (рисунок 2.5).

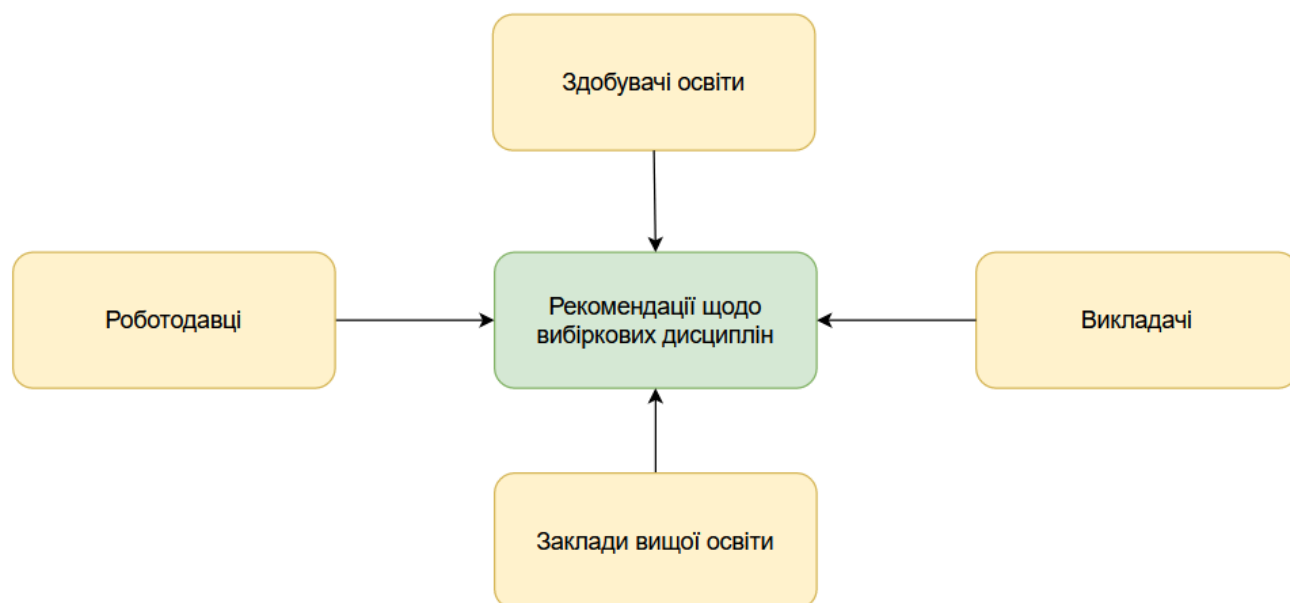


Рисунок 2.5 – Схема формування рекомендацій, щодо вибірових дисциплін

Цей перелік містить дисципліни, що сприяють розвитку здобувача вищої освіти в частині поглиблення професійних компетентностей, опанування нових професійних навичок та розвитку вже здобутих компетенцій (рисунок 2.6).

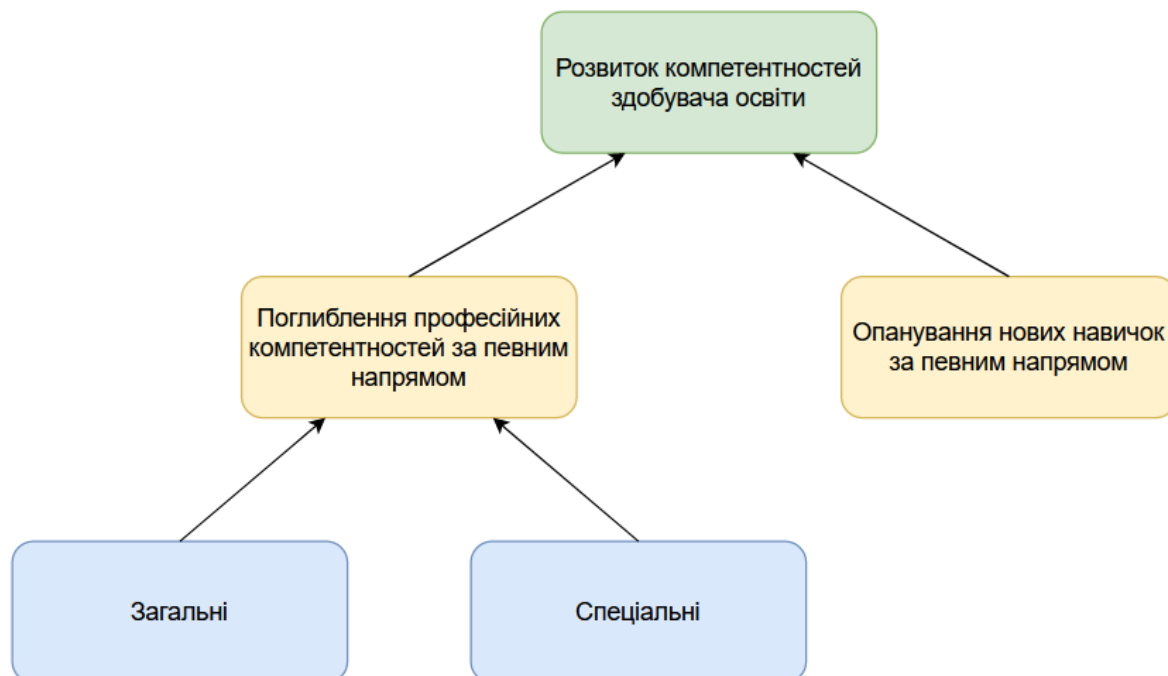


Рисунок 2.6 – Види розвитку компетентностей здобувача освіти

Загальний перелік вибірових дисциплін оприлюднюється для здобувачів вищої освіти у вигляді окремого документу, який студенти можуть знайти на веб-сайті університету або факультету. У цьому документі наведені назви дисциплін, їх опис та кількість кредитів, які можуть бути зараховані студентам за їх вивчення.

Кожен студент обирає дисципліни зі загального переліку вибірових дисциплін у процесі формування індивідуального навчального плану. ІНП – це документ, який містить перелік дисциплін, які студент повинен вивчити для отримання кваліфікації відповідного рівня. При формуванні ІНП студент може обирати дисципліни зі загального переліку вибірових дисциплін, з урахуванням вимог освітнього стандартів та власних інтересів.

Після того, як студент складає ІНП, він погоджує його з гарантом освітньої програми та куратором. На етапі погодження здобувачем вищої освіти

можуть враховуватися рекомендації куратора, роботодавців, гаранта ОПП, особисті вподобання, ЗВО (рисунок 2.7).

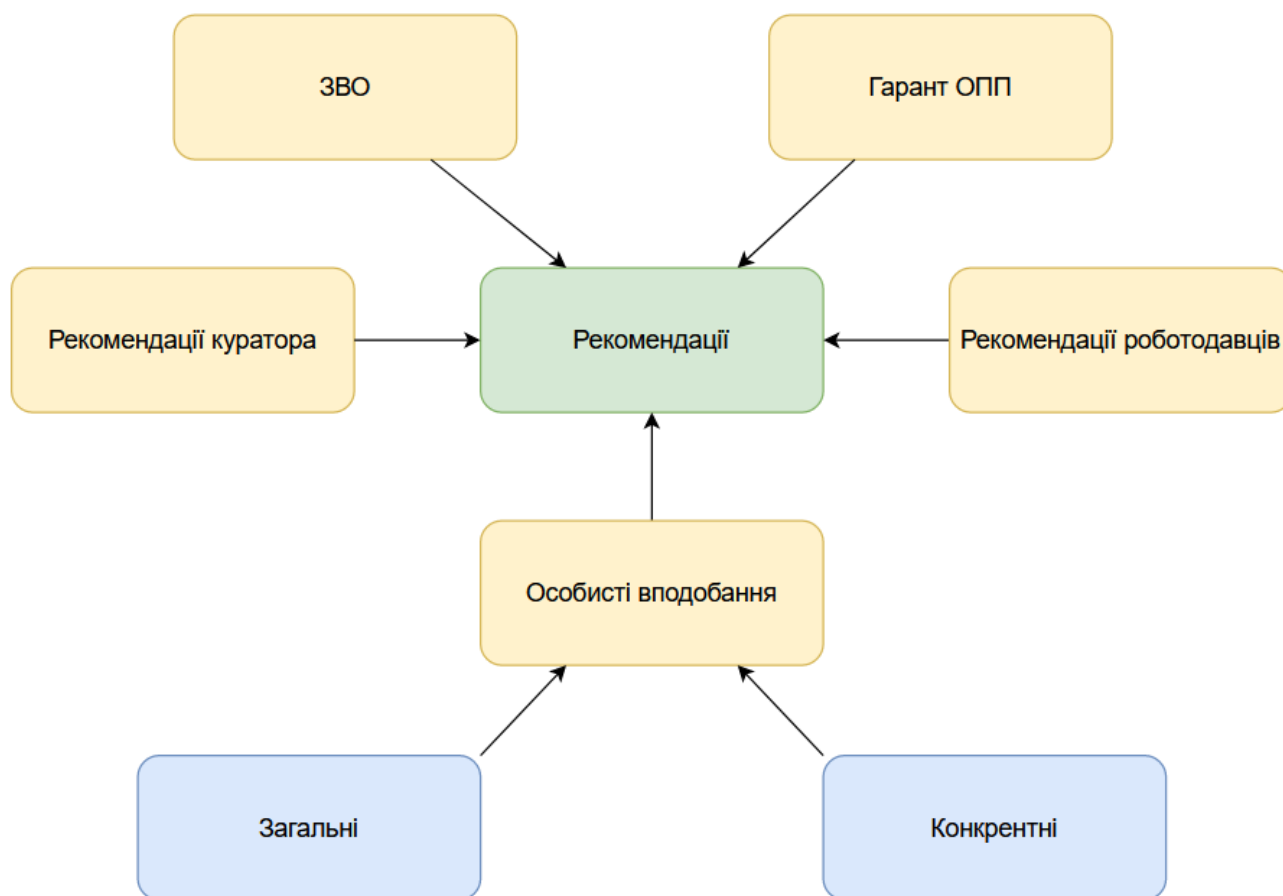


Рисунок 2.7 – Види рекомендацій, що можуть враховуватись здобувачем освіти

Після того, як студент визначився з дисциплінами, які він бажає вивчати, йому потрібно зареєструватися на ці курси. Реєстрація відбувається онлайн на платформі університету або факультету. Реєстраційний період визначається заздалегідь і має обмежений термін.

Після того, як студент зареєструвався на курси вибіркової дисципліни, він починає їх вивчати відповідно до розкладу занять. Зазвичай, вибіркові дисципліни проходять в різних форматах, таких як: лекції, семінари, практичні заняття, лабораторні роботи, тощо.

Коли студент успішно закінчує курс вибіркової дисципліни, виставляється оцінка за цю дисципліну. Зазвичай, оцінювання здійснюється на основі тестів, письмових робіт, домашніх завдань та інших видів робіт.

Зарахування кредитів залежить від оцінки, отриманої студентом за курс, та від кількості кредитів, призначеної за цей курс в ІНП студента.

Перелік вибірових дисциплін оновлюється періодично, залежно від потреб ринку праці та розвитку науки. Нові дисципліни можуть бути додані до переліку, а деякі із старих можуть бути вилучені. Це дає студентам можливість вибирати з більш широкого спектру дисциплін та забезпечує актуальність навчального плану.

Також слід зауважити, що у випадку коли кількість студентів, які обрали дисципліну, менша за мінімальну кількість, необхідну для проведення дисципліни, то заклад має право скасувати її вивчення. У такому випадку студентам, які обрали цю дисципліну, пропонують інші варіанти вибірових дисциплін.

Таким чином, правильно сформований перелік вибірових дисциплін допоможе студентам отримати необхідні знання та навички для майбутньої кар'єри, а також позитивно вплине на розвиток закладу вищої освіти.

2.3 Особливості використання генетичного алгоритму для задачі побудови бажаного пулу вибірових навчальних дисциплін для здобувача вищої освіти

2.3.1. Загальна схема оптимізаційної задачі

У багатьох задачах вимагається не просто знайти який небудь розв'язок, а підібрати серед усіх розв'язків оптимальний, тобто найкращий. Модель оптимізаційної задачі складається з таких елементів: змінні, цільова функція, критерій та обмеження (рисунок 2.8).

Змінні – це невідомі величини, значення яких потрібно знайти в результаті розв'язання задачі. В даному випадку змінні представлені у вигляді пулів вибірових дисциплін, а в результаті необхідно знайти пул вибірових дисциплін, що найкраще резонує з особистими вподобаннями здобувача освіти.

Цільова функція – величина, яка залежить від змінних і значення якої потрібно максимізувати чи мінімізувати. В контексті задачі формування пулу

вибіркових навчальних дисциплін, цільовою функцією можна вважати функцію пристосованості, що відповідає за обчислення значення пристосованості змінної до вимог, які були задані користувачем.

Критерій – це вимога мінімізації або максимізації цільової функції. Відповідно до задачі необхідно саме максимізувати цільову функцію, тобто знайти змінну з найбільшим значенням функції пристосованості.

Обмеження – це умови, яким мають задовольняти змінні. Основними обмеженнями у даному випадку є кількість кредитів ЄКТС та кількість навчальних дисциплін для одного пулу.

$$f(x) = c_{x1} + t_{x1} + \dots + c_{xn} + t_{xn} \rightarrow \max$$

$$e_1 + \dots + e_n = 60$$

Рисунок 2.8 – Модель оптимізаційної задачі

Тут змінною є x , цільовою функцією – $f(x)$, критерієм – вимога максимізації ($\rightarrow \max$), а обмеженням – $e_1 + \dots + e_n = 60$, що відповідає кількості встановлених кредитів ЄКТС.

Отже, математична модель оптимізаційної задачі враховує змінні, цільову функцію та обмеження. Змінні описують пул вибіркових дисциплін, а цільова функція визначається функцією пристосованості, яка враховує бажання користувача. Обмеження включають кількість кредитів ЄКТС та кількість навчальних дисциплін у пулі.

2.3.2. Загальна структура генетичного алгоритму

Розробка ефективного методу формування пулу вибіркових дисциплін може допомогти закладам вищої освіти покращити навчальний процес та забезпечити студентам можливість вибрати найбільш корисні та цікаві курси для

свого майбутнього розвитку. Такий метод також може сприяти підвищенню рівня освіти студентів, оскільки дозволить підвищити якість викладання та розширити спектр доступних дисциплін.

Перш ніж почати опис роботи генетичного алгоритму, важливо ознайомитися з деякою базовою термінологією, що буде використовуватись надалі.

Популяція – це множина хромосом, що використовується для пошуку оптимального розв’язку задачі. Популяція може складатися з кількох десятків або сотень індивідуумів. У даному випадку популяція складається з певної кількості пулів вибіркового дисциплін (рисунок 2.9).



Рисунок 2.9 – Приклад структури однієї популяції

Хромосома – це послідовність генетичних елементів, які кодують інформацію про характеристики індивідуума. У генетичному алгоритмі хромосома представляє кандидата на рішення задачі. В контексті задачі хромосома – це пул вибіркового дисциплін (рисунок 2.10).

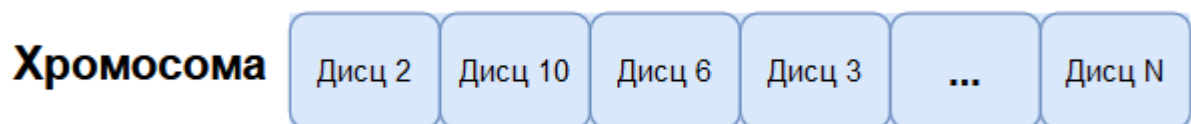


Рисунок 2.10 – Приклад структури однієї хромосоми

Ген – це окрема одиниця у складі хромосоми, яка кодує конкретну характеристику індивідуума, тобто геном. В контексті даної задачі, геном є унікальна для хромосоми навчальна дисципліна, що має наступний список

властивостей: факультет, кафедру, викладача та кількість кредитів ЄКТС (рисунок 2.11).

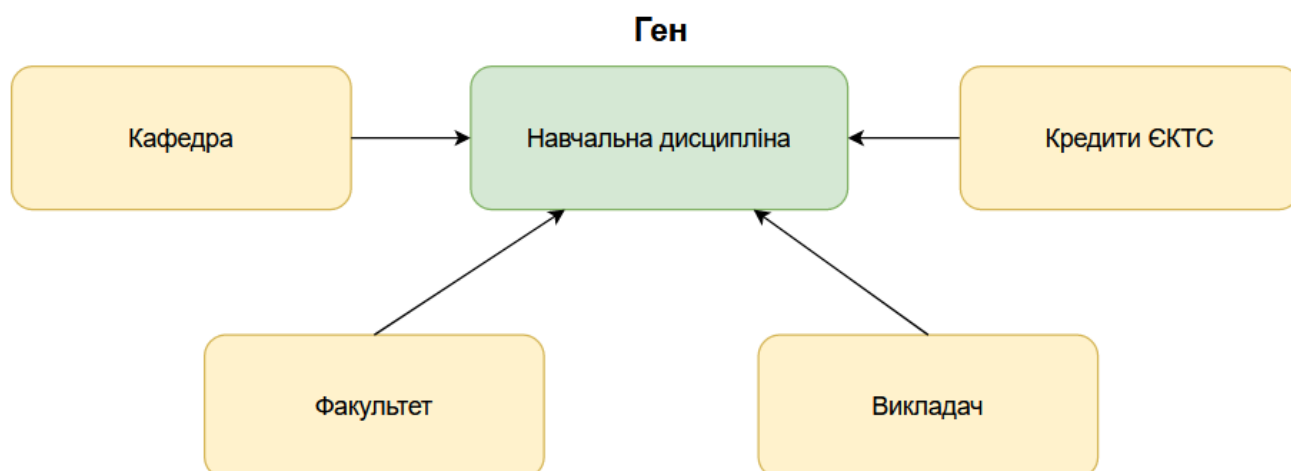


Рисунок 2.11 – Структура одного гену

Генетичний алгоритм може бути корисним для розв’язання задачі побудови бажаного пулу вибіркових навчальних дисциплін для здобувача вищої освіти. Основна ідея полягає в тому, щоб створити популяцію потенційних пулів навчальних дисциплін і використовувати еволюційний підхід, щоб знайти оптимальний пул для поданого здобувача вищої освіти (рисунок 2.12).

Створення початкової популяції у такому випадку можна здійснити за допомогою випадкового вибору дисциплін з наявного пулу. Перш за все, потрібно визначити параметри, які характеризують дисципліни, наприклад, складність, корисність, популярність тощо. Після цього можна випадковим чином вибрати кілька дисциплін, використовуючи відповідний генератор випадкових чисел.

Отримана популяція може містити дублікати дисциплін або ж бути недостатньо різноманітною. У такому випадку можна застосувати додаткові методи селекції, наприклад, метод турніру.

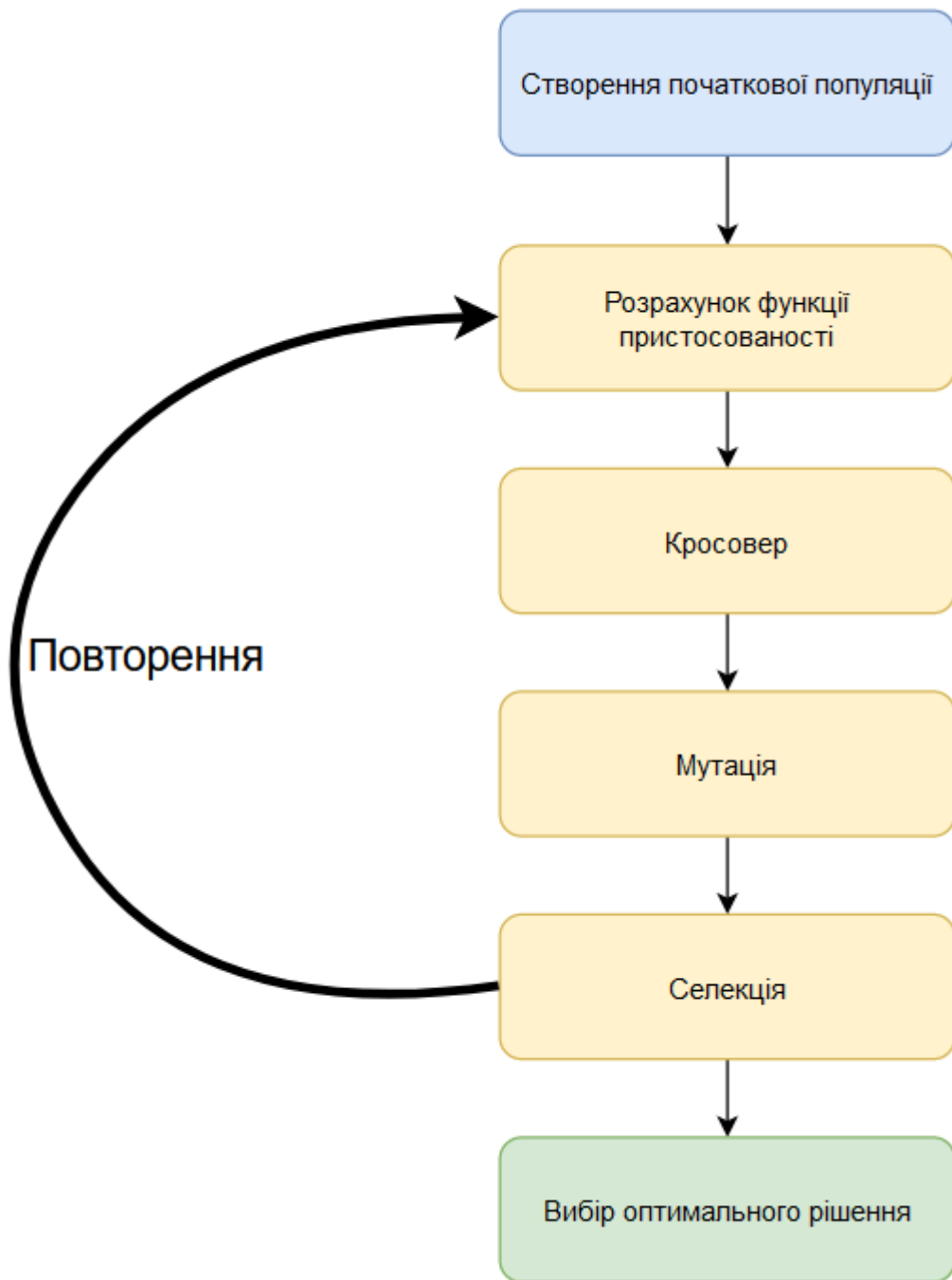


Рисунок 2.12 – Схема роботи генетичного алгоритму

Турнірний відбір є одним з методів відбору особин у генетичному алгоритмі, який полягає у тому, щоб обрати кращих особин з декількох випадково вибраних підмножин популяції. Цей метод дозволяє вибрати кращі особини з популяції, не прив'язуючись до конкретної оцінки фітнесу. Процес виконання турніру може бути описаний наступним чином:

1. Випадковим чином обирається певна кількість особин з популяції;
2. Серед вибраних особин обирається найкраща;

3. Вибрана особина стає батьківською для наступного покоління.

