

Ім'я користувача:  
Владислав Петрович Лещук

ID перевірки:  
1016363720

Дата перевірки:  
15.06.2024 19:11:55 EEST

Тип перевірки:  
Doc vs Library

Дата звіту:  
15.06.2024 19:37:32 EEST

ID користувача:  
5115

Назва документа: 2024\_Leshchuk\_126\_bakalavr.docx

Кількість сторінок: 110 Кількість слів: 17459 Кількість символів: 140679 Розмір файлу: 2.77 MB ID файлу: 1016169204

## 28.3% Схожість

Найбільша схожість: 6.1% з джерелом з Бібліотеки (ID файлу: 1004050487)

Пошук збігів з Інтернетом не проводився

28.3% Джерела з Бібліотеки 1000 ..... Сторінка 112

## 1.83% Цитат

Цитати 20 ..... Сторінка 113

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел

## Модифікації

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

Замінені символи 33

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

Національний університет водного господарства та природокористування  
Навчально-науковий інститут кібернетики, інформаційних технологій  
та інженерії

Кафедра комп'ютерних технологій та економічної кібернетики

**Допущено до захисту:**

Завідувач кафедри

\_\_\_\_\_ д. е. н., професор П. М. Грицюк

« \_\_\_\_\_ » \_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття ступеня «бакалавр»

за освітньо-професійною програмою «Інформаційні системи та технології»

спеціальності 126 «Інформаційні системи та технології»

на тему «Інформаційна система обліку неформальної освіти НУВГП»

**Виконав:**

Здобувач вищої освіти 4 курсу, групи ІСТ-41

**Лещук Владислав Петрович**

**Керівник:**

к. т. н, доцент Джоші О.І.

**Рецензент:**

к. е. н., доцент Бабич Т.Ю.

Рівне – 2024

## АНОТАЦІЯ

Лещук В.П. Інформаційна система обліку неформальної освіти НУВГП. Кваліфікаційна робота на здобуття освітнього ступеня «бакалавр»: 76 ст., 53 рис., 13 табл., 4 додатки на 32 ст., 24 літературних джерел.

Об'єкт дослідження – неформальна освіта в НУВГП.

Предмет досліджень – інформаційна система обліку неформальної освіти.

Методи дослідження – методи аналізу джерел інформації, системного аналізу, класифікації, моделювання, системного підходу, а також програмування та тестування програмної реалізації та ефективності системи.

Кваліфікаційна робота присвячена розробці інформаційної системи на основі реляційної системи управління баз даних MS Access, крос-платформній розробці з використанням мови програмування C#. Результатом роботи стало дослідження особливостей створення бази даних для обліку документів неформальної освіти, зроблений професійний огляд основних підходів вирішення задачі, що будуть використані для розробки застосунку. Розглянуто основні патерни розробки та рекомендації, щодо побудови схожої архітектури таких застосунків.

**Ключові слова:** неформальна освіта, інформаційна система, НУВГП, технології управління освітою, автоматизація процесів, бази даних, крос-платформна розробка, C# програмування, Windows Forms, MS Access.

## ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ СТАНУ ВИВЧЕННЯ ПРОБЛЕМИ	
.....	7
1.1 Аналіз предметної області та особливості етапів побудови ІС.....	7
1.2 Аналіз та вибір мови програмування.....	16
1.3 Аналіз та вибір СУБД.....	24
РОЗДІЛ 2. ОПИС МОДЕЛІ ТА МЕТОДІВ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ НЕФОРМАЛЬНОЇ ОСВІТИ.....	31
2.1 Розробка концептуальної моделі.....	31
2.2 ER-модель та нормалізація відношень.....	41
РОЗДІЛ 3. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ.....	47
3.1 Backend.....	47
3.2 Frontend.....	52
3.3 Запити інформаційної системи.....	66
ВИСНОВКИ.....	71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	74
ДОДАТКИ.....	77

## ВСТУП

Сучасний швидкозмінний світ вимагає постійного вдосконалення освітніх підходів та систем управління освітнім процесом. Однією з ключових сфер у цьому контексті є неформальна освіта, яка стає все більш актуальною в умовах швидкого технологічного розвитку та змін у суспільстві. Вона включає в себе різноманітні навчальні заходи, такі як курси, семінари, тренінги, які можуть бути важливим елементом у професійному розвитку студентів. Однак існує проблема ефективного обліку цієї форми освіти, що ускладнює контроль та аналіз результативності таких програм.

На фоні цієї проблеми виникає необхідність створення інформаційної системи обліку неформальної освіти, яка забезпечуватиме збір, аналіз та візуалізацію даних щодо участі студентів у позааудиторних освітніх програмах. Ця система має на меті підвищення ефективності управління неформальною освітою в Національному університеті водного господарства та природокористування (НУВГП), сприяючи оптимізації процесів прийняття рішень та вдосконаленню якості освіти.