Кількість особин, які беруть участь у турнірі, може бути встановлена заздалегідь. Зазвичай використовують значення від 2 до 10. Збільшення кількості учасників турніру збільшує ймовірність вибору кращої особини, але при цьому збільшується обчислювальна складність (рисунок 2.13).

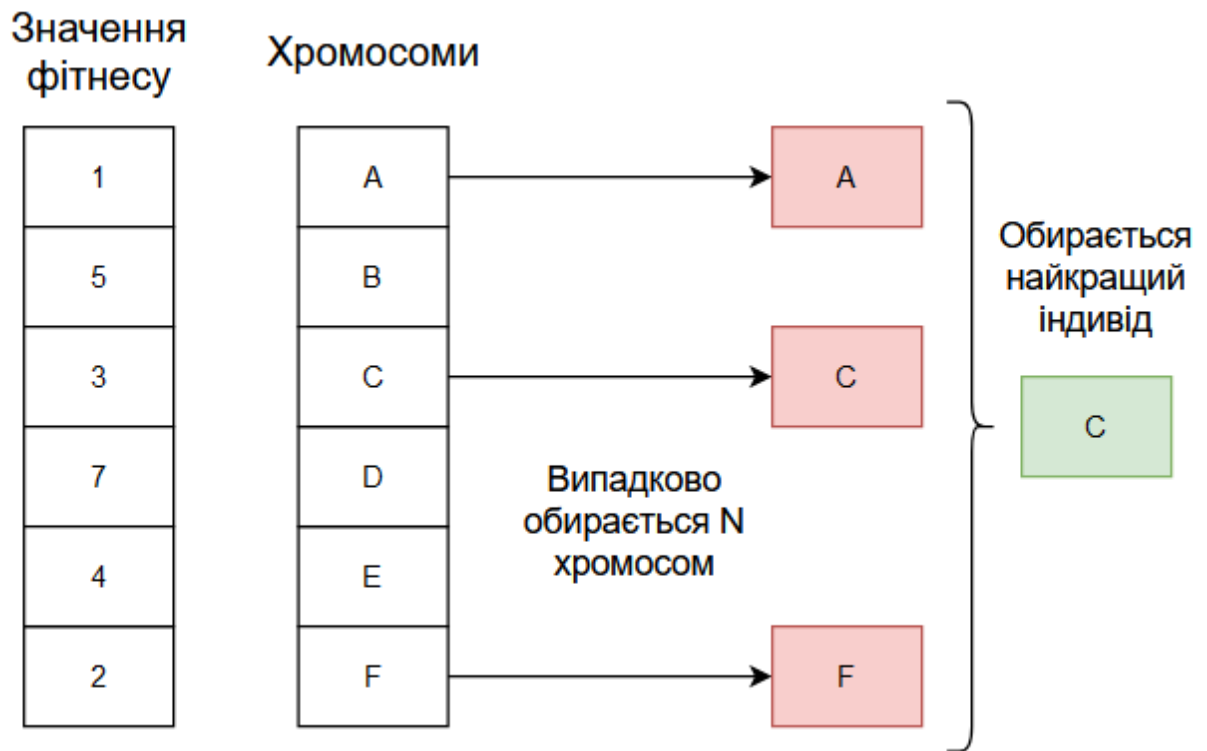


Рисунок 2.13 – Схема створення початкової популяції за допомогою методу турнірного відбору

Наступним кроком після формування початкової популяції, виконується кросинговер. Схрещування або кросинговер – це операція, яка здійснюється між двома батьківськими хромосомами і призводить до створення нової хромосоми-потомка. Цей процес полягає у випадковому виборі двох батьківських особин, з яких будуть обрані певні відрізки геномів. Далі, випадково обирається точка перетину (точка ділення геному), і генетична інформація до цієї точки береться з однієї батьківської особини, а від заданої точки і до кінця геному, з іншої

батьківської особини. Таким чином, створюються нові особини, що містять комбінації генів батьків (рисунок 2.14).

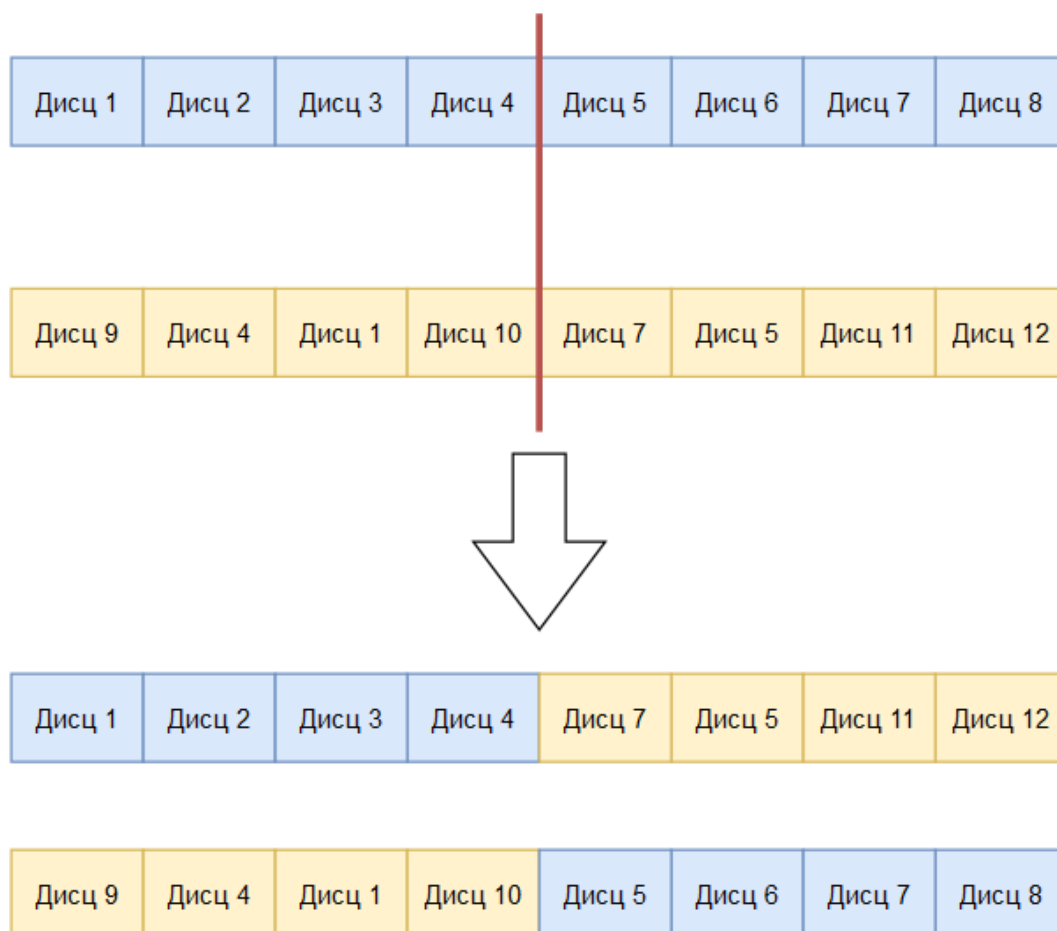


Рисунок 2.14 – Приклад роботи методу схрещування

Далі по списку необхідно створити можливість мутації для нових особин. Мутація – це процес, в результаті якого змінюється значення одного або кількох генів на хромосомі. Мутація використовується для додавання різноманітності до популяції і забезпечення можливості виявлення оптимального розв’язку, який не був би доступний за умови використання лише схрещування. Цей процес може бути здійснений різними способами, наприклад, заміною гена на інший випадковим чином, збільшенням або зменшенням значення гена, додаванням нових генів, видаленням генів, тощо (рисунок 2.15).

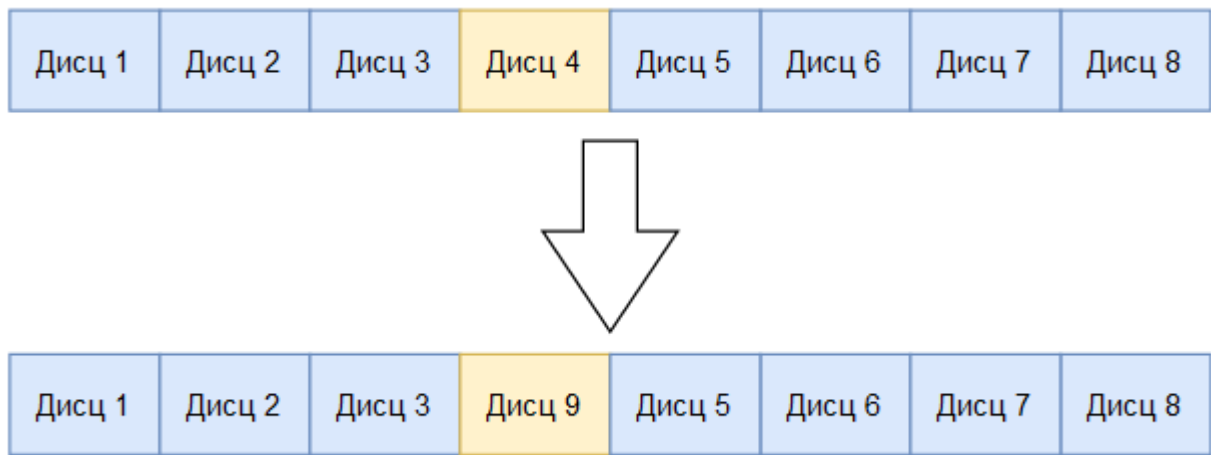


Рисунок 2.15 – Приклад роботи методу мутації

Імовірність мутації зазвичай є значно меншою, ніж імовірність кросинговеру, оскільки мутації повинні бути достатньо рідкісними, щоб зберегти різноманітність популяції, але й не дозволяти відхилитись від оптимального розв'язку.

Фітнес-функція або функція-приспосованості – це функція, яка оцінює якість кожної хромосоми у популяції. Функція пристосованості для задачі формування пулу вибірових дисциплін може бути доволі складною, оскільки повинна відображати багато різних факторів, що впливають на ефективність кожної дисципліни. Один з можливих підходів до формування функції пристосованості полягає в поєднанні різних факторів, таких як:

- вимоги законодавства та стратегії університету, щодо вибірових дисциплін;
- рівень зацікавленості, у співпраці, здобувача освіти із конкретним викладачем;
- рівень інтересу кандидата до дисципліни, що може впливати на його мотивацію та продуктивність;
- належність до певного факультету та кафедри.

Функція пристосованості може бути сформована як сума оцінок кожного з цих факторів, або як їх комбінація за допомогою певного математичного методу. Також слід зазначити, що формування функції пристосованості є ітеративним процесом, тому може бути покращеним за допомогою аналізу

результатів попередніх ітерацій та корегування ваг факторів відповідно до отриманих даних.

Розглядаючи обмеження, що обумовлені законодавством важливо відзначити те, що для отримання освітнього ступеня бакалавра необхідно успішно виконати та засвоїти навчальну програму, обсягом 240 кредитів ЄКТС, 25% яких складають вибіркові дисципліни. Отже кількість годин, сформованого пулу вибіркових дисциплін, повинна дорівнювати 1800 або 60 кредитам ЄКТС (рисунок 2.16).

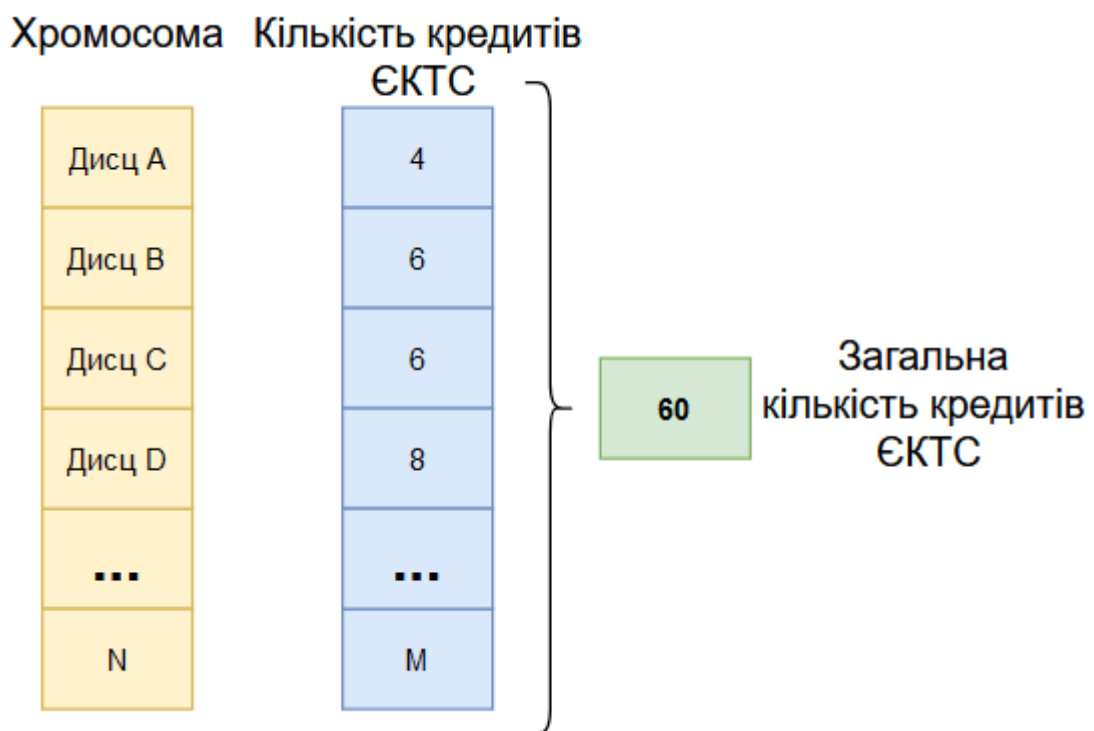


Рисунок 2.16 – Обмеження функції пристосованості

Отже, можна підсумувати, що за допомогою генетичного алгоритму можна знайти оптимальний пул навчальних дисциплін, які задовольняють вимоги здобувача вищої освіти та допоможуть досягти навчальних цілей. Однак, слід враховувати те, що генетичний алгоритм може зайняти значну кількість обчислювальних ресурсів, тому даний алгоритм слід застосовувати з обережністю.

2.4 Функціональна структура інформаційної системи

Інформаційна система для формування пулу вибіркових навчальних дисциплін призначена для автоматизації роботи таких користувачів, як: студент та адміністратор.

Система повинна реалізовувати функції реєстрації та авторизації користувача, а також надати можливість для керування особистими даними (рисунок 2.17). Процес роботи з особистими даними проходить наступним чином:

1. Реєстрація на сайті, використовуючи інтерфейс сторінки «Реєстрація».
2. Авторизація на сайті, використовуючи інтерфейс сторінки «Авторизація».
3. Робота з особистими даними на сторінці «Особисті дані», де користувач може переглядати, додавати, редагувати та видаляти дані.

Також система повинна надавати користувачу можливість працювати з внутрішніми даними системи (рисунок 2.18).

До внутрішньої складової належать основні сутності бази даних, такі як: факультети, кафедри, викладачі та дисципліни. Процес роботи з внутрішніми даними проходить наступним чином:

1. Користувач відкриває одну з відповідних сторінок, що відносяться до внутрішніх даних.
2. За допомогою інтерфейсу сторінки користувач переглядає, додає, редагує та видаляє внутрішні дані системи.

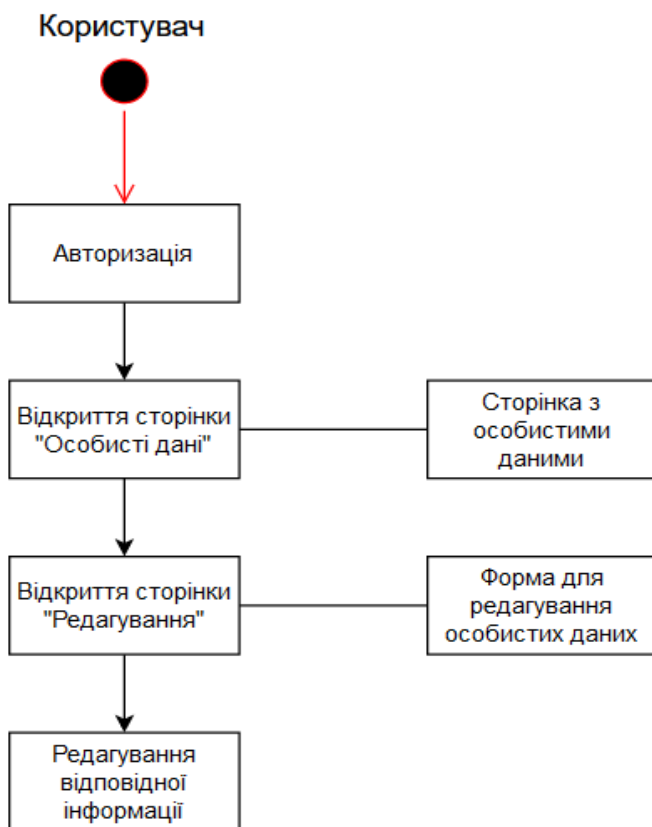


Рисунок 2.17 – Діаграма дій для роботи з особистими даними

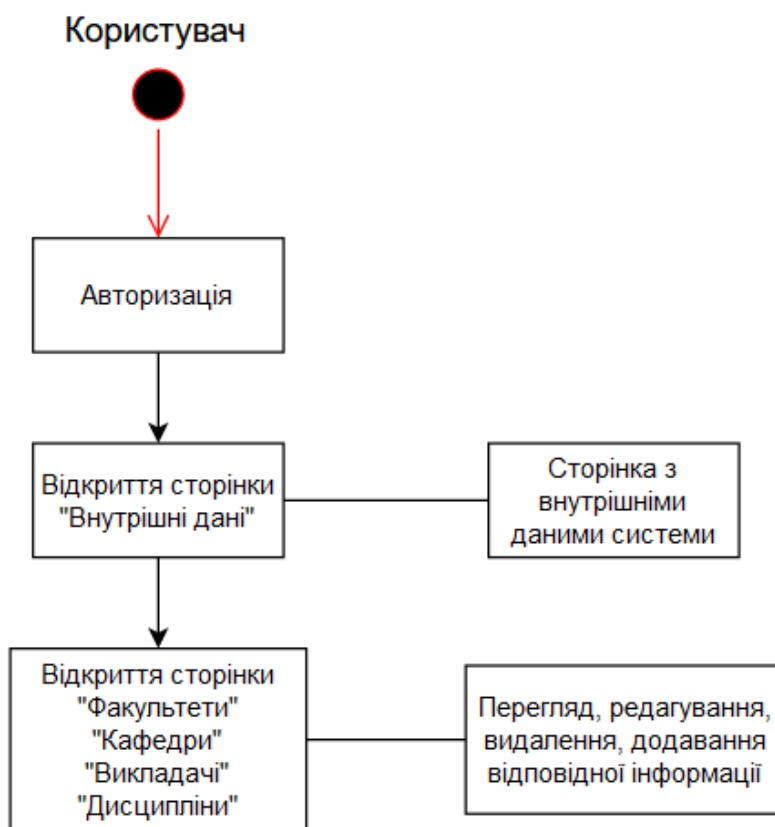


Рисунок 2.18 – Діаграма дій для роботи з внутрішніми даними системи

Все ж, головною функцією системи є допомога користувачу у формуванні індивідуального пулу вибіркових дисциплін (рисунок 2.19). Ця підсистема аналізує побажання користувача, щодо викладачів та освітніх сфер, і надає найбільш оптимальний пул вибіркових дисциплін. Для роботи з даною системою необхідно:

1. Зареєструватися та авторизуватися у системі.
2. Перейти на сторінку «Вибіркові дисципліни».
3. Проходження опитування для визначення освітніх вподобань користувача.
4. Система проаналізує введені дані та згенерує індивідуальний пул вибіркових дисциплін.
5. Збереження обраних дисциплін.