Однак, незважаючи на існуючі дослідження в галузі управління освітою та розробки інформаційних систем, проблема обліку неформальної освіти залишається недостатньо вивченою та вирішеною. Наше дослідження спрямоване на розв'язання цієї невирішеної проблеми шляхом розробки та впровадження інформаційної системи обліку неформальної освіти в НУВГП. Відповідно до актуальності і предметної області, де один сертифікат деякого здобувача освіти закриває одну дисципліну, виникає необхідність розробки та впровадження інформаційної системи обліку неформальної освіти.

Метою даної роботи є розробка та реалізація інформаційної системи, яка забезпечить зручний та ефективний облік неформальної освіти, спростить процес аналізу даних та допоможе приймати обґрунтовані управлінські рішення

на основі отриманих результатів в Національному університеті водного господарства та природокористування (НУВГП).

*Об'єктом дослідження* є неформальна освіта в НУВГП, а предметом дослідження - розробка інформаційної системи обліку цієї освіти.

*Методи дослідження* включають аналіз джерел інформації, системний аналіз, класифікації, моделювання, системного підходу, а також програмування та тестування програмної реалізації та ефективності системи.

При цьому, основними завданнями дослідження є:

- аналіз сучасного стану проблеми обліку неформальної освіти;
- аналіз та вибір моделі життєвого циклу розробки ІС обліку неформальної освіти НУВГП; мови програмування та СУБД;
- розробка концептуальної моделі ІС;
- розробка ER-моделі відповідно до вимог 2НФ або 3НФ;
- реалізація backend-у та frontend-у ІС обліку неформальної освіти НУВГП: додавання файлів фотозображень сертифікатів, запити щодо обліку показників неформальної освіти та їх візуалізація;
- тестування та оцінка ефективності роботи ІС.

Ця робота має практичне значення для університетської спільноти та адміністрації, а також може послужити вагомим внеском у плани розвитку освіти на рівні кафедри, університету та державних програм загалом.

Логіка дослідження зумовила структуру кваліфікаційної роботи на здобуття ступеня «бакалавр»: вступ, три розділи, висновки, список використаних джерел.

В першому розділі аналізуються етапи створення інформаційної системи та їхні особливості, визначається оптимальна система керування базами даних, що використовуватиметься у нашій системі, та мова програмування, необхідна для реалізації; у другому розділі розглядається процес розробки і реалізації системи, включаючи розробку концептуальної моделі бази даних, нормалізацію відношень ER-моделі, а також реалізацію backend та frontend складових. У третьому розділі описується операційна діяльність системи, зокрема, виконання

6

запитів до бази даних та генерація звітів, а завершується робота висновками, що містять у собі результати досліджень та перелік виконаних завдань, і у заключному розділі наводиться список використаних джерел інформації.

Програмні продукти, що використовувались під час написання кваліфікаційної роботи: Microsoft Visual Studio, Microsoft Word, Microsoft Access, C#, Windows Forms, Photoshop, Corel PHOTO-PAINT.

## РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ СТАНУ ВИВЧЕННЯ ПРОБЛЕМИ

### 1.1 Аналіз предметної області та особливості етапів побудови ІС

*Неформальна освіта* — освіта, яка інституціоналізована, навмисна та запланована освітнім закладом. Визначальною характеристикою неформальної освіти є те, що вона є альтернативою та/або доповненням до формальної освіти в процесі навчання людей *протягом усього життя*. Це часто пропонується для гарантування права доступу до *освіти для всіх*. Вона обслуговує людей різного віку, але не обов'язково застосовує безперервну структуру шляху; вона може бути короткою за тривалістю та/або низькою інтенсивністю, і, як правило, вона надається у формі коротких курсів, семінарів або семінарів. Неформальна освіта здебільшого призводить до кваліфікацій, які не визнаються відповідними національними освітніми органами як формальні кваліфікації, або взагалі не мають кваліфікації. Неформальна освіта може охоплювати програми, що сприяють грамотності та освіті дорослих та молоді для дітей, які не навчаються, а також програми з життєвих навичок, трудових навичок та соціального чи культурного розвитку.

Неформальна освіта є частиною *концепції безперервної освіти (long life education)*, що дозволяє молодим людям і дорослим набувати і підтримувати на належному рівні вміння та компетенції, необхідні для адаптації в постійно змінному середовищі [14].

В структурі Національного університету водного господарства та природокористування відповідно до Закону України «Про освіту», «Про вищу освіту» та Статуту НУВГП функціонує Центр неформальної освіти [15], який здійснює організаційну, освітню та навчально-наукову діяльність в сфері неформальної освіти. Відповідно до Положення про неформальну та інформальну освіту в НУВГП [16] реалізується процедура визнання та порядок

зарахування результатів неформальної та інформальної освіти учасників формального освітнього процесу в університеті.