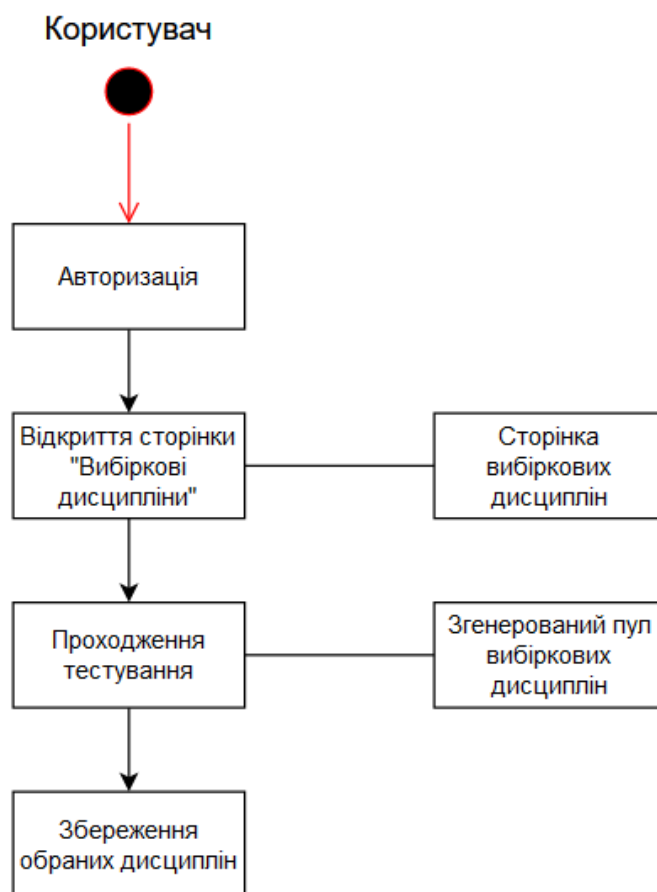


Рисунок 2.19 – Діаграма дій для роботи з системою формування індивідуального пулу вибіркових дисциплін

Маючи зазначену функціональну структуру, система зможе забезпечити ефективну роботу з особистими даними користувачів, внутрішніми даними системи та системою для створення пулів вибірових дисциплін. Забезпечення цих функцій позитивно вплине на користувацький досвід. Отже, розробка та впровадження такої системи може бути важливим кроком у поліпшенні якості сфери освіти та підвищення рівня задоволення здобувачів освіти.

2.5 Проектування структури інформаційної системи формування пулу вибірових навчальних дисциплін для здобувача вищої освіти

Інформаційна структура системи передбачає, що дані будуть перенесені із бази даних на сервер, де будуть проводитись обчислення для формування необхідних пулів вибірових навчальних дисциплін на основі даних запиту, потім – з сервера до клієнта. Схему цієї інформаційної структури зображено на (рисунок 2.20).

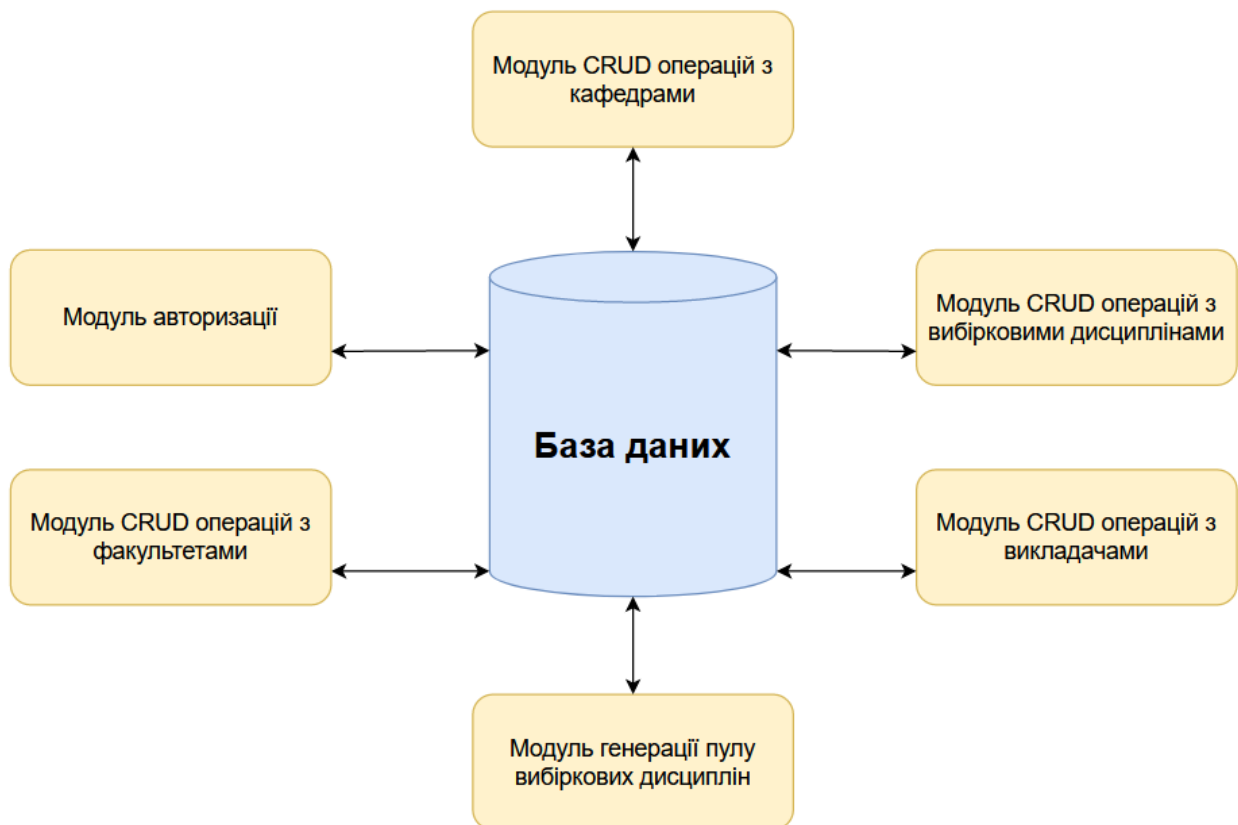


Рисунок 2.20 – Інформаційна структура системи

База даних є одним з ключових модулів системи, який забезпечує збереження та організацію даних, необхідних для роботи системи. База даних містить інформацію про студентів, їх вибір дисциплін, доступні дисципліни та іншу важливу інформацію, яка потрібна для правильного функціонування системи. На рисунку 2.21 зображена даталогічна модель бази даних, яка реалізує такі таблиці: Roles (ролі), Users (користувачі), Faculty (факультет), Chairs (кафедри), Teachers (викладачі), Disciplines (дисципліни), Pools of optional disciplines (згенеровані пули вибіркового навчальних дисциплін).

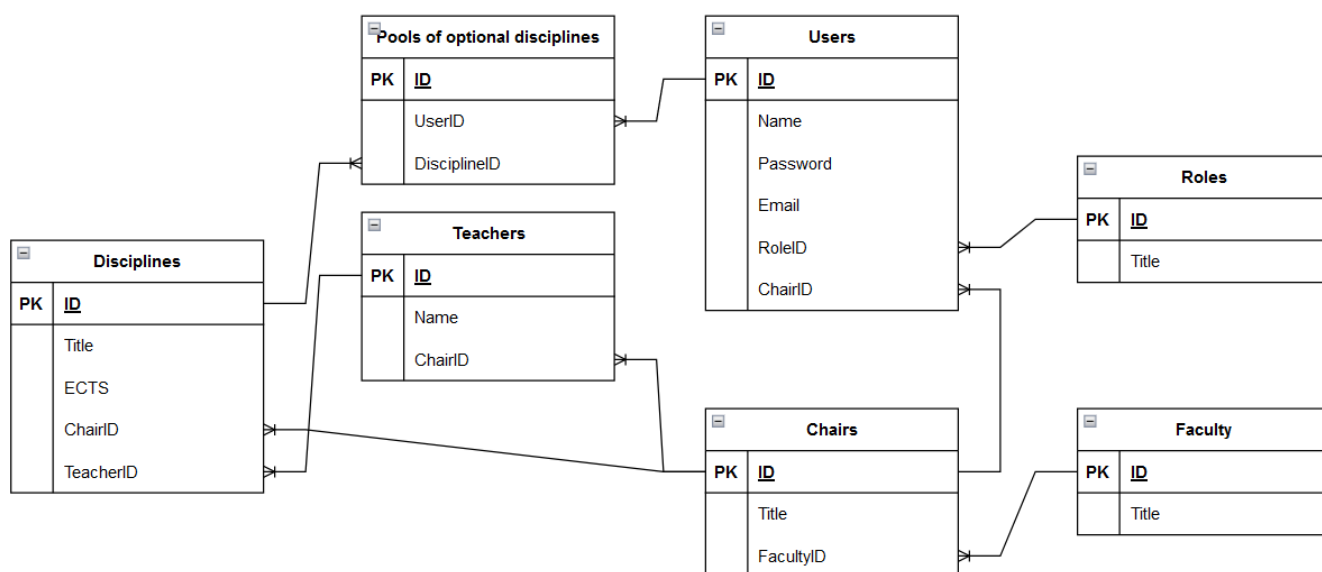


Рисунок 2.21 – Даталогічна модель бази даних

Таблиця «Users» призначена для збереження даних користувачів та має такі атрибути: Id, Name, Email, Password (таблиця 2.1).

Таблиця 2.1 – Атрибути таблиці «Users»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	Id	int	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	Name	nvarchar(255)	Ім'я користувача
3.	Email	nvarchar(255)	Електронна пошта користувача
4.	Password	nvarchar(255)	Пароль користувача
5.	RoleID	int	Вторинний ключ (роль)
6.	ChairID	int	Вторинний ключ (кафедра)

Таблиця «Roles» призначена для збереження даних про системні ролі та має такі атрибути: Id, Title (таблиця 2.2).

Таблиця 2.2 – Атрибути таблиці «Roles»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	Id	int	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	Title	nvarchar(255)	Назва системної ролі

Таблиця «Faculty» призначена для збереження даних про факультети та має такі атрибути: Id, Title (таблиця 2.3).

Таблиця 2.3 – Атрибути таблиці «Faculty»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	Id	int	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	Title	nvarchar(255)	Назва факультету

Таблиця «Chairs» призначена для збереження даних про кафедри та має такі атрибути: Id, Title, FacultyID (таблиця 2.4).

Таблиця 2.4 – Атрибути таблиці «Chairs»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	Id	int	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	Title	nvarchar(255)	Назва кафедри
3.	FacultyID	int	Вторинний ключ (факультет)

Таблиця «Teachers» призначена для збереження даних викладачів та має такі атрибути: Id, Name, ChairID (таблиця 2.5).

Таблиця 2.5 – Атрибути таблиці «Teachers»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	Id	int	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	Name	nvarchar(255)	Ім'я викладача
3.	ChairID	int	Вторинний ключ (кафедра)

Таблиця «Disciplines» призначена для збереження даних дисциплін та має такі атрибути: Id, Title, ECTS, ChairID, TeachersID (таблиця 2.6).

Таблиця 2.6 – Атрибути таблиці «Disciplines»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	Id	int	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	Title	nvarchar(255)	Назва дисципліни
3.	ECTS	int	Кількість кредитів ECTS
4.	ChairID	int	Вторинний ключ (кафедра)
5.	TeachersID	int	Вторинний ключ (викладач)

Таблиця «Pools of optional disciplines» призначена для збереження даних сформованих пулів вибіркового навчальних дисциплін та має такі атрибути: Id, UserID, DisciplineID (таблиця 2.7).

Таблиця 2.7 – Атрибути таблиці «Pools of optional disciplines»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	Id	int	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	UserID	int	Вторинний ключ (користувач)
3.	DisciplineID	int	Вторинний ключ (дисципліна)

У результаті проектування структури інформаційної системи для формування пулу вибіркового навчальних дисциплін для здобувача вищої освіти було розроблено комплексну систему. Були спроектовані модулі формування вхідних моделей, обчислення генетичного алгоритму та генерації пулів вибіркового навчальних дисциплін. Дана структура проекту дозволить забезпечити здобувачам вищої освіти можливість отримати оптимальний набір навчальних дисциплін для їхньої майбутньої професійної діяльності, зменшуючи час та зусилля, необхідні для ручного відбору дисциплін. Така система може стати корисним інструментом для викладачів та методистів, які мають завдання з формування програм навчання, а також для студентів та абітурієнтів, які мають можливість вибрати найбільш відповідні дисципліни для власного навчання.

2.6 Висновки до розділу 2

Реалізовано спосіб формування пулу вибіркового навчальних дисциплін на основі генетичного алгоритму.

Представлено послідовність формування пулу вибіркового навчальних дисциплін, яка включає в себе такі етапи, як ініціалізація популяції, обчислення функції пристосованості, відбір, кроссовер і мутація.

У реалізованому формуванні пулу вибіркового навчальних дисциплін на основі генетичного алгоритму за популяцію приймалося сукупність можливих пулів вибіркового дисциплін, за індивід – окремий пул. Геном формується з чотирьох блоків – факультет, кафедра, викладач, дисципліна. Обмеженнями виступали кількість кредитів ECTS та кількість навчальних дисциплін, що вивчаються. За цільову функцію способу формування вибіркового дисциплін приймалися вимоги законодавства та рівень зацікавленості здобувача освіти в поглибленні певних компетентностей.

Сформовано та спроектовано структуру інформаційної системи для реалізації способу формування пулу вибіркового навчальних дисциплін на основі генетичного алгоритму.

Розділ 3 Програмна реалізація способу формування пулу вибіркового навчальних дисциплін на основі генетичного алгоритму

3.1 Структура модулів системи, їх взаємозв'язок

Модуль авторизації є одним з найважливіших модулів в системі. Його основне призначення полягає в тому, щоб забезпечити захист від несанкціонованого доступу до системи, а також забезпечити контроль доступу до різних ресурсів системи на основі рівнів доступу. Модуль авторизації вимагає від користувача введення логіну та пароля для того, щоб ідентифікувати користувача та надати йому відповідний рівень доступу.

Для системи з ролями адміністратора та студента, модуль авторизації виконує наступні функції:

- для адміністратора модуль авторизації перевіряє логін та пароль, адміністратор має доступ до всіх функцій системи, таких як CRUD операції з факультетами, кафедрами, викладачами та дисциплінами;
- для студента модуль авторизації перевіряє логін та пароль, студент має доступ до опитування на основі якого й буде сформовано пул вибіркового дисциплін.

Переглянути детальніше роботу модуля авторизації можна на рисунку 3.1.

Модуль CRUD операцій використовується для роботи з базою даних та забезпечення доступу до даних для інших модулів системи. Його призначення полягає в забезпеченні можливості створення, читання, оновлення та видалення даних у базі даних.

Модуль CRUD містить чотири базові функції, які надають інтерфейс для взаємодії з даними:

1. Create – ця функція дозволяє створити новий запис у базі даних.
2. Read – ця функція дозволяє читати дані з бази даних та відображати їх для користувача.
3. Update – ця функція дозволяє змінювати існуючі дані у базі даних.

4. Delete – ця функція дозволяє видаляти існуючі дані з бази даних.

Для кожної з цих функцій розробляється окремий інтерфейс, що дозволяє виконувати відповідні дії з даними.

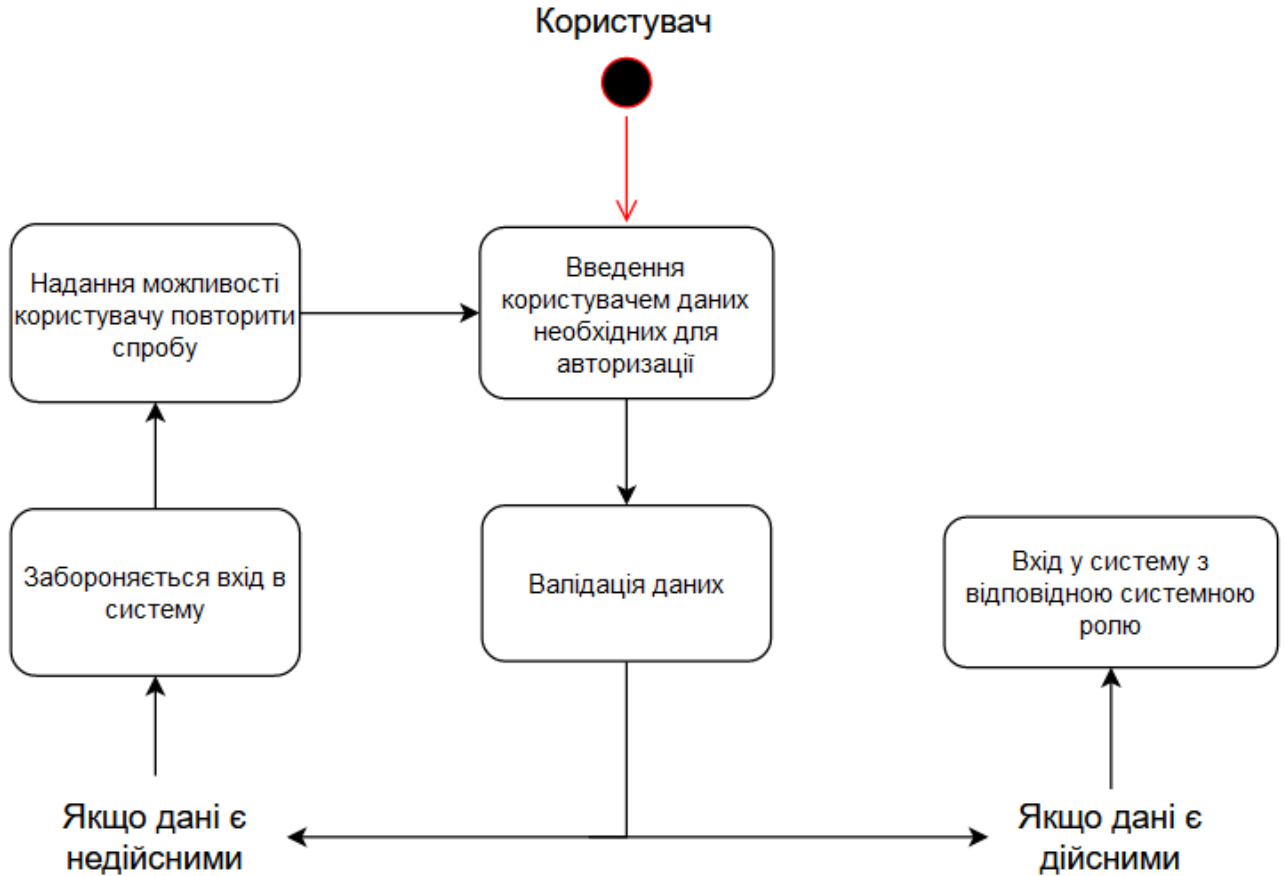


Рисунок 3.1 – Діаграма роботи модуля авторизації

У системи формування пулу вибірових дисциплін, модуль CRUD операцій використовується для створення, збереження та оновлення даних про факультети, кафедри, викладачів та дисципліни у базі даних системи. Це дозволяє забезпечити доступність даних для інших модулів, наприклад, модуля генерації пулу вибірових дисциплін, який опрацьовує дані опитування користувачів та рекомендує навчальні дисципліни на основі цих даних (рисунок 3.2).



Рисунок 3.2 – Діаграма роботи модуля CRUD операцій

Модуль генерації пулу вибірових дисциплін є ключовим модулем системи. Цей модуль відповідає за генерацію пулу вибірових дисциплін для студента на основі його вподобань, які він вказав у опитуванні. Модуль отримує дані опитування з модулю авторизації та взаємодіє з модулями CRUD операцій для отримання інформації про доступні дисципліни.

Для генерації пулу вибірових дисциплін модуль має такий функціонал (рисунок 3.3):

1. Отримання даних опитування студентів. Модуль отримує дані опитування, що містять інформацію про вподобання студентів.
2. Аналіз даних опитування. На основі даних опитування модуль аналізує вподобання студентів та формує список дисциплін, які можуть відповідати заданим вимогам.
3. Оцінка пристосованості дисциплін. Модуль використовує функцію пристосованості для оцінки того, наскільки добре дисципліна відповідає вимогам студентів.

4. Формування пулу вибірових дисциплін. На основі результатів оцінки пристосованості дисциплін модуль формує пул вибірових дисциплін.
5. Збереження результатів. Модуль зберігає результати формування пулу вибірових дисциплін.



Рисунок 3.3 – Діаграма роботи модуля генерації пулу вибірових дисциплін

Взаємозв'язок між модулями є важливою частиною проектування системи та забезпечує роботу системи в цілому. Модуль авторизації забезпечує

доступ до інших модулів для користувачів з певними ролями, модулі CRUD операцій забезпечують зберігання та управління інформацією для модулю генерації пулу вибірових дисциплін, а модуль генерації пулу вибірових дисциплін в свою чергу отримує дані опитування користувачів та використовує інформацію про дисципліни з модулів CRUD операцій (рисунок 3.4).

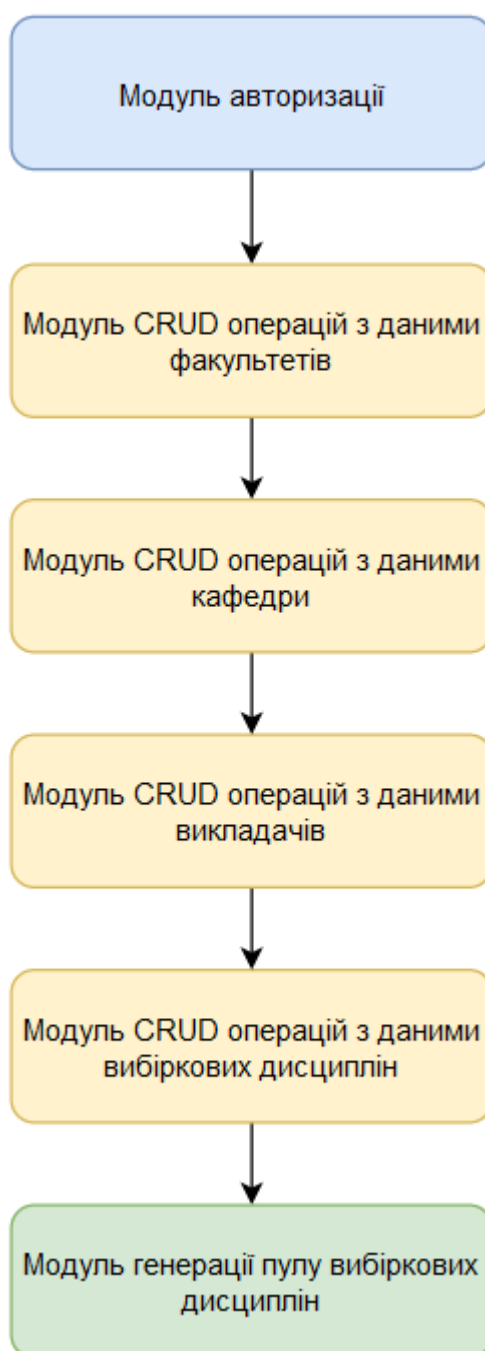


Рисунок 3.4 – Послідовність виконання модулів інформаційної системи

Підсумовуючи можна зазначити, що розглянута структура модулів системи формування пулу вибіркових навчальних дисциплін складається з трьох модулів: модуль авторизації, модулі CRUD операцій з базою даних та модуль генерації пулу вибіркових дисциплін.

3.2 Опис функціональних можливостей інформаційної системи

Для початку розберемо функціональні можливості адміністратора системи. Щоб розпочати працювати з системою у ролі адміністратора, необхідно перейти на сторінку авторизації та ввести логін і пароль, що відповідають обліковому запису адміністратора системи (рисунок 3.5).

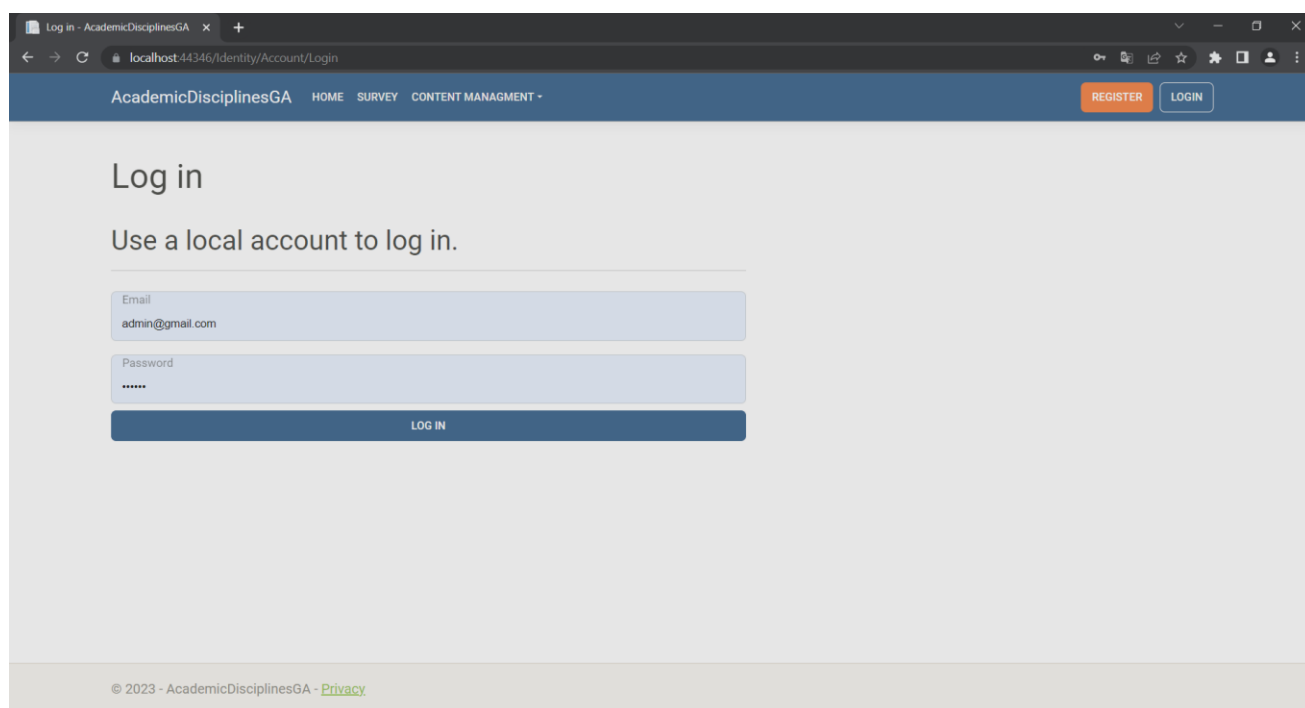


Рисунок 3.5 – Сторінка входу в систему

Коли користувач успішно авторизується в системі, у ролі адміністратора, йому стане доступним модуль управління даними, що включає в себе можливість перегляду, додавання, редагування та видалення записів про факультети, кафедри, викладачів та навчальні дисципліни (Рисунок 3.6).

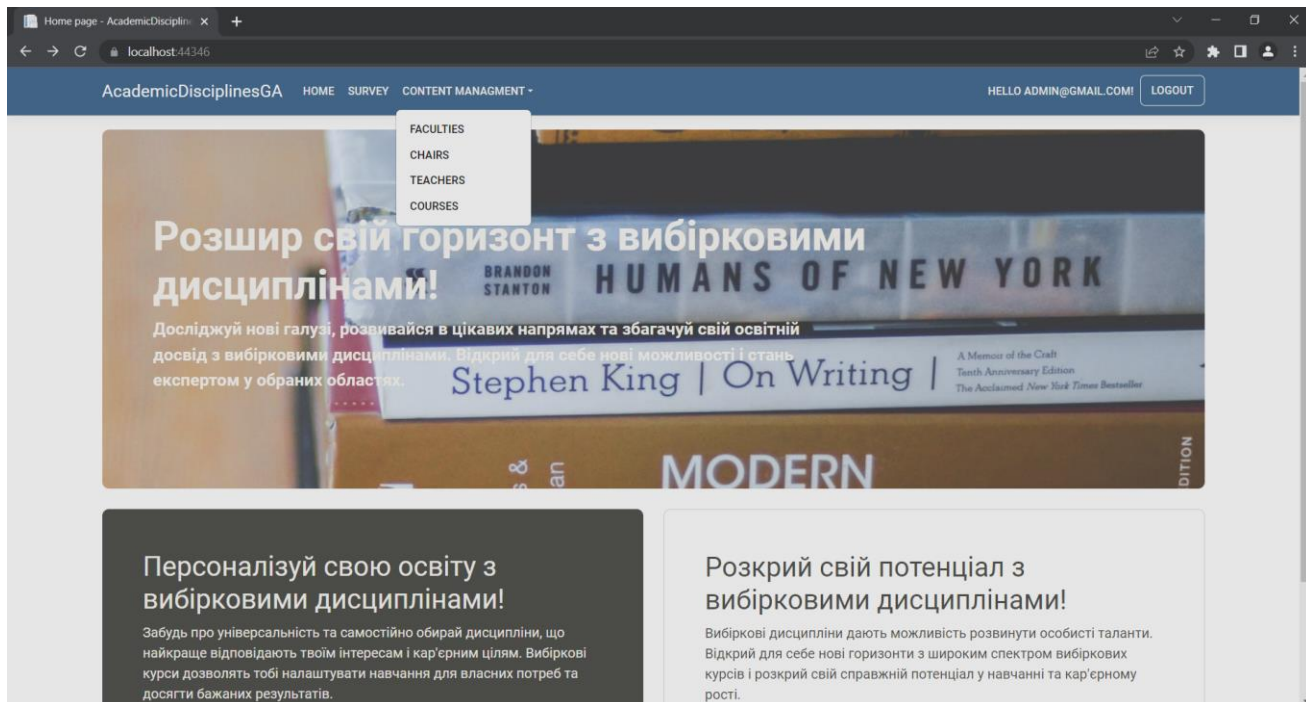


Рисунок 3.6 – Головна сторінка

Першою по важливості є сутність факультету, на основі якої створюються усі інші записи. Сторінку для керування даними факультетів можна побачити на рисунку 3.7.

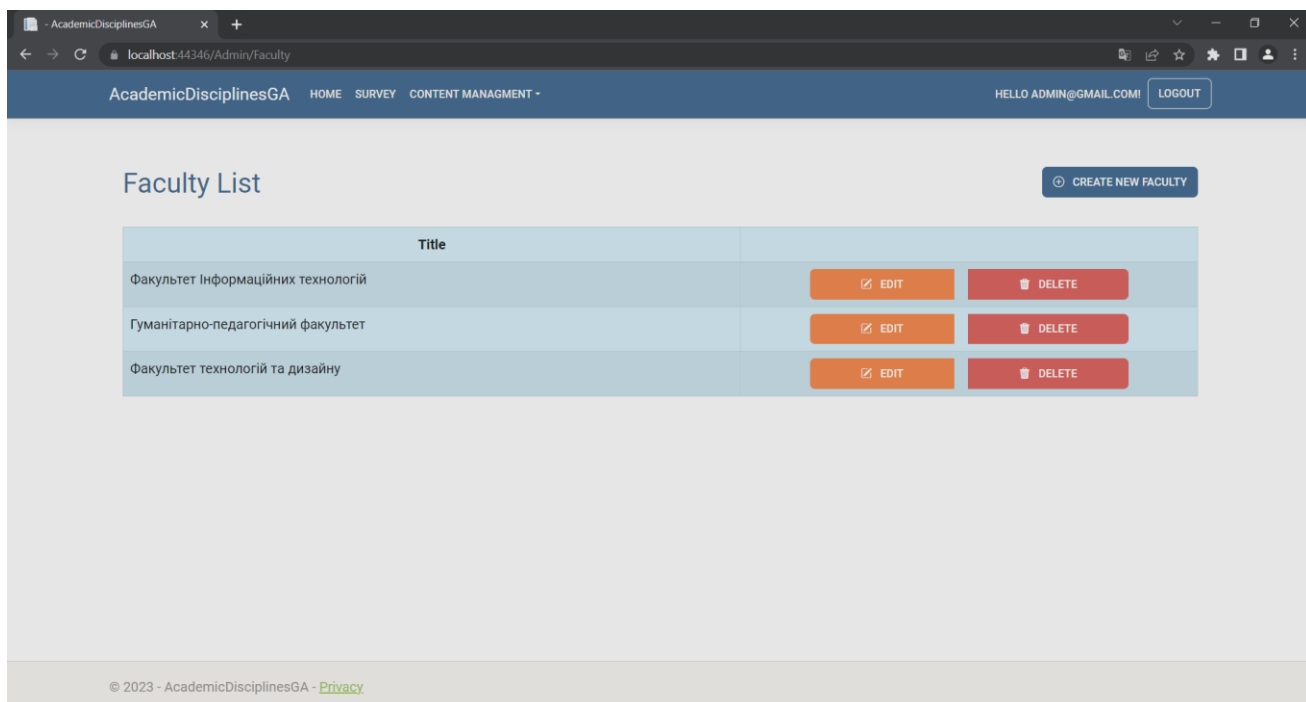


Рисунок 3.7 – Адміністративна сторінка керування даними факультетів

При створенні нового запису факультету чи оновленні існуючого, адміністратора буде переадресовано на відповідну сторінку, якщо поля сторінки порожні, створюється новий запис (рисунок 3.8).

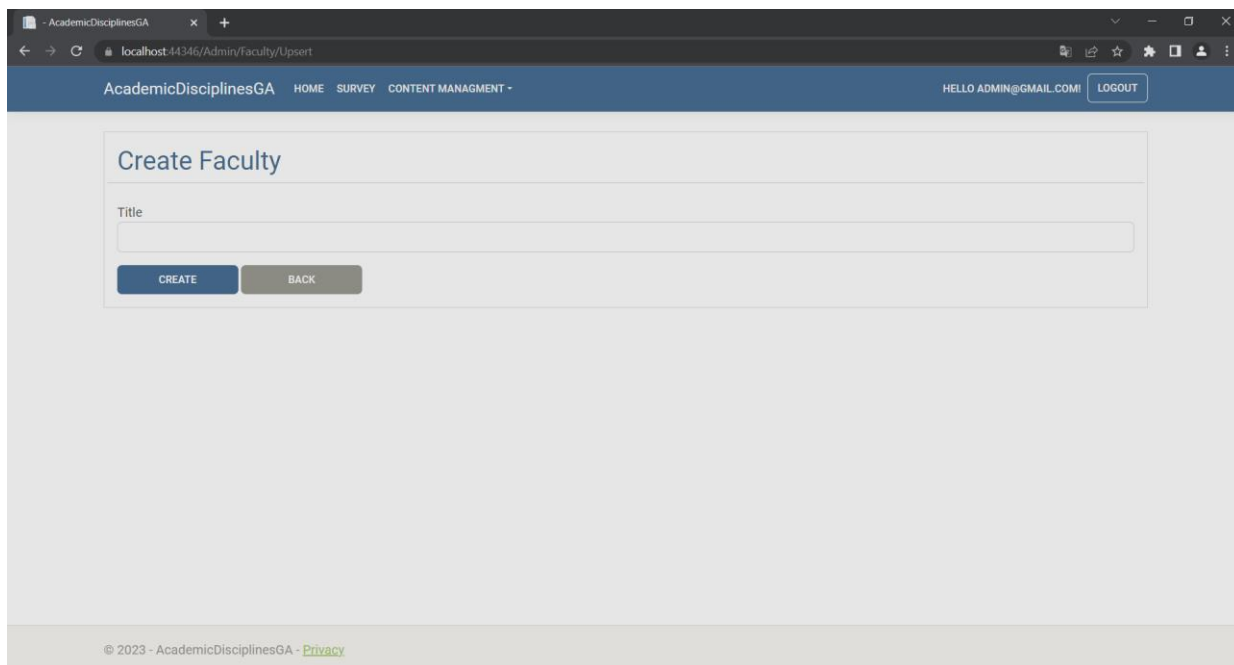


Рисунок 3.8 – Сторінка створення записів факультетів

У випадку коли ж поля відповідають значенням існуючого факультету, тоді виконується операція оновлення запису (рисунок 3.9).

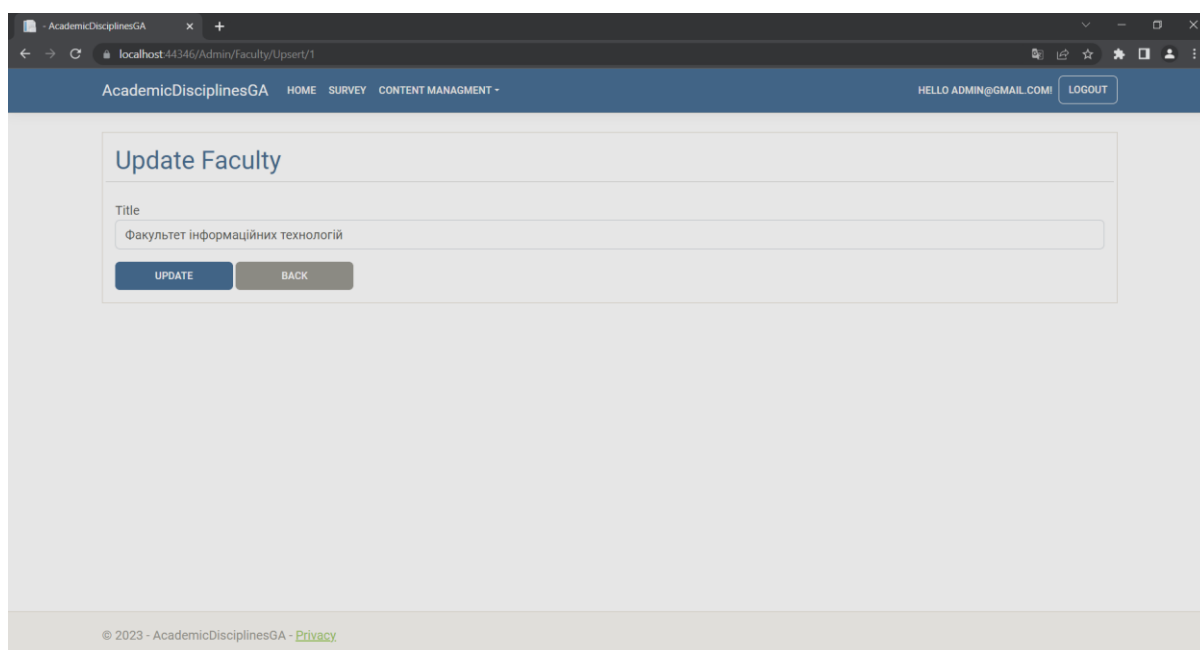


Рисунок 3.9 – Сторінка оновлення записів факультетів

Наступною сутністю після факультету слідує кафедра. Сторінка для керування даними кафедр представлена на рисунку 3.10. Властивостями даної сутності є: назва кафедри та факультет, до якого належить кафедра.

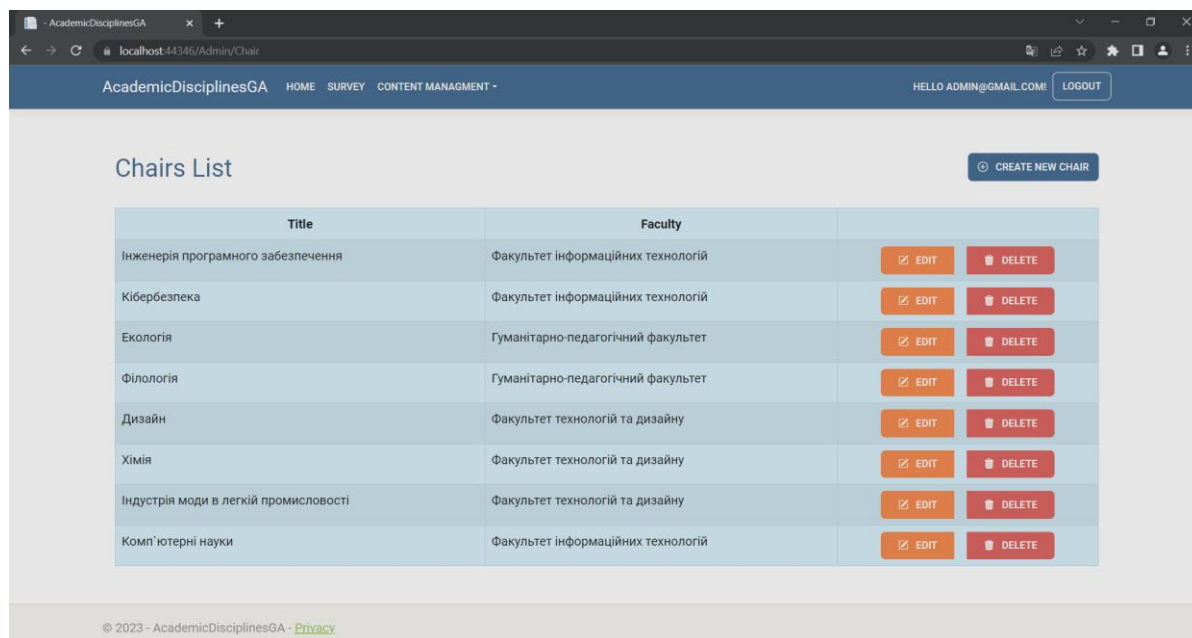


Рисунок 3.10 – Адміністративна сторінка керування даними кафедр

Сторінка для створення та оновлення записів кафедр зображена на рисунку 3.11.

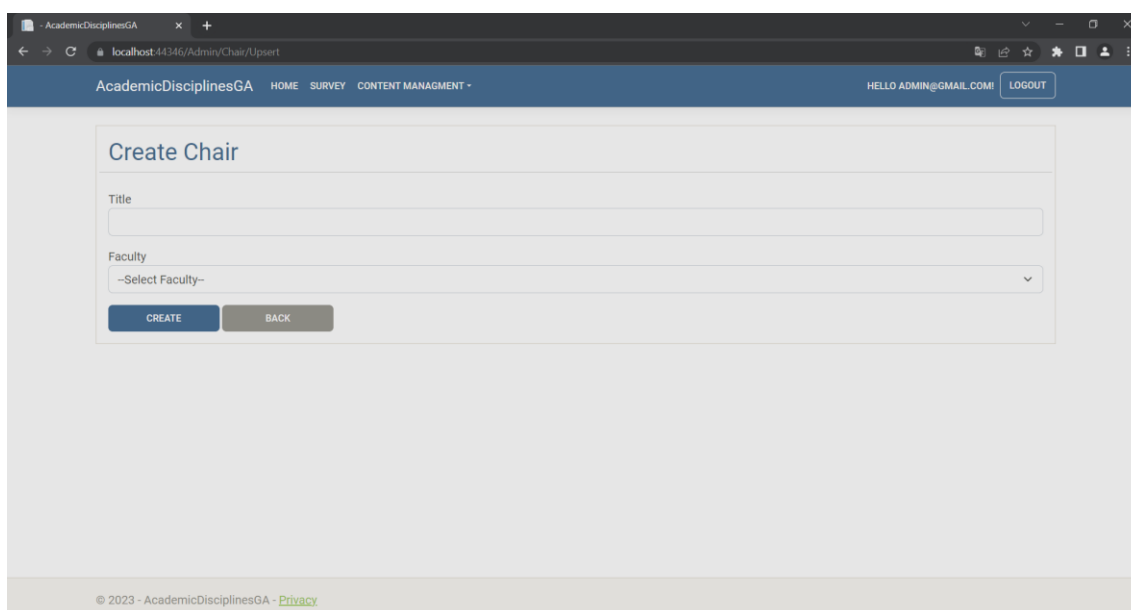


Рисунок 3.11 – Сторінка створення та оновлення записів кафедр

Далі розглянемо сторінку для огляду даних викладачів (рисунок 3.12).

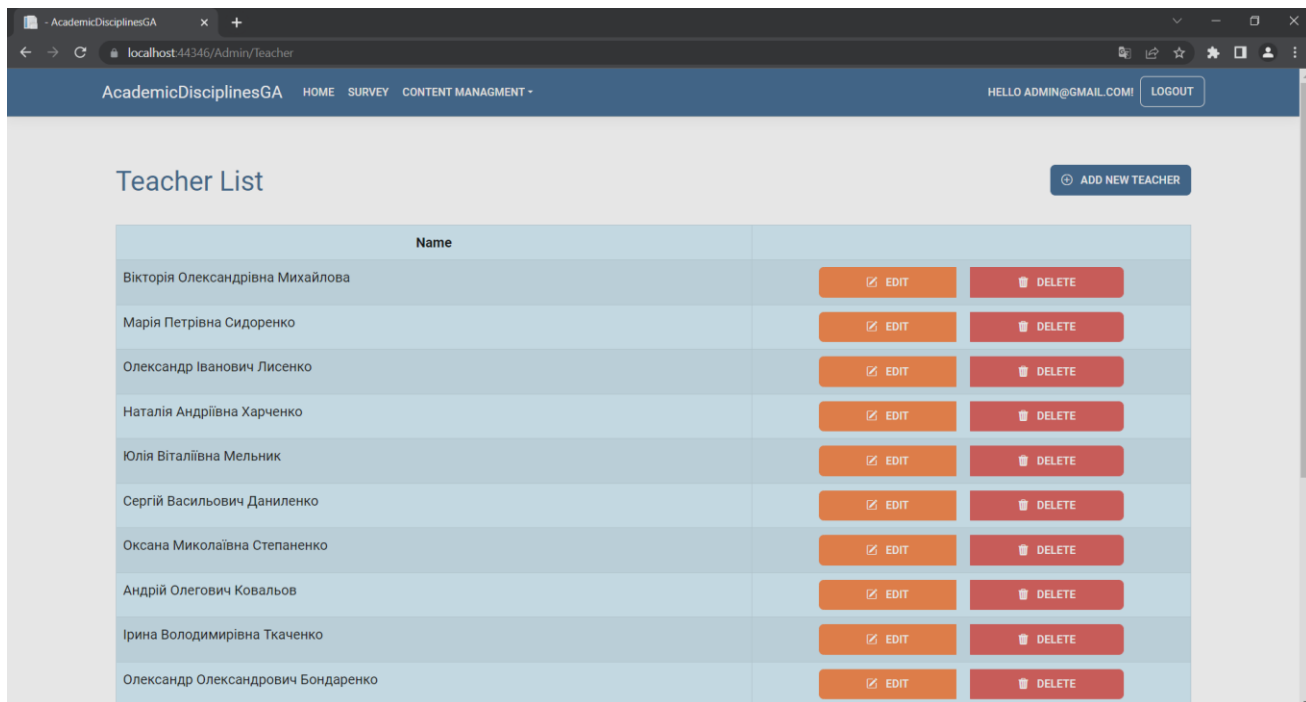


Рисунок 3.12 – Адміністративна сторінка керування даними викладачів

На рисунку 3.13 зображено сторінку, що призначена для створення та оновлення записів викладачів.

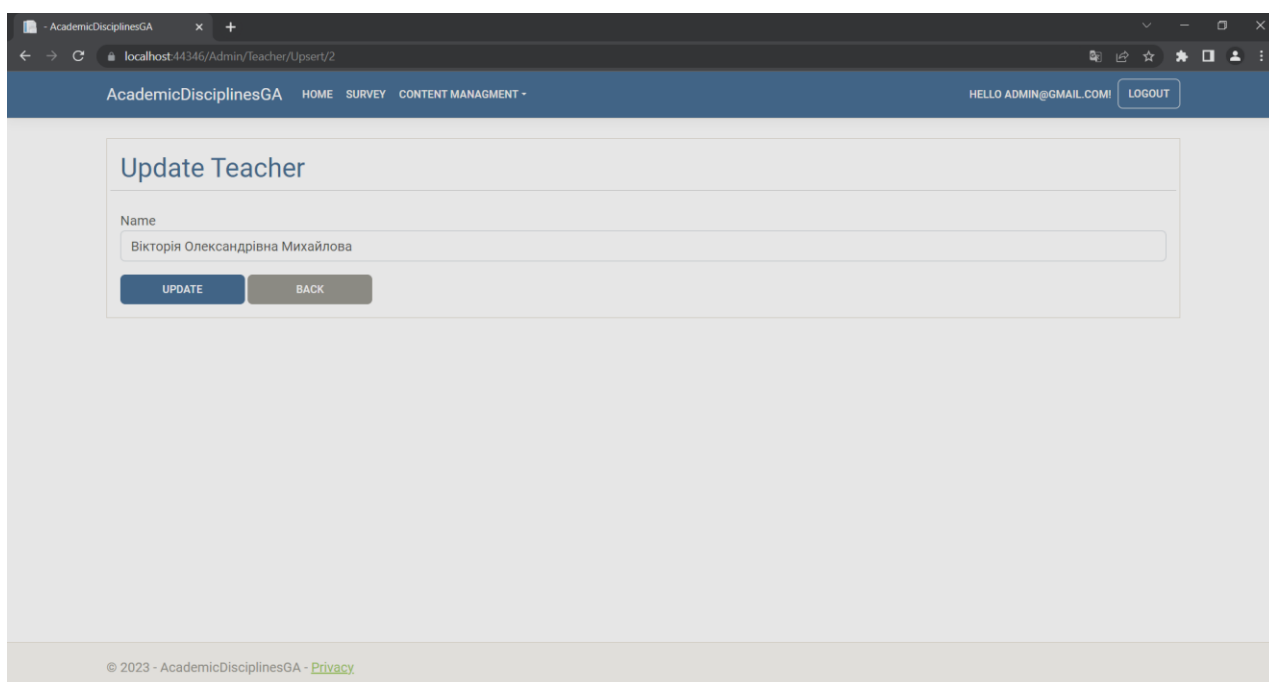


Рисунок 3.13 – Сторінка створення та оновлення записів викладачів

Останньою та основною сутністю з якою буде працювати розроблений алгоритм, є навчальна дисципліна. Сторінка для роботи з даними навчальних дисциплін зображена на рисунку 3.14. Ця сутність представлена у вигляді таких властивостей: назва навчальної дисципліни, кількість кредитів ECTS, викладач, що викладає відповідну дисципліну, та кафедра, яка є відповідальною за викладання відповідної дисципліни.

The screenshot shows a web browser window with the URL localhost:44346/Admin/Course. The page title is 'AcademicDisciplinesGA' and it includes navigation links for HOME, SURVEY, and CONTENT MANAGEMENT. A user is logged in as HELLO ADMIN@GMAIL.COM. The main content area is titled 'Courses List' and features a 'CREATE NEW COURSE' button. Below this is a table with the following data:

Title	ECTS	Chair	Teacher	EDIT	DELETE
Методи та системи штучного інтелекту	6	Комп'ютерні науки	Вікторія Олександрівна Михайлова		
WEB-технології та WEB-дизайн	6	Комп'ютерні науки	Вікторія Олександрівна Михайлова		
Об'єктно-орієнтоване програмування	6	Комп'ютерні науки	Олександр Іванович Лисенко		
Проектування баз даних	6	Комп'ютерні науки	Олександр Іванович Лисенко		
Основи інформаційної безпеки	6	Кібербезпека	Марія Петрівна Сидоренко		
Теорія інформації та кодування	6	Кібербезпека	Марія Петрівна Сидоренко		
Безпека вебресурсів	6	Кібербезпека	Наталія Андріївна Харченко		

Рисунок 3.14 – Адміністративна сторінка керування даними вибірових дисциплін

На рисунку 3.15 зображено сторінку, яка призначена для створення та оновлення записів навчальних дисциплін.

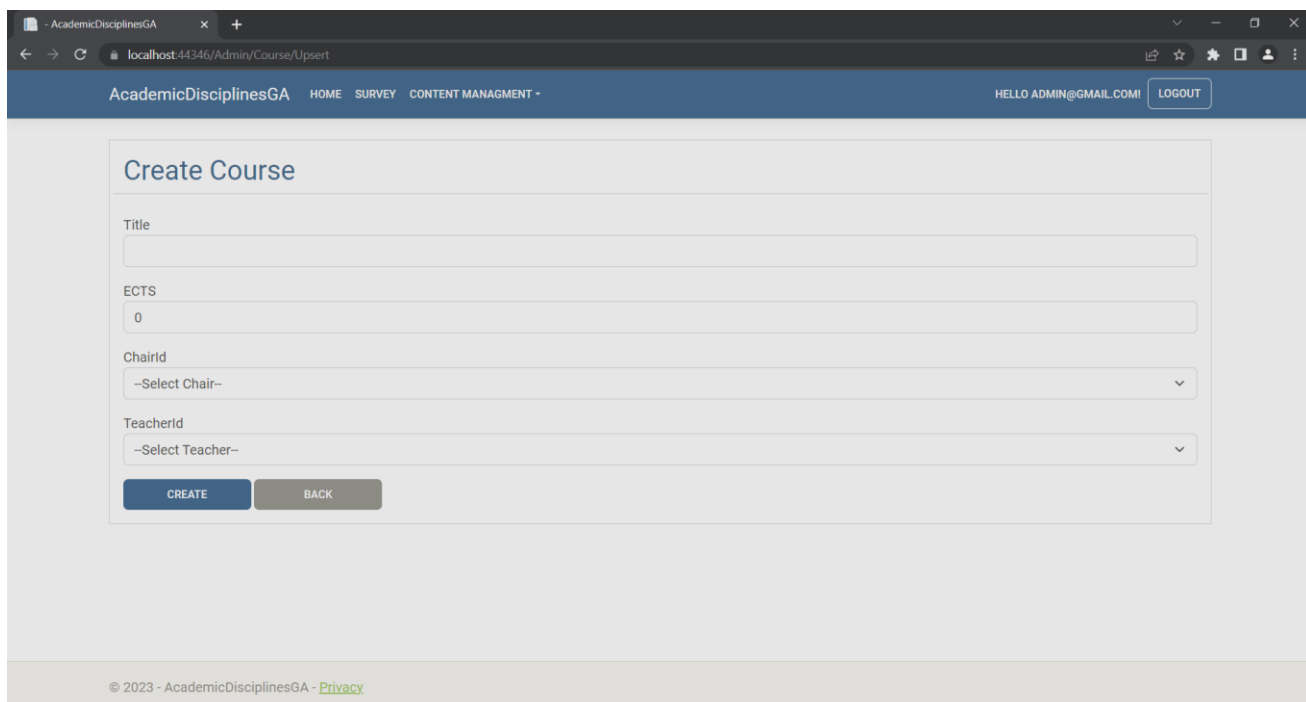


Рисунок 3.15 – Сторінка створення та оновлення записів вибіркових дисциплін

Тепер розглянемо основні функціональні можливості студента. Для початку здобувачу освіти необхідно авторизуватись у системі, якщо користувач вперше користується застосунком, йому потрібно перейти на відповідну сторінку та зареєструватись у системі (рисунок 3.16).

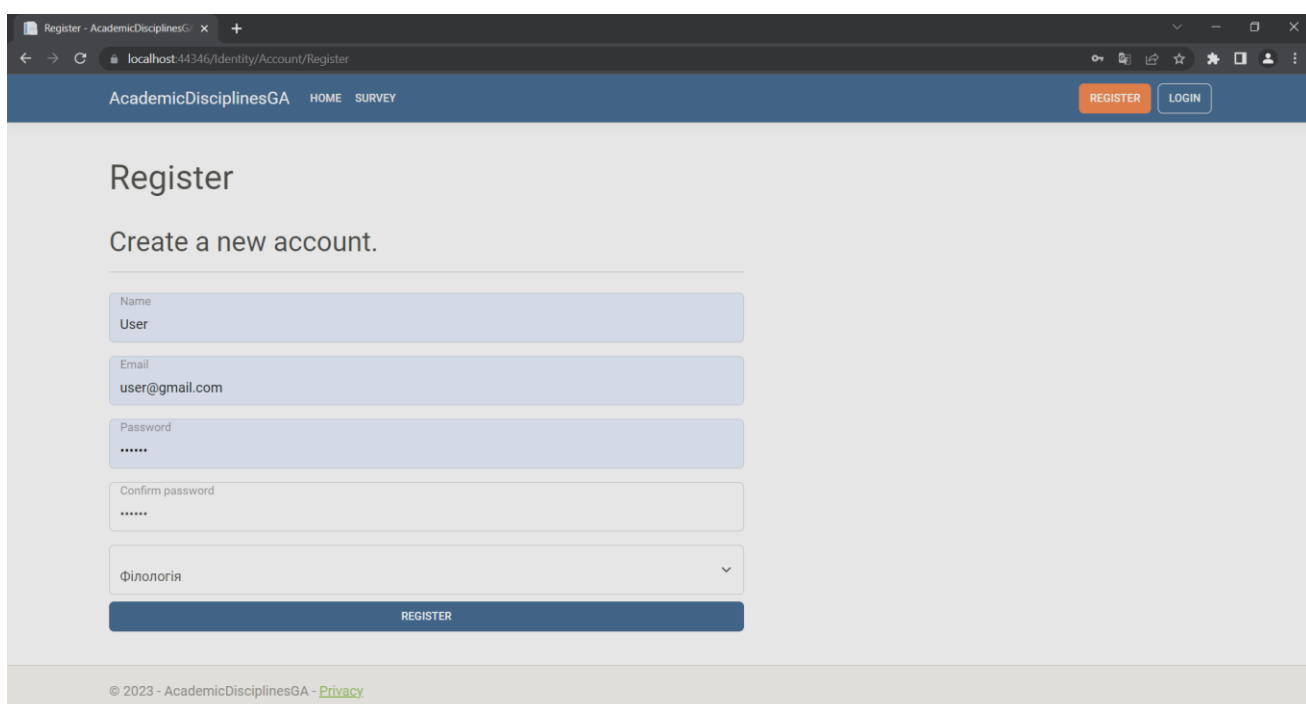


Рисунок 3.16 – Сторінка реєстрації облікового запису

Після авторизації у системі користувача буде автоматично перенаправлено на головну сторінку, та надано доступ до проходження опитування, на основі якого формується індивідуальний пул вибіркового дисциплін. Опитування акцентує увагу саме на визначенні основних вподобань користувача, таких як: опанування нових чи поглиблення вже існуючих компетенцій, навчальний напрям, улюблений викладач (рисунок 3.17).

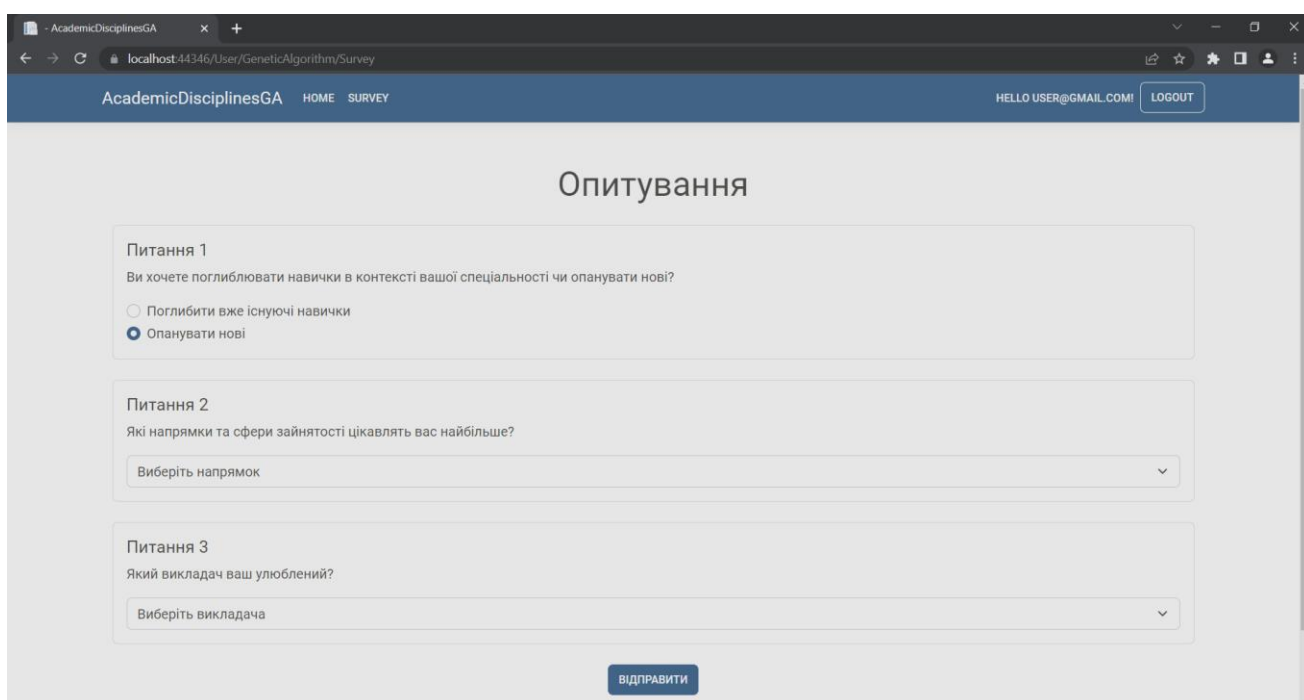


Рисунок 3.17 – Сторінка користувацького опитування

Тепер розглянемо декілька результатів роботи модулю генерації вибіркового дисциплін, на основі повторюваних та різних даних опитування.

Розглянемо користувача, що навчається на кафедрі філології, Гуманітарно-педагогічного факультету. Студент прагне здобути якомога більше компетентностей у різноманітних галузях, найбільше його цікавить напрям комп'ютерних наук. Крім того, здобувачу освіти до вподоби навчальні дисципліни викладача Ірини Михайлівни Черниш. Відповідно до зазначених користувачем вимог, сформовано відповіді до опитування (рисунок 3.18).

AcademicDisciplinesGA HOME SURVEY HELLO USER@GMAIL.COM! LOGOUT

Опитування

Питання 1
Ви хочете поглиблювати навички в контексті вашої спеціальності чи опанувати нові?

Поглибити вже існуючі навички
 Опанувати нові

Питання 2
Які напрямки та сфери зайнятості цікавлять вас найбільше?

Комп'ютерні науки

Питання 3
Який викладач ваш улюблений?

Ірина Михайлівна Черниш

ВІДПРАВИТИ

Рисунок 3.18 – Сторінка пройденого опитування

Після проходження вище зазначеного опитування користувача буде переадресовано на сторінку з результатами, тобто згенерованим пулом вибіркових дисциплін (рисунок 3.19).

AcademicDisciplinesGA HOME SURVEY HELLO USER@GMAIL.COM! LOGOUT

Пул вибіркових дисциплін

Title	ECTS	Chair	Teacher
Грунтознавство	6	Екологія	Оксана Миколаївна Степаненко
Теорія інформації та кодування	6	Кібербезпека	Марія Петрівна Сидоренко
Основи проектування виробів	6	Індустрія моди в легкій промисловості	Ірина Михайлівна Черниш
WEB-технології та WEB-дизайн	6	Комп'ютерні науки	Вікторія Олександрівна Михайлова
Архітектура комп'ютера	6	Інженерія програмного забезпечення	Сергій Васильович Даниленко
Гідрологія	6	Екологія	Оксана Миколаївна Степаненко
Основи технологій виробів	6	Індустрія моди в легкій промисловості	Ірина Михайлівна Черниш
Об'єктно-орієнтоване програмування	6	Комп'ютерні науки	Олександр Іванович Лисенко
Кольорознавство	6	Дизайн	Євгенія Володимирівна Мельник
Проектування баз даних	6	Комп'ютерні науки	Олександр Іванович Лисенко

© 2023 - AcademicDisciplinesGA - [Privacy](#)

Рисунок 3.19 – Сторінка зі сформованим пулом вибіркових дисциплін

У результаті сформований пул містить дисципліни, що відповідають запиту користувача, а саме відносяться до кафедри «Комп'ютерні науки» та викладаються Іриною Михайлівною Черниш. Якщо ж використати тіж самі дані для опитування, результат буде дещо відрізнятись від попереднього, оскільки здобувача освіти цікавить опанування компетентностей з різних напрямів (рисунок 3.20).

Title	ECTS	Chair	Teacher
Основи технології виробів	6	Індустрія моди в легкій промисловості	Ірина Михайлівна Черниш
Проектування баз даних	6	Комп'ютерні науки	Олександр Іванович Лисенко
WEB-технології та WEB-дизайн	6	Комп'ютерні науки	Вікторія Олександрівна Михайлова
Моделювання і прогнозування стану довкілля	6	Екологія	Андрій Олегович Ковальов
Комп'ютерна дизайн-графіка	6	Дизайн	Сергій Віталійович Ковальов
Гідрологія	6	Екологія	Оксана Миколаївна Степаненко
Методи та системи штучного інтелекту	6	Комп'ютерні науки	Вікторія Олександрівна Михайлова
Основи проектування виробів	6	Індустрія моди в легкій промисловості	Ірина Михайлівна Черниш
Об'єктно-орієнтоване програмування	6	Комп'ютерні науки	Олександр Іванович Лисенко
Фізичні методи дослідження речовини	6	Хімія	Михайло Андрійович Жуков

Рисунок 3.20 – Сторінка зі сформованим пулом вибірових дисциплін з використанням повторюваних даних

Розглянемо результати роботи алгоритму з іншими даними опитування. Тепер здобувач освіти прагне поглибити навички у контексті його спеціальності, крім того він цікавиться кібербезпекою та прагне записатись на курси до викладача Наталії Миколаївни Шевчук. Відповідно до вказаних побажань користувача формуються відповіді на опитування (рисунок 3.21).

Рисунок 3.21 – Сторінка пройденого опитування

У сформованому пулі вибірових дисциплін представлено навчальні дисципліни з Гуманітарно-педагогічного факультету, кафедри Кібербезпеки та ті, що викладаються Наталією Миколаївною Шевчук (рисунок 3.22).

Title	ECTS	Chair	Teacher
Теорія інформації та кодування	6	Кібербезпека	Марія Петрівна Сидоренко
Основи технології виробів	6	Індустрія моди в легкій промисловості	Ірина Михайлівна Черниш
Основи інформаційної безпеки	6	Кібербезпека	Марія Петрівна Сидоренко
Усна народна творчість	6	Філологія	Олександр Олександрович Бондаренко
Екологічна безпека	6	Екологія	Андрій Олегович Ковальов
Безпека вебресурсів	6	Кібербезпека	Наталія Андріївна Харченко
Моделювання і прогнозування стану довкілля	6	Екологія	Андрій Олегович Ковальов
Історія української літератури	6	Філологія	Ірина Володимирівна Ткаченко
Загальна хімічна технологія	6	Хімія	Наталія Миколаївна Шевчук
Екологічна безпека хімічних виробництв	6	Хімія	Наталія Миколаївна Шевчук

Рисунок 3.22 – Сторінка зі сформованим пулом вибірових дисциплін

Отже, інформаційна система формування пулу вибірових дисциплін допомагає покращити процес вибору навчальних дисциплін студентами та забезпечити ефективну роботу адміністратора системи.

3.3 Висновки до розділу 3

Реалізовано програмний застосунок для реалізації способу формування пулу вибірових навчальних дисциплін на основі генетичного алгоритму.

Розроблена структура модулів інформаційної системи формування пулу вибірових навчальних дисциплін, включає адміністративну та користувацьку систему, кожна із яких забезпечує виконання усіх необхідних функцій.

Виконана перевірка довела ефективність способу формування пулу вибірових навчальних дисциплін на основі генетичного алгоритму та підтвердила можливість отримання результату з меншими ресурсними витратами.

Модуль генетичного алгоритму ефективно і точно генерує оптимальні пули вибірових дисциплін, забезпечуючи максимальне врахування вимог та обмежень, таких як кількість кредитів ЄКТС та кількість дисциплін у пулі.

Програмна реалізація також підтримує можливість додавання, оновлення та видалення даних про факультети, кафедри, викладачів та дисципліни. Модуль CRUD дозволяє адміністраторам легко керувати цими даними.

Програмна реалізація забезпечує зручний та інтуїтивно зрозумілий інтерфейс для користувачів, що дозволяє легко взаємодіяти з системою і здійснювати вибір дисциплін на основі особистих уподобань.

Висновки

В результаті виконання кваліфікаційної роботи бакалавра на тему «Спосіб формування пулу вибіркового навчальних дисциплін на основі генетичного алгоритму» було розроблено ефективний метод оптимізації процесу вибору навчальних дисциплін студентами.

Для досягнення поставленої мети були вирішені такі задачі:

- проведено аналіз методів формування пулу вибіркового дисциплін, його мети та завдань;
- проведено аналіз можливостей, переваг та недоліків генетичного алгоритму;
- реалізовано спосіб формування пулу вибіркового навчальних дисциплін на основі генетичного алгоритму;
- реалізовано інформаційну технологію визначення пулу вибіркового навчальних дисциплін на основі генетичного алгоритму;
- проведено експериментальне тестування інформаційної технології.

В роботі була розглянута загальна схема оптимізаційної задачі формування пулу вибіркового дисциплін, а також розроблено математичну модель, яка враховує змінні, цільову функцію та обмеження задачі. Для розв'язання цієї задачі було використано генетичний алгоритм, який дозволяє знаходити оптимальні рішення шляхом еволюції та взаємодії генетичних операторів.

У процесі реалізації програмного модуля були враховані функціональні можливості, необхідні для ефективного взаємодії користувачів з системою. Модуль авторизації забезпечує безпечний доступ до системи з різними ролями, а модуль CRUD операцій дозволяє адміністраторам здійснювати додавання, оновлення та видалення даних про факультети, кафедри, викладачів та дисципліни.

Особливу увагу слід звернути на модуль генерації пулу вибіркового дисциплін, що опрацьовує дані опитування користувача та застосовує

генетичний алгоритм для вибору оптимального пулу. Цей модуль дозволяє забезпечити максимальну персоналізацію вибору дисциплін, враховуючи особисті вподобання студентів.

Отримані результати показали ефективність запропонованого способу формування пулу вибіркового навчальних дисциплін на основі генетичного алгоритму. Цей метод може бути використаний для оперативного формування студентоцентризованих навчальних планів та забезпечення більшої задоволеності від навчання.

Отже, в результаті було розроблено ефективний спосіб формування пулу вибіркового навчальних дисциплін на основі генетичного алгоритму, що може бути використаний в інформаційній системі формування навчальних планів.

Перелік посилань

1. Навчальний заклад. URL: https://leksika.com.ua/15941024/legal/navchalniy_zaklad
2. Навчальна дисципліна. URL: https://web.archive.org/web/20150316234542/http://eduknigi.com/ped_view.php?id=126
3. Освітня програма. URL: <https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/navchalni-programi>
4. Студентоцентроване навчання. URL: <https://nmetau.edu.ua/ua/mqual/i3003/p3332>
5. Kim, M. (2017). "Individual learning pathways in higher education: A systematic review of empirical studies." *Higher Education Research & Development*, 36(2), 217-233.
6. Індивідуальна освітня траєкторія. URL: <https://nus.org.ua/bez-rubryky/indyvidualna-osvitnya-trayektoriya-ta-indyvidualnyj-navchalnyj-plan-shho-tse-i-dlya-kogo/>
7. Cairns, L., & Krägeloh, C. (2010). "Flexible learning: Designing for success." *Journal of Open, Flexible, and Distance Learning*, 14(1), 41-54.
8. Нормативна та варіативна складова освітньої програми. URL: <https://www.hneu.edu.ua/wp-content/uploads/2018/11/Metodychni-pidhody-do-formuvannya-variativnoyi-skladovoyi-osvitnih-prohram-1.pdf>
9. Положення про вибіркові дисципліни. URL: <https://mon.gov.ua/storage/app/media/gromadske-obgovorennya/2022/12/09/Zvit-NO-projekt.nakazu.Pro.catv.Typ.pol.pro.orh.osv.prots.v.zakl.FPO.09.12.2022.pdf>
10. Майнор. URL: <https://normative.sumdu.edu.ua/doc/1/00e9ba12-a107-ea11-9ad5-001a4be6d04a>
11. Мейджор. URL: <https://normative.sumdu.edu.ua/doc/1/00e9ba12-a107-ea11-9ad5-001a4be6d04a>

12. Dhillon, G., & Khosla, R. (2018). "An investigation of factors influencing students' choice of elective courses in higher education." *Journal of Education and Training Studies*, 6(11), 80-91.

13. Оптимізація. URL: https://shron1.chtyvo.org.ua/Zhaldak_Myrosлав/Osnovy_teorii_i_metodiv_optymizat_sii.pdf?PHPSESSID=срpktqvm1e49rqamn1sp7sll1m6

14. "Genetic Algorithms in Search, Optimization, and Machine Learning" by David E. Goldberg.

15. Еволюційні обчислення. URL: http://eir.zntu.edu.ua/bitstream/123456789/2642/1/Oliinyk_Evolutionary_Computing_and.pdf

16. "Evolutionary Optimization Algorithms" by Dan Simon.

17. NP-складна задача. URL: <https://en.wikipedia.org/wiki/NP-hardness>

18. Градієнтні методи. URL: https://www.researchgate.net/profile/Vasy1-Teslyuk/publication/328967632_GRADIENTNI_METODI_ROZV%27AZANNA_ZADAC_BEZUMOVNOI_OPTIMIZACII/links/5bedaf4592851c6b27c24419/GRADIENTNI-METODI-ROZVAZANNA-ZADAC-BEZUMOVNOI-OPTIMIZACII.pdf

19. Застосунок для формування освітньої траєкторії здобувачів вищої освіти. URL: <http://elar.khmnu.edu.ua/jspui/bitstream/123456789/12814/10/%d0%9a%d0%a0%d0%9c%20%d0%9c%d0%b0%d0%bb%d0%b0%d0%b9%d0%ba%d0%be%20%281%29.pdf>

20. "Introduction to Evolutionary Computing" by A.E. Eiben and J.E. Smith.

21. "Handbook of Genetic Algorithms" by Lawrence Davis.

22. "Genetic Algorithms + Data Structures = Evolution Programs" by Zbigniew Michalewicz.

23. Kuh, G. D. (2008). "High-impact educational practices: What they are, who has access to them, and why they matter".

24. Individualized Education Program (IEP) Process. URL: <https://www.understood.org/articles/at-a-glance-anatomy-of-an-iep>

25. "The Art and Science of Personalized Learning" by Shawn C. Smith and Kirsten B. Jones.
26. Mayhew, K. P., & Rockenbach, A. N. (2018). "A closer look at the elective course experience: Understanding student selection, motivation, and satisfaction".
27. Huisman, J., & Adelman, C. (Eds.). (2018). "International perspectives on the governance of higher education: Alternative frameworks for coordination".
28. "Personalized Learning: A Guide for Engaging Students with Technology" by Peggy Grant and Dale Basye.
29. Horn, M. B., & Staker, H. (2015). "Blended: Using disruptive innovation to improve schools." *Phi Delta Kappan*, 96(4), 8-13.
30. Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II". *Lecture Notes in Computer Science*, 1917, 849-858.
31. Mitchell, M. (1996). "An introduction to genetic algorithms". MIT Press.

ДОДАТКИ

Додаток А

Програмні коди

Лістинг ChairController.cs:

```

namespace
AcademicDisciplinesGA.Areas.Admin.Controllers
{
    [Area("Admin")]
    [Authorize(Roles = "Admin")]
    public class ChairController : Controller
    {
        private readonly ApplicationDbContext
        _context;

        public
        ChairController(ApplicationDbContext context)
        {
            _context = context;
        }

        public IActionResult Index()
        {
            IEnumerable<Chair> chairList =
            _context.Chairs.Include(x => x.Faculty).ToList();
            return View(chairList);
        }

        public IActionResult Upsert(int? id)
        {
            ChairVM chairVM = new()
            {
                Chair = new(),
                FacultyList =
                _context.Faculties.Select(x => new SelectListItem
                { Text = x.Title, Value = x.Id.ToString() })
            };

            if (id == 0 || id == null)
            {
                return View(chairVM);
            }
            else
            {
                chairVM.Chair =
                _context.Chairs.FirstOrDefault(x => x.Id == id);
                return View(chairVM);
            }
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public IActionResult Upsert(ChairVM obj)
        {
            if (ModelState.IsValid)
                {
                    if(obj.Chair.Id == 0)
                    {
                        _context.Chairs.Add(obj.Chair);
                    }
                    else
                    {
                        _context.Chairs.Update(obj.Chair);
                    }
                    _context.SaveChanges();
                    return RedirectToAction("Index");
                }
                return View(obj);
            }

            [HttpDelete]
            public IActionResult Delete(int id)
            {
                var chair =
                _context.Chairs.FirstOrDefault(x => x.Id == id);

                if (chair == null)
                {
                    return NotFound();
                }

                _context.Chairs.Remove(chair);
                _context.SaveChanges();
                return RedirectToAction("Index");
            }
        }
    }
}

```

Лістинг CourseController.cs:

```

namespace
AcademicDisciplinesGA.Areas.Admin.Controllers
{
    [Area("Admin")]
    [Authorize(Roles = "Admin")]
    public class CourseController : Controller
    {
        private readonly ApplicationDbContext
        _context;

        public
        CourseController(ApplicationDbContext context)
        {
            _context = context;
        }
    }
}

```



```

        faculty = private readonly ApplicationDbContext
_context.Faculties.FirstOrDefault(x => x.Id == _context;
id);

        return View(faculty);
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public IActionResult Upsert(Faculty obj)
    {
        if (ModelState.IsValid)
        {
            if (obj.Id == 0)
            {
                _context.Faculties.Add(obj);
            }
            else
            {
                _context.Faculties.Update(obj);
            }
            _context.SaveChanges();
            return RedirectToAction("Index");
        }
        return View(obj);
    }

    [HttpDelete]
    public IActionResult Delete(int id)
    {
        var faculty =
_context.Faculties.Find(id);

        if (faculty == null)
        {
            return NotFound();
        }

        _context.Faculties.Remove(faculty);
        _context.SaveChanges();
        return RedirectToAction("Index");
    }
}

Лістинг TeacherController.cs:
namespace
AcademicDisciplinesGA.Areas.Admin.Controllers
{
    [Area("Admin")]
    [Authorize(Roles = "Admin")]
    public class TeacherController : Controller
    {
        public
TeacherController(ApplicationDbContext context)
        {
            _context = context;
        }

        public IActionResult Index()
        {
            IEnumerable<Teacher> teacherList =
_context.Teachers.ToList();
            return View(teacherList);
        }

        public IActionResult Upsert(int? id)
        {
            Teacher teacher = new();

            if (id == 0 || id == null)
            {
                return View(teacher);
            }
            else
            {
                teacher =
_context.Teachers.FirstOrDefault(x => x.Id == id);
                return View(teacher);
            }
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public IActionResult Upsert(Teacher obj)
        {
            if (ModelState.IsValid)
            {
                if (obj.Id == 0)
                {
                    _context.Teachers.Add(obj);
                }
                else
                {
                    _context.Teachers.Update(obj);
                }
                _context.SaveChanges();
                return RedirectToAction("Index");
            }
            return View(obj);
        }

        [HttpDelete]
        public IActionResult Delete(int id)
        {

```

```

        var teacher =
_context.Teachers.FirstOrDefault(x => x.Id == id);
        if (teacher == null)
        {
            return NotFound();
        }
        _context.Teachers.Remove(teacher);
        _context.SaveChanges();
        return RedirectToAction("Index");
    }
}

Лістинг ChairVM.cs:
namespace AcademicDisciplinesGA.Areas.Admin.Models
{
    public class ChairVM
    {
        public Chair Chair { get; set; }
        [ValidateNever]
        public IEnumerable<SelectListItem>
FacultyList { get; set; }
    }
}

Лістинг CourseVM.cs:
namespace AcademicDisciplinesGA.Areas.Admin.Models
{
    public class CourseVM
    {
        public Course Course { get; set; }
        [ValidateNever]
        public IEnumerable<SelectListItem>
TeacherList { get; set; }
        [ValidateNever]
        public IEnumerable<SelectListItem>
ChairList { get; set; }
    }
}

Лістинг Login.cs:
namespace
AcademicDisciplinesGA.Areas.Identity.Pages.Account
{
    public class LoginModel : PageModel
    {
        private readonly SignInManager<IdentityUser> _signInManager;
        private readonly ILogger<LoginModel>
_logger;
        public
LoginModel(SignInManager<IdentityUser>
signInManager, ILogger<LoginModel> logger)
        {
            _signInManager = signInManager;
            _logger = logger;
        }
        [BindProperty]
        public InputModel Input { get; set; }
        public
IList<AuthenticationScheme>
ExternalLogins { get; set; }
        public string returnUrl { get; set; }
    }
    [TempData]
    public string ErrorMessage { get; set; }
}

public class InputModel
{
    [Required]
    [EmailAddress]
    public string Email { get; set; }
    [Required]
    [DataType(DataType.Password)]
    public string Password { get; set; }
    [Display(Name = "Remember me?")]
    public bool RememberMe { get; set; }
}

    public async Task OnGetAsync(string
returnUrl = null)
    {
        if
(!string.IsNullOrEmpty(ErrorMessage))
        {
            ModelState.AddModelError(string.Empty,
ErrorMessage);
        }
        returnUrl ??= Url.Content("~/");
        await
HttpContext.SignOutAsync(IdentityConstants.Externa
lScheme);
        ExternalLogins = (await
_signInManager.GetExternalAuthenticationSchemesAsy
nc()).ToList();
        returnUrl = returnUrl;
    }
}

```

```

        public async Task<IActionResult>
OnPostAsync(string returnUrl = null)
    {
        returnUrl ??= Url.Content("~/");

        ExternalLogins = (await
_signInManager.GetExternalAuthenticationSchemesAsy
nc()).ToList();

        if (ModelState.IsValid)
        {
            var result = await
_signInManager.PasswordSignInAsync(Input.Email,
Input.Password, Input.RememberMe,
lockoutOnFailure: false);
            if (result.Succeeded)
            {
                _logger.LogInformation("User
logged in.");
                return
LocalRedirect(returnUrl);
            }
            if (result.RequiresTwoFactor)
            {
                return
RedirectToPage("./LoginWith2fa", new { ReturnUrl =
returnUrl, RememberMe = Input.RememberMe });
            }
            if (result.IsLockedOut)
            {
                _logger.LogWarning("User
account locked out.");
                return
RedirectToPage("./Lockout");
            }
            else
            {
                ModelState.AddModelError(string.Empty, "Invalid
login attempt.");
                return Page();
            }
        }

        return Page();
    }
}

```

Лістинг Logout.cs:

```

namespace
AcademicDisciplinesGA.Areas.Identity.Pages.Account
{
    public class LogoutModel : PageModel
    {

```

```

        private readonly
SignInManager<IdentityUser> _signInManager;
        private readonly ILogger<LogoutModel>
_logger;

        public
LogoutModel(SignInManager<IdentityUser>
signInManager, ILogger<LogoutModel> logger)
        {
            _signInManager = signInManager;
            _logger = logger;
        }

        public async Task<IActionResult>
OnPost(string returnUrl = null)
        {
            await _signInManager.SignOutAsync();
            _logger.LogInformation("User
logged out.");

            if (returnUrl != null)
            {
                return LocalRedirect(returnUrl);
            }
            else
            {
                return RedirectToPage();
            }
        }
    }
}

```

Лістинг Register.cs:

```

namespace
AcademicDisciplinesGA.Areas.Identity.Pages.Account
{
    public class RegisterModel : PageModel
    {
        private readonly
SignInManager<IdentityUser> _signInManager;
        private readonly UserManager<IdentityUser>
_userManager;
        private readonly IUserStore<IdentityUser>
_userStore;
        private readonly
IUserEmailStore<IdentityUser> _emailStore;
        private readonly ILogger<RegisterModel>
_logger;
        private readonly IEmailSender
_emailSender;
        private readonly RoleManager<IdentityRole>
_roleManager;
        private readonly ApplicationDbContext
_context;

        public RegisterModel(

```

```

    UserManager<IdentityUser> userManager,           public      IEnumerable<SelectListItem>
    IUserStore<IdentityUser> userStore,             RoleList { get; set; }
    SignInManager<IdentityUser>                    [ValidateNever]
signInManager,                                     public      IEnumerable<SelectListItem>
    ILogger<RegisterModel> logger,                  ChairList { get; set; }
    IEmailSender emailSender,                       }
    RoleManager<IdentityRole> roleManager,
    ApplicationDbContext context)
    {
        _userManager = userManager;
        _userStore = userStore;
        _emailStore = GetEmailStore();
        _signInManager = signInManager;
        _logger = logger;
        _emailSender = emailSender;
        _roleManager = roleManager;
        _context = context;
    }

    [BindProperty]
    public InputModel Input { get; set; }
    public string returnUrl { get; set; }
    public      IList<AuthenticationScheme>
ExternalLogins { get; set; }

    public class InputModel
    {
        [Required]
        [EmailAddress]
        [Display(Name = "Email")]
        public string Email { get; set; }

        [Required]
        [StringLength(100, ErrorMessage = "The
{0} must be at least {2} and at max {1} characters
long.", MinimumLength = 6)]
        [DataType(DataType.Password)]
        [Display(Name = "Password")]
        public string Password { get; set; }

        [DataType(DataType.Password)]
        [Display(Name = "Confirm password")]
        [Compare("Password", ErrorMessage =
"The password and confirmation password do not
match.")]
        public string ConfirmPassword { get;
set; }

        [Required]
        public string Name { get; set; }
        public string Role { get; set; }
        public int? ChairId { get; set; }

        [ValidateNever]
        public      IEnumerable<SelectListItem>
RoleList { get; set; }
        [ValidateNever]
        public      IEnumerable<SelectListItem>
ChairList { get; set; }
        public      async      Task      OnGetAsync(string
returnUrl = null)
        {
            if
(!_roleManager.RoleExistsAsync(RoleHelper.Role_Adm
in).GetAwaiter().GetResult())
            {
                _roleManager.CreateAsync(new
IdentityRole(RoleHelper.Role_Admin)).GetAwaiter().
GetResult();
                _roleManager.CreateAsync(new
IdentityRole(RoleHelper.Role_User)).GetAwaiter().G
etResult();
            }
            returnUrl = returnUrl;
            ExternalLogins      =      (await
_signInManager.GetExternalAuthenticationSchemesAsy
nc()).ToList();
            Input = new InputModel()
            {
                RoleList      =
_roleManager.Roles.Select(x => x.Name).Select(i =>
new SelectListItem { Text = i, Value = i }),
                ChairList      =
_context.Chairs.Select(x => new SelectListItem {
Text = x.Title, Value = x.Id.ToString() })
            };
        }
        public      async      Task<IActionResult>
OnPostAsync(string returnUrl = null)
        {
            returnUrl ??= Url.Content("~/");
            ExternalLogins      =      (await
_signInManager.GetExternalAuthenticationSchemesAsy
nc()).ToList();
            if (ModelState.IsValid)
            {
                var user = CreateUser();
                await
_userStore.SetUserNameAsync(user,      Input.Email,
Cancellation.Token.None);

```

```

        await
_emailStore.SetEmailAsync(user, Input.Email, RedirectToPage("RegisterConfirmation", new { email
CancellationToken.None); = Input.Email, returnUrl = returnUrl });
        user.Name = Input.Name;
        user.ChairId = Input.ChairId;
        var result = await
_userManager.CreateAsync(user, Input.Password);
        if (result.Succeeded)
        {
            _logger.LogInformation("User
created a new account with password.");
            if(Input.Role == null)
            {
                await
_userManager.AddToRoleAsync(user, RoleHelper.Role_User);
            }
            else
            {
                await
_userManager.AddToRoleAsync(user, Input.Role);
            }
            var userId = await
_userManager.GetUserIdAsync(user);
            var code = await
_userManager.GenerateEmailConfirmationTokenAsync(u
ser);
            code =
WebEncoders.Base64UrlEncode(Encoding.UTF8.GetBytes
(code));
            var callbackUrl = Url.Page(
                "/Account/ConfirmEmail",
                pageHandler: null,
                values: new { area =
"Identity", userId = userId, code = code,
returnUrl = returnUrl },
                protocol: Request.Scheme);
            await
_emailSender.SendEmailAsync(Input.Email, "Confirm
your email",
                $"Please confirm your
account by <a
href='{HtmlEncoder.Default.Encode(callbackUrl)}'>c
licking here</a>.");
            if
(_userManager.Options.SignIn.RequireConfirmedAccou
nt)
            {
                await
                return
                LocalRedirect(returnUrl);
                }
                foreach (var error in
result.Errors)
                {
                    ModelState.AddModelError(string.Empty,
error.Description);
                }
                return Page();
            }
            private ApplicationUser CreateUser()
            {
                try
                {
                    return
                    Activator.CreateInstance<ApplicationUser>();
                }
                catch
                {
                    throw
                    new
                    InvalidOperationException($"Can't create an
                    instance of '{nameof(IdentityUser)}'. " +
                    $"Ensure that
                    '{nameof(IdentityUser)}' is not an abstract class
                    and has a parameterless constructor, or
                    alternatively " +
                    $"override the register page
                    in
                    /Areas/Identity/Pages/Account/Register.cshtml");
                }
            }
            private IUserEmailStore<IdentityUser>
            GetEmailStore()
            {
                if (!_userManager.SupportsUserEmail)
                {
                    throw
                    new
                    NotSupportedException("The default UI requires a
                    user store with email support.");
                }
            }
        }
    }
}

```

```

    }
    return
(IUserEmailStore<IdentityUser>)_userStore;
    }
}

```

ЛІСТИНГ GeneticAlgorithmController.cs:

```

namespace
AcademicDisciplinesGA.Areas.User.Controllers
{
    [Area("User")]
    [Authorize]
    public class GeneticAlgorithmController :
Controller
    {
        DisciplinesPopulation population;
        private readonly ApplicationDbContext
_context;
        DisciplinesChromosome
disciplinesChromosomes;

        public
GeneticAlgorithmController(ApplicationDbContext
db)
        {
            _context = db;
        }

        public IActionResult Survey()
        {
            var claimsIdentity =
(ClaimsIdentity)User.Identity;
            var claims =
claimsIdentity.FindFirst(ClaimTypes.NameIdentifier
);
            var user =
_context.ApplicationUsers.FirstOrDefault(x => x.Id
== claims.Value);

            SurveyVM surveyVM = new()
            {
                User = user,
                ChairList =
_context.Chairs.Select(x => new SelectListItem {
Text = x.Title, Value = x.Id.ToString() }),
                TeacherList =
_context.Teachers.Select(x => new SelectListItem {
Text = x.Name, Value = x.Id.ToString() })
            };

            return View(surveyVM);
        }

        [HttpPost]

```

```

[ValidateAntiForgeryToken]
public IActionResult Survey(SurveyVM obj)
{
    if (ModelState.IsValid)
    {
        List<Teacher> teachers = new
List<Teacher>();
        List<Chair> chairs = new
List<Chair>();
        var user =
_context.ApplicationUsers.FirstOrDefault(x => x.Id
== obj.User.Id);

        teachers.Add(_context.Teachers.FirstOrDefault(x =>
x.Id == obj.Teacher.Id));
        var chairRec =
_context.Chairs.FirstOrDefault(x => x.Id ==
obj.Chair.Id);

        if (obj.ImproveExisting)
        {
            var facultyId =
_context.Chairs.Where(c => c.Id ==
user.ChairId).Select(c =>
c.FacultyId).FirstOrDefault();

            var facultyChairs =
_context.Chairs.Where(c => c.FacultyId ==
facultyId).ToList();

            chairs.AddRange(facultyChairs);

            if
(!chairs.Contains(chairRec))
            {
                chairs.Add(chairRec);
            }
        }
        else
        {
            chairs.Add(chairRec);
        }

        population = new
DisciplinesPopulation(_context, teachers, chairs);

        disciplinesChromosomes = Run();

        ViewData["Courses"] =
ConvertToCourses(disciplinesChromosomes.Sequence);

        //_context.Chairs.Add(obj.Chair);
        //_context.SaveChanges();
    }
}

```

```

        return View("Result");
    }
    return View(obj);
}

private DisciplinesChromosome Run()
{
    while (population.GenerationCount <
GAConfig.MaxGenerations
population.NoImprovementCount
GAConfig.MaxNoImprovementCount)
    {
        population.DoGeneration();
    }

    return population.GetBestIndividual();
}

public List<Course>
ConvertToCourses(List<CourseChromosome>
courseChromosomes)
{
    var courseTeacherIds =
courseChromosomes.Select(c
c.TeacherId).ToList();
    var courseChairIds =
courseChromosomes.Select(c => c.ChairId).ToList();

    var teachers =
_context.Teachers.Where(t
courseTeacherIds.Contains(t.Id)).ToList();
    var chairs = _context.Chairs.Where(c
=> courseChairIds.Contains(c.Id)).ToList();

    var courses =
courseChromosomes.Select(cc =>
{
    var course = new Course
    {
        Id = cc.Id,
        Title = cc.Title,
        ECTS = cc.ECTS,
        TeacherId = cc.TeacherId,
        ChairId = cc.ChairId
    };

    course.Teacher =
teachers.FirstOrDefault(t => t.Id ==
course.TeacherId);
    course.Chair =
chairs.FirstOrDefault(c => c.Id ==
course.ChairId);

    return course;
}).ToList();
}

return courses;
}
}
}

Лістинг SurveyVM.cs:
namespace AcademicDisciplinesGA.Areas.User.Models
{
    public class SurveyVM
    {
        [ValidateNever]
        public ApplicationUser User { get; set; }

        public bool ImproveExisting {get; set;}

        [ValidateNever]
        public IEnumerable<SelectListItem>
TeacherList { get; set; }

        [ValidateNever]
        public IEnumerable<SelectListItem>
ChairList { get; set; }

        [ValidateNever]
        public Teacher Teacher { get; set; }
        [ValidateNever]
        public Chair Chair { get; set; }
    }
}

Лістинг DisciplinesChromosome.cs:
namespace AcademicDisciplinesGA.GA
{
    public class DisciplinesChromosome
    {
        private readonly ApplicationDbContext
_dataContext;
        public List<CourseChromosome> Sequence {
get; set; }
        public int Rank { get; set; }
        public int ECTSCount { get; set; }
        public int TeacherFitness { get; set; }
        public int ChairFitness { get; set; }
        public List<Teacher> Teachers { get; set; }

        public List<Chair> Chairs { get; set; }

        public static int _length = 10;
        static Random Random = new Random();

        public
DisciplinesChromosome(ApplicationDbContext
dataContext, List<Teacher> teachers, List<Chair>
chairs)
        {

```

```

        _dataContext = dataContext;
        Generate(_length);
        Teachers = teachers;
        Chairs = chairs;
        ECTSCount = GetTotalECTS();
        TeacherFitness
IsTeacherSelected(Teachers);
        ChairFitness
IsChairSelected(Chairs);
    }
    public
DisciplinesChromosome(List<CourseChromosome>
courses, List<Teacher> teachers, List<Chair>
chairs)
    {
        Sequence = courses.ToList();
        Teachers = teachers;
        Chairs = chairs;
        ECTSCount = GetTotalECTS();
        TeacherFitness
IsTeacherSelected(Teachers);
        ChairFitness
IsChairSelected(Chairs);
    }

    public int GetTotalECTS()
    {
        var total = 0;

        for (int i = 0; i < Sequence.Count();
i++)
        {
            var fromCourse = Sequence[i];

            total += fromCourse.ECTS;
        }

        return total;
    }

    public int IsTeacherSelected(List<Teacher>
teachers)
    {
        var totalSelected = 0;
        for (int i = 0; i < Sequence.Count(); + 1);
i++)
        {
            for (int j = 0; j < (selectedCoursesIds.Contains(courseId));
teachers.Count; j++)
            {
                if
                (Sequence[i].TeacherId.Equals(teachers[j].Id))
                {
                    totalSelected++;
                }
            }
        }
    }
}

}

return totalSelected;
}

public void Generate(int length)
{
    var courses = _dataContext.Courses
.Include(course => course.Teacher)
.Include(course =>
course.Chair).ToList();
    var selectedCoursesIds = new
HashSet<int>();
    var result = new
List<CourseChromosome>();

    for (int i = 0; i < length; i++)
    {
        int courseId;
        do
        {
            courseId =
Random.Next(courses.First().Id, courses.Last().Id)
        }
        while
        selectedCoursesIds.Add(courseId);
        var selectedCourse =
courses.Find(c => c.Id.Equals(courseId));
        result.Add(new CourseChromosome()
        {
            Id = selectedCourse.Id,
            Title = selectedCourse.Title,

```

```

                ECTS = selectedCourse.ECTS,
                ChairId
selectedCourse.ChairId,
                TeacherId
selectedCourse.TeacherId
            });
        }

        Sequence = result;
    }

}

```

Лістинг DisciplinesPopulation.cs:

```

namespace AcademicDisciplinesGA.GA
{
    public class DisciplinesPopulation
    {
        private readonly ApplicationDbContext
_dataContext;
        public List<DisciplinesChromosome>
Population { get; set; }
        public List<Teacher> Teachers { get; set; }
    }

    public List<Chair> Chairs { get; set; }
    public int GenerationCount { get; set; } =
0;

    public List<int> FitnessOverTime { get;
private set; }
    public int NoImprovementCount { get;
private set; } = 0;
    public bool HasConverged =>
        GenerationCount >
GAConfig.MaxGenerations
        || NoImprovementCount >
GAConfig.MaxNoImprovementCount;

    private float previousConvergenceArea =
float.MaxValue;

    public DisciplinesPopulation(ApplicationDbContext
dbContext, List<Teacher> teachers, List<Chair>
chairs)
    {
        _dataContext = dbContext;
        Teachers = teachers;
        Chairs = chairs;
        FitnessOverTime = new List<int>();
        Spawn();
    }

    public void Spawn()
    {
                var result =
                = PopulationHelper.SpawnPopulation(_dataContext,
                Teachers, Chairs);
                Population = result;
            }

        public void DoGeneration()
        {
            GenerationCount++;

            var offspring = new
List<DisciplinesChromosome>();

            while (offspring.Count <
GAConfig.PopulationCount)
            {
                var mother = GetParent();
                var father = GetParent();

                while (mother == father)
                {
                    father = GetParent();
                }

                var (offspringA, offspringB) =
GetOffspring(mother, father);

                (offspringA, offspringB) =
Mutate(offspringA, offspringB);

                offspring.Add(offspringA);
                offspring.Add(offspringB);
            }

            Population.AddRange(offspring);

            MultiObjectiveHelper.UpdatePopulationFitness(Population);

            var newPopulation = new
List<DisciplinesChromosome>();

            foreach (var individual in
Population.OrderBy(i => i.Rank))
            {
                if
(!newPopulation.Contains(individual))
                {
                    newPopulation.Add(individual);
                }
            }
        }
    }
}

```

```

        newPopulation = Mutate(DisciplinesChromosome individualA,
newPopulation.Take (GACConfig.PopulationCount).ToList DisciplinesChromosome individualB)
t());
        {
            return
        Population.Clear();
        PopulationHelper.Mutate(individualA, individualB,
        _dataContext, Teachers, Chairs);
        newPopulation.ForEach(i => )
        Population.Add(i));
        private (DisciplinesChromosome,
        DisciplinesChromosome)
        var firstRank = Population.OrderBy(c DisciplinesChromosome)
=> c.ChairFitness).ThenBy(t => t.TeacherFitness);
        GetOffspring(DisciplinesChromosome individualA,
        var currentArea = DisciplinesChromosome individualB)
        MultiObjectiveHelper.CalculateArea(firstRank);
        {
            var offspringA =
            if (Math.Abs(previousConvergenceArea - DoCrossover(individualA, individualB);
            currentArea) < 0.1)
            var offspringB =
            {
                DoCrossover(individualB, individualA);
                NoImprovementCount++;
            }
            return (offspringA, offspringB);
        }
        else
        {
            NoImprovementCount = 0;
            private DisciplinesChromosome
            previousConvergenceArea = DoCrossover(DisciplinesChromosome individualA,
            DisciplinesChromosome individualB)
            currentArea;
        }
        {
            return
        }
        PopulationHelper.DoCrossover(individualA,
        public DisciplinesChromosome individualB, Teachers, Chairs);
        }
        GetBestIndividual()
        {
            DisciplinesChromosome bestChromosome = private DisciplinesChromosome GetParent()
            null;
            {
                var (candidate1, candidate2) =
                int maxFitness = int.MinValue;
                PopulationHelper.GetCandidateParents(Population);
                foreach (var chromosome in Population)
                {
                    return
                    if (chromosome.TeacherFitness + PopulationHelper.TournamentSelection(candidate1,
                    chromosome.ChairFitness > maxFitness)
                    candidate2);
                    {
                        }
                        maxFitness =
                    chromosome.TeacherFitness +
                    chromosome.ChairFitness;
                    }
                    bestChromosome = chromosome;
                }
            }
            return bestChromosome;
        }
        //var firstRank = Population.GroupBy(i
        => i.Rank).First().ToList();
        0.05;
        //return firstRank;
        public static int PopulationCount => 20;
        public static int MaxNoImprovementCount =>
        20;
        private (DisciplinesChromosome,
        DisciplinesChromosome)
    
```

Лістинг GAConfig.cs:

```

namespace AcademicDisciplinesGA.GA
{
    public static class GAConfig
    {
        public static int MaxGenerations => 1000;
        public static double MutationChance =>
        0.05;
        public static int PopulationCount => 20;
        public static int MaxNoImprovementCount =>
        20;
    }
}
    
```



```

    }

    public static DisciplinesChromosome
DoCrossover(DisciplinesChromosome individualA,
DisciplinesChromosome individualB, List<Teacher>
teachers, List<Chair> chairs, int
crossoverPosition = -1)
    {
        crossoverPosition = crossoverPosition
== -1
        ?
        random.Next(1,
individualA.Sequence.Count - 1)
        : crossoverPosition;

        var offspringSequence =
individualA.Sequence.Take(crossoverPosition).ToLis
t();
        var appeared =
offspringSequence.ToHashSet();

        foreach (var course in
individualB.Sequence)
        {
            if (appeared.Contains(course))
            {
                continue;
            }

            if (offspringSequence.Count ==
individualA.Sequence.Count)
            {
                break;
            }

            offspringSequence.Add(course);
        }

        return new DisciplinesChromosome(offspringSequence, teachers,
chairs);

        public static DisciplinesChromosome
DoMutate(DisciplinesChromosome individual,
ApplicationDbContext dataContext, List<Teacher>
teachers, List<Chair> chairs)
        {
            var courses = dataContext.Courses
.Include(course => course.Teacher)
.Include(course =>
course.Chair).ToList();

            var sequence = individual.Sequence;
            int randomIndex = random.Next(0,
sequence.Count);

            int newCourseId;
            CourseChromosome newCourse;

            do
            {
                newCourseId =
random.Next(courses.First().Id, courses.Last().Id
+ 1);
                var selectedCourse =
courses.Find(c => c.Id.Equals(newCourseId));
                newCourse = new CourseChromosome()
                {
                    Id = selectedCourse.Id,
                    Title = selectedCourse.Title,
                    ECTS = selectedCourse.ECTS,
                    ChairId =
selectedCourse.ChairId,
                    TeacherId =
selectedCourse.TeacherId
                };
            } while (sequence.Contains(newCourse));

            sequence[randomIndex] = newCourse;

            return new
DisciplinesChromosome(sequence, teachers, chairs);
        }

        public static (DisciplinesChromosome,
DisciplinesChromosome)
Mutate(DisciplinesChromosome individualA,
DisciplinesChromosome individualB,
ApplicationDbContext dataContext, List<Teacher>
teachers, List<Chair> chairs)
        {
            var newIndividualA = new
DisciplinesChromosome(individualA.Sequence,
teachers, chairs);
            var newIndividualB = new
DisciplinesChromosome(individualB.Sequence,
teachers, chairs);

            if (random.NextDouble() <
GAConfig.MutationChance)
            {
                newIndividualA =
DoMutate(individualA, dataContext, teachers,
chairs);
            }

            if (random.NextDouble() <
GAConfig.MutationChance)
            {

```

```

        newIndividualB
    DoMutate(individualB,    dataContext,    teachers,    protected    override    void
    chairs);
    }
    return    (newIndividualA,
newIndividualB);
    }
}

```

Лістинг RoleHelper.cs:

```

namespace AcademicDisciplinesGA.Helpers
{
    public class RoleHelper
    {
        public const string Role_Admin = "Admin";
        public const string Role_User = "User";
    }
}

```

Лістинг Slice.cs:

```

namespace AcademicDisciplinesGA.Helpers
{
    public struct Slice
    {
        public Slice(float xLower, float yLower,
float xUpper, float yUpper)
        {
            XLower = xLower;
            YLower = yLower;
            XUpper = xUpper;
            YUpper = yUpper;
        }

        public float XLower { get; }
        public float YLower { get; }
        public float XUpper { get; }
        public float YUpper { get; }
        public float Area => (XUpper - XLower) *
(YUpper - YLower);
    }
}

```

Лістинг ApplicationDbContext.cs:

```

namespace AcademicDisciplinesGA.Models
{
    public class ApplicationDbContext : IdentityDbContext
    {
        public
ApplicationDbContext(DbContextOptions<ApplicationD
bContext> options) : base(options)
        {
        }
    }
}

```

```

    OnModelCreating(ModelBuilder modelBuilder)
    {
        base.OnModelCreating(modelBuilder);
    }

    public DbSet<Chair> Chairs { get; set; }
    public DbSet<Course> Courses { get; set; }
    public DbSet<Faculty> Faculties { get;
set; }
    public DbSet<Teacher> Teachers { get; set;
}
    public
    DbSet<ApplicationUser>
ApplicationUsers { get; set; }
}

```

Лістинг ApplicationUser.cs:

```

namespace AcademicDisciplinesGA.Models
{
    public class ApplicationUser : IdentityUser
    {
        [Required]
        public string Name { get; set; }

        [Display(Name = "Chair")]
        public int? ChairId { get; set; }
        [ForeignKey("ChairId")]
        [ValidateNever]
        public Chair Chair { get; set; }
    }
}

```

Лістинг Chair.cs:

```

namespace AcademicDisciplinesGA.Models
{
    public class Chair
    {
        public int Id { get; set; }
        public string Title { get; set; }
        [Display(Name = "Faculty")]
        public int FacultyId { get; set; }
        [ValidateNever]
        public Faculty Faculty { get; set; }
    }
}

```

Лістинг Course.cs:

```

namespace AcademicDisciplinesGA.Models
{
    public class Course
    {
        public int Id { get; set; }
        public string Title { get; set; }
        public int ECTS { get; set; }
        public int TeacherId { get; set; }
    }
}

```

```

        [ValidateNever]
        public Teacher Teacher { get; set; }
        public int ChairId { get; set; }
        [ValidateNever]
        public Chair Chair { get; set; }
    }
}

```

Лістинг CourseChromosome.cs:

```

namespace AcademicDisciplinesGA.Models
{
    public struct CourseChromosome
    {
        public int Id { get; set; }
        public string Title { get; set; }
        public int ECTS { get; set; }
        public int ChairId { get; set; }
        public int TeacherId { get; set; }
    }
}

```

Лістинг Faculty.cs:

```

namespace AcademicDisciplinesGA.Models
{
    public class Faculty
    {
        public int Id { get; set; }
        public string Title { get; set; }
        [ValidateNever]
        public List<Chair> Chairs { get; set; }
    }
}

```

Лістинг Teacher.cs:

```

namespace AcademicDisciplinesGA.Models
{
    public class Teacher
    {
        public int Id { get; set; }
        public string Name { get; set; }
        [ValidateNever]
        public List<Course> Courses { get; set; }
    }
}

```

Лістинг FormedPool.cs:

```

namespace AcademicDisciplinesGA.Models
{
    public class FormedPool
    {
        public int Id { get; set; }
        public List<Course> FormedCoursesPool {
get; set; }
        public int UserId { get; set; }
        public ApplicationUser User { get; set; }
    }
}

```

```

}

```

Лістинг Program.cs:

```

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllersWithViews();
builder.Services.AddRazorPages();
builder.Services.AddDbContext<ApplicationDbContext>
(options =>
options.UseSqlServer(builder.Configuration.GetConn
ectionString("DefaultConnection")));
builder.Services.AddIdentity<IdentityUser,
IdentityRole>().AddDefaultTokenProviders().AddEnti
tyFrameworkStores<ApplicationDbContext>();
builder.Services.AddSingleton<IEmailSender,
EmailSender>();
builder.Services.ConfigureApplicationCookie(option
s =>
{
    options.LoginPath =
"/Identity/Account/Login";
    options.LogoutPath =
"/Identity/Account/Logout";
    options.AccessDeniedPath =
"/Identity/Account/AccessDenied";
});

var app = builder.Build();

if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Error");
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();
app.UseAuthentication();
app.UseAuthorization();

app.MapRazorPages();

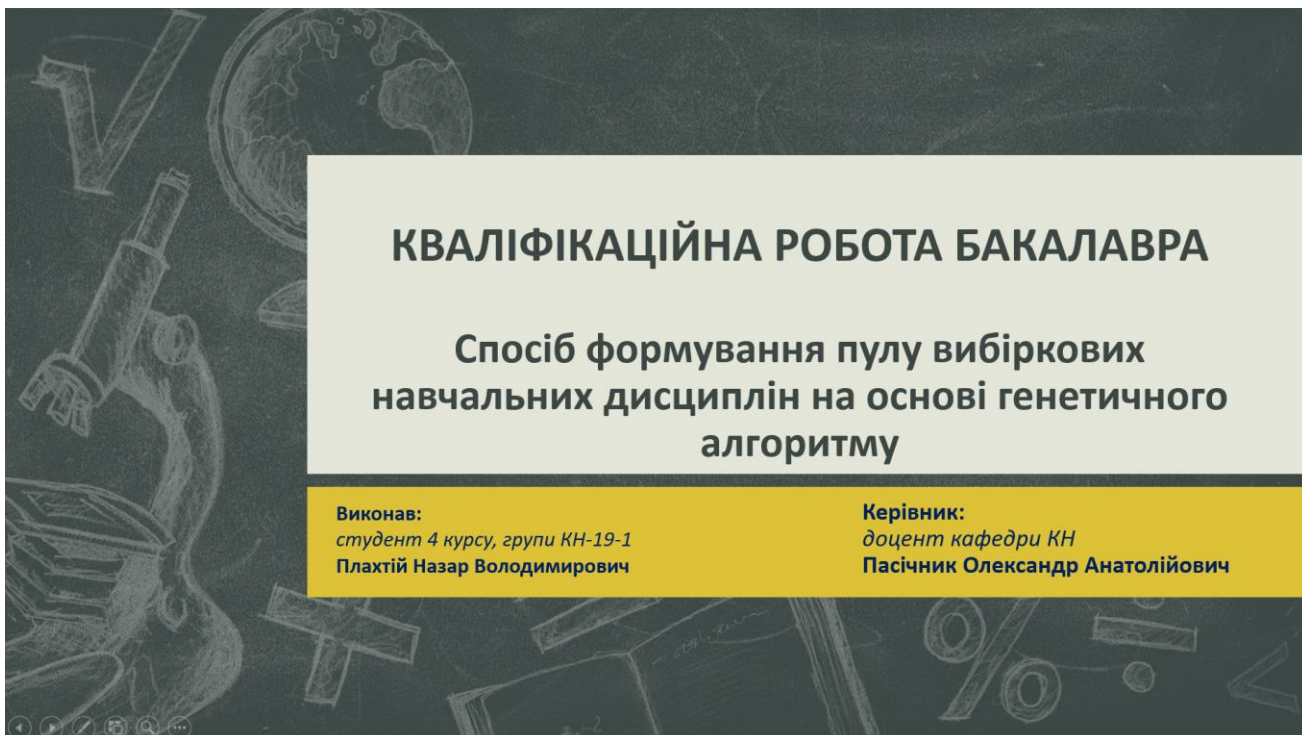
app.MapControllerRoute(
    name: "default",
    pattern:
"{area=Admin}/{controller=Home}/{action=Index}/{id
?}");

app.Run();

```

Додаток Б

Презентаційний матеріал



КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Спосіб формування пулу вибірових навчальних дисциплін на основі генетичного алгоритму

Виконав: студент 4 курсу, групи КН-19-1 Плахтій Назар Володимирович	Керівник: доцент кафедри КН Пасічник Олександр Анатолійович
--	--

Актуальність

Сучасні тенденції розвитку суспільства характеризуються широким використанням інформаційних технологій, які постійно удосконалюються і набувають більшого інтелектуального потенціалу.

Генетичні алгоритми є гнучким та ефективним інструментом для вирішення різноманітних практичних задач. Ще однією важливою тенденцією є широка модернізація системи освіти та переосмислення її ролі в суспільному житті. Це включає в себе впровадження нових підходів та парадигм формування освітнього процесу.

Основним принципом сучасної вищої освіти є студентоцентризований підхід, що дозволяє студентам брати активну участь у формуванні своєї освітньої траєкторії та надає їм широкі можливості вибору напрямку та рівня отримуваних знань та вмінь. Задоволення потреб студентів передбачає наявність обов'язкових та вибірових дисциплін у навчальній програмі.

Навчальні заклади пропонують різноманітні вибірові дисципліни, щоб задовольнити потреби студентів, наприклад, у Хмельницькому національному університеті їх кількість перевищує 1000. Визначення переліку вибірових дисциплін є складною задачею, оскільки ускладнюється різноманіттям побажань студентів щодо результатів навчання.

Мета і задачі роботи

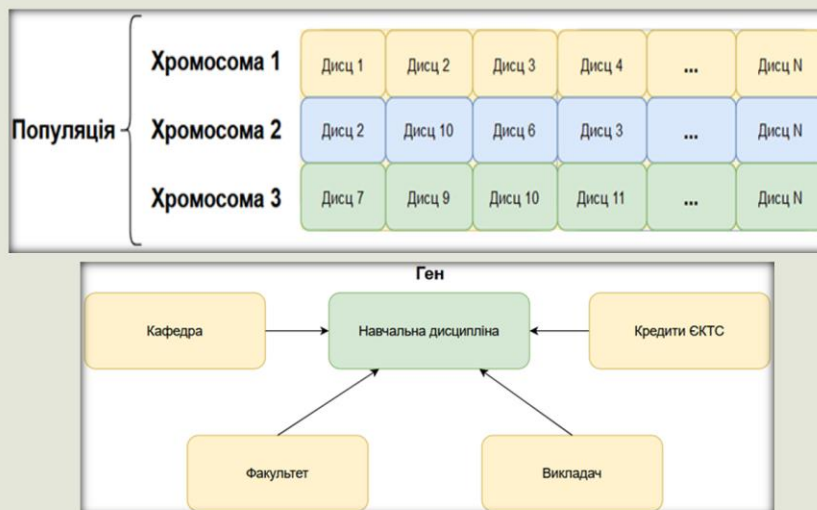
Метою кваліфікаційної роботи бакалавра є розробка методики формування набору вибірових навчальних дисциплін з використанням генетичного алгоритму. Для досягнення цієї мети були визначені наступні дослідницькі задачі:

- Провести аналіз різних методів формування набору вибірових дисциплін, їх мети та завдань.
- Оцінити можливості, переваги та недоліки генетичного алгоритму для використання у цьому контексті.
- Розробити інформаційну технологію визначення набору вибірових навчальних дисциплін на основі генетичного алгоритму.
- Провести експериментальне тестування розробленої інформаційної технології.

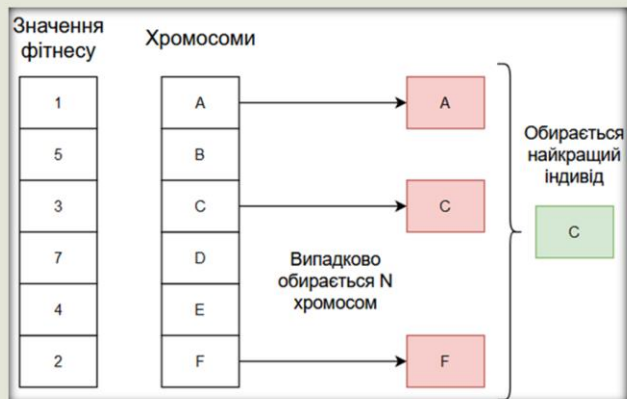
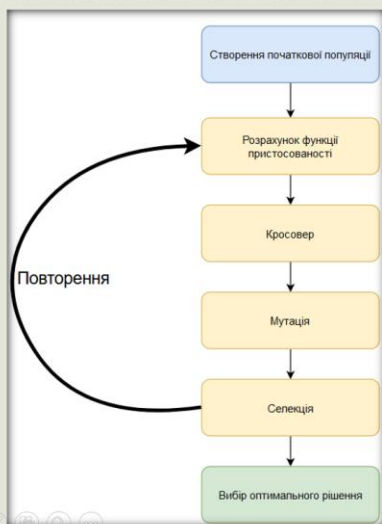
Об'єкт дослідження – процес побудови пулу вибірових навчальних дисциплін на основі генетичного алгоритму.

Предмет дослідження – моделі, алгоритми та засоби для створення способу пулу вибірових навчальних дисциплін на основі генетичного алгоритму.

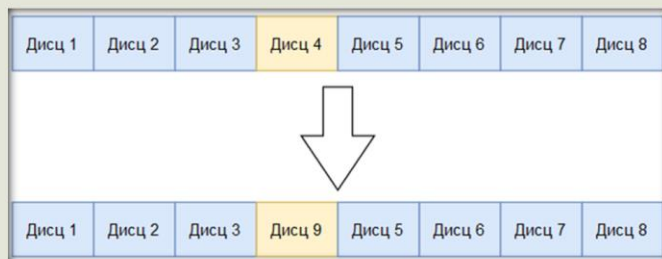
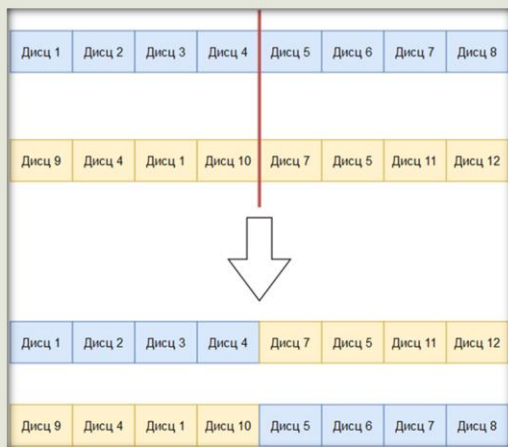
Спосіб формування пулу вибірових навчальних дисциплін на основі генетичного алгоритму



Схеми методів роботи генетичного алгоритму та формування початкової популяції



Етапи роботи генетичного алгоритму



Структура модулів системи та їх взаємозв'язок



Програмна реалізація способу формування пулу вибіркових навчальних дисциплін на основі генетичного алгоритму

The screenshot shows the 'Опитування' (Survey) page. It contains three questions:

- Питання 1: Як ви краще полюбляєте навчати в контексті вашої спеціальності чи отримувати навчання? (How do you prefer to learn in the context of your specialty or receive education?)
- Питання 2: Як напрямки та форми зайнятості цікавлять вас найбільше? (Which directions and forms of activity interest you the most?)
- Питання 3: Вий викладач вашої улюбленої? (Is the lecturer of your favorite?)

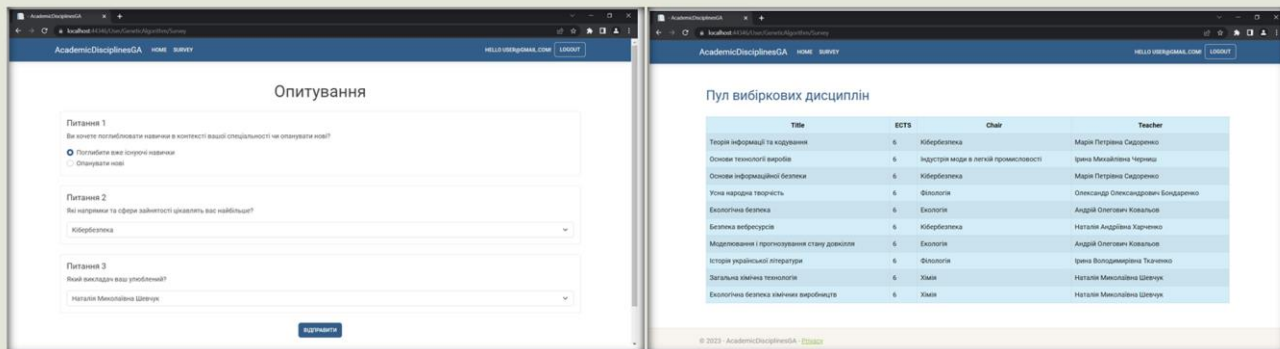
Each question has a text input field and a dropdown menu. A 'Відповісти' (Answer) button is at the bottom.

The screenshot shows the 'Пул вибіркових дисциплін' (Pool of elective disciplines) page, displaying a table of disciplines.

Title	ECTS	Chair	Teacher
Грунтознавство	6	Екологія	Оксана Михайлівна Степанченко
Теорія інформації та кодвання	6	Кибербезпека	Марія Петрівна Садорченко
Основи проєктування виробів	6	Індустрія моди в легкій промисловості	Ірина Михайлівна Черниш
WEB-технології та WEB-дизайн	6	Комп'ютерна наука	Вікторія Олександрівна Михайлова
Архітектура комп'ютера	6	Інженерія програмного забезпечення	Сергій Васильович Данилюк
Гідрологія	6	Екологія	Оксана Михайлівна Степанченко
Основи технології виробів	6	Індустрія моди в легкій промисловості	Ірина Михайлівна Черниш
Об'єктно-орієнтоване програмування	6	Комп'ютерна наука	Олександр Іванович Лисенко
Кольоровознавство	6	Дизайн	Євгенія Володимирівна Мельник
Проєктування баз даних	6	Комп'ютерна наука	Олександр Іванович Лисенко

Приклад № 1

Програмна реалізація способу формування пулу вибіркових навчальних дисциплін на основі генетичного алгоритму



Приклад № 2

Висновки

В результаті виконання кваліфікаційної роботи бакалавра на тему «Спосіб формування пулу вибіркових навчальних дисциплін на основі генетичного алгоритму» було розроблено ефективний метод оптимізації процесу вибору навчальних дисциплін студентами.

У процесі реалізації програмного модуля були враховані функціональні можливості, які необхідні для забезпечення ефективної взаємодії користувачів з системою. Модуль авторизації забезпечує безпечний доступ до системи з різними ролями, а модуль CRUD операцій дозволяє адміністраторам додавати, оновлювати та видаляти дані про факультети, кафедри, викладачів та дисципліни.

Особлива увага була приділена модулю генерації пулу вибіркових дисциплін, який обробляє дані опитування користувача та застосовує генетичний алгоритм для вибору оптимального пулу. Цей модуль дозволяє забезпечити максимальну персоналізацію вибору дисциплін, враховуючи індивідуальні вподобання студентів.

Отже, в результаті було розроблено ефективний спосіб формування пулу вибіркових навчальних дисциплін на основі генетичного алгоритму, що може бути використаний в інформаційній системі формування навчальних планів.

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 6%**

ID: 114257 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА Додано в БД: 2023-05-29 Автора: Н.В. Плахтій Керівники: О.А. Пасічник Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	54793	852	1538 (3%)	27 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:
Кафедра КН

Дата перевірки:
29.05.2023 20:14:10 EEST

Дата звіту:
29.05.2023 20:16:16 EEST

ID перевірки:
1015304763

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005671

Назва документа: КН-19-1 Плахтій

Кількість сторінок: 61 Кількість слів: 8288 Кількість символів: 64863 Розмір файлу: 2.04 MB ID файлу: 1014976252

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

10.8% Схожість

Найбільша схожість: 2.22% з джерелом з Бібліотеки (ID файлу: 1011230523)

10.6% Джерела з Інтернету

666

Сторінка 63

4.63% Джерела з Бібліотеки

162

Сторінка 67

1% Цитат

Цитати

8

Сторінка 68

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

Модифікації

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

Підозріле форматування

18
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: Спосіб формування пулу вибіркового дисциплін на основі генетичного алгоритму

Автор: Плахтій Назарій Володимирович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

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

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, є законними і не є плагіатом, оскільки:

1) за програмою Anti-Plagiarism виявлені 1 %.

2) за програмою UNICHECK виявлені 10.8%; Найбільша схожість: 2.28% з Найбільша схожість: 2.22% з джерелом з Бібліотеки (ID файлу: 1011230523) яке містить матеріали огляду предметної області; інші схожості є фрагментарними – містять поширені конструкції, загальновідомі терміни, скорочення та визначення.

збігів/ідентичності/схожості, складає 1 % і 10.8% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КН

Олександр ПАСІЧНИК

Олександр МАЗУРЕЦЬ

Олександр БАРМАК



**ВІДГУК НАУКОВОГО КЕРІВНИКА
на кваліфікаційну роботу бакалавра**

студента *гр. КН-19-1 Плахтія Назарія Володимировича*

за темою Спосіб формування пулу вибіркового навчальних дисциплін на основі генетичного алгоритму

1. Актуальність теми

Панівною тенденцією розвитку сьогодення є широке, практично всеохоплююче запровадження інформаційних технологій зі стійким трендом до їх поглибленої інтелектуалізації. Гнучким та ефективним кібернетичним інструментом вирішення достатнього широкого кола практичних задач є генетичні алгоритми. Іншою панівною тенденцією є широка модернізація системи освіти, переосмислення її місця та ролі у суспільному житті. Це супроводжується запровадженням нової парадигми та новітніх підходів щодо формування змісту освітнього процесу. Основним принципом сучасної вищої освіти є студентоцентризований підхід. Практична реалізація зазначеного підходу передбачає наявність в освітній програмі обов'язкових та вибіркового навчальних дисциплін. Широка номенклатура вибіркового дисциплін робить нетривіальною задачу визначення їх переліку, що додатково ускладнюється різноманіттям прагнень здобувачів вищої освіти щодо результатів навчання. Розробка такого способу є актуальною задачею комп'ютерних наук.

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 способу формування пулу вибіркового дисциплін на основі генетичного алгоритму та розробка програмної системи реалізації зазначеного способу. При вирішенні поставленої задачі використано відповідний алгоритми розв'язання задачі, а саме генетичний алгоритм, що дозволило, як реалізувати сам спосіб формування пулу вибіркового дисциплін, так, й розробити відповідну програмну систему. 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 Плахтій Назарій Володимирович проявив себе кваліфікованим фахівцем та дисциплінованим студентом, вчасно виконуючи поставлені етапи дослідження. Як в процесі написання пояснювальної записки, так і при розробці прикладного програмного забезпечення проявив достатні для одержання успішного результату компетентності та результати навчання. Опанував загальні та фахові компетентності за напрямком «Комп'ютерні науки».

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 реалізації 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 розроблено програмне забезпечення для валідації та верифікації запропонованого способу.

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 спосіб та його програмна реалізація може бути використана гарантами освітніх програм, кураторами академічних груп та здобувачами вищої освіти для формування пулу вибіркових дисциплін, який найкраще відповідає вимогам та потребам студента.

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 Олександр ПАСІЧНИК



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КН-19-1 Плахтія Назарія Володимировича

за темою: Спосіб формування пулу вибіркових навчальних дисциплін на основі генетичного алгоритму

1. Актуальність обраної теми

Сучасний етап розвитку цивілізації характеризується новою промисловою революцією, суть якої полягає у тотальній інтелектуалізації. Відповіддю системи освіти на цей виклик є запровадження студентоцентрованого підходу, одним із проявів якого є впровадження у навчальні плани вибіркових дисциплін. Різноманіття можливих цілей при опануванні вибіркових дисциплін та їх великий перелік визначають нетривіальність задачі вибору, а відповідний інструментарій дозволяє реалізувати індивідуальний, персоналізований підхід з максимальним урахуванням потреб та прагнень здобувача вищої освіти.

2. Повнота розкриття мети та завдань роботи

В кваліфікаційній роботі повністю розкрито мета роботи та виконано усі поставлені завдання. А саме проведено аналіз методів формування пулу вибіркових дисциплін, можливостей генетичного алгоритму; реалізовано інформаційну технологію запропонованого способу та проведено її експериментальне тестування.

3. Зміст кожного розділу роботи

В першому розділі виконано аналіз предметної області, а саме сучасних підходів формування пулу вибіркових дисциплін в ЗВО. Сформульовано мету та визначено задачі. В другому розділі розроблено спосіб формування пулу вибіркових навчальних дисциплін на основі генетичного алгоритму з урахуванням особливостей його імплементації, визначено функціональну структуру інформаційної системи.

В третьому розділі виконана програмна реалізація способу та проведено її експериментальне тестування

4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблений у роботі спосіб та його програмна реалізація може бути використана практично всіма групами стейкхолдерів для формування пулу вибіркових дисциплін, який найкраще відповідає вимогам та потребам.

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 кваліфікаційній роботі не відзначено, як враховується послідовність вивчення вибіркових дисциплін протягом всього періоду отримання вищої освіти за відповідним освітнім рівнем

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

Доц. каф ТМІТ (КНУ) Олександр [підпис]