Тому на сьогоднішній день проектування і розробка єдиної інформаційної системи обліку неформальної освіти в НУВГП є актуальною задачею.

*Інформаційна система* - це організаційно-упорядкована взаємопов'язана сукупність засобів, і методів інформаційних технологій (ІТ), використовуваних для зберігання, обробки та видачі інформації в інтересах досягнення поставленої мети. Таке розуміння інформаційної системи припускає використання в якості основного технічного засобу переробки інформації ЕОМ і засобів зв'язку, що реалізують інформаційні процеси і видачу інформації, необхідної в процесі прийняття рішень задач з будь-якої області [1].

*Життєвий цикл* інформаційних систем - це невід'ємна частина їхнього існування, яка охоплює всі етапи, від їхнього створення до відключення. Цей цикл визначається не лише технічними аспектами, але й стратегічними та організаційними потребами. Розуміння цього циклу допомагає організаціям ефективно управляти інформаційними системами протягом їхнього життєвого шляху.

Життєвий цикл інформаційної системи характеризується періодом часу від ідеї створення інформаційної системи і закінчуючи моментом виведення її з експлуатації та включає в себе наступні стадії:

1. Передпроектне обстеження (планування);
2. Збір та аналіз вимог;
3. Проектування;
4. Створення інформаційної системи;
5. Тестування;
6. Супровід.

БД, як частина інформаційної системи, теж має свій життєвий цикл, що складається з таких етапів:

1. Планування БД;

2. Аналіз вимог до БД;
3. Проектування БД (концептуальне, логічне, фізичне);
4. Розробка застосунів;
5. Реалізація та завантаження даних;
6. Тестування;
7. Експлуатація [2].

Життєвий цикл інформаційних систем та їхніх баз даних, включаючи всі перелічені етапи наведено на рис. 1.1.

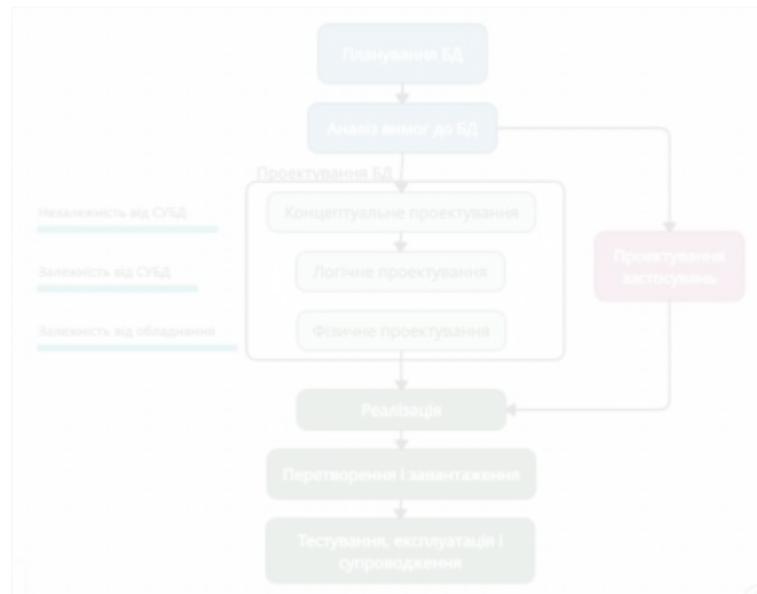


Рис. 1.1 Життєвий цикл ІС [2]

Найбільш популярними є наступні моделі життєвого циклу:

- каскадна (водоспадна) або послідовна;
- ітеративна і інкрементально-еволюційна (гібридна, змішана);
- спіральна (модель Боема).

**Каскадна (водоспадна) модель** - припускає строго послідовне (у часі) і однократне виконання всіх фаз проекту з жорстким (детальним) попереднім

плануванням в контексті наперед визначених або одного разу і цілком визначених вимог до програмної системи (рис. 1.2).

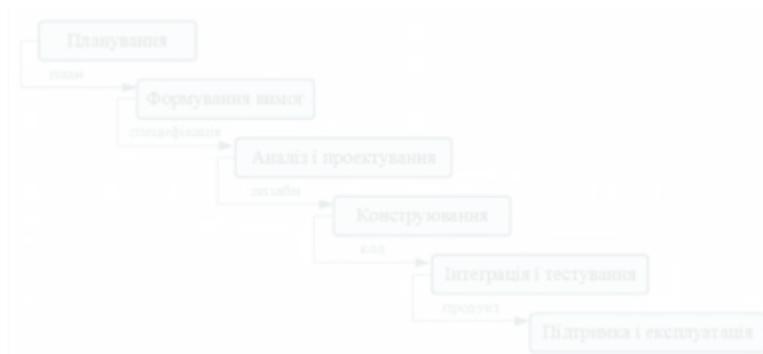


Рис. 1.2 Каскадна модель життєвого циклу ІС [1]

Практика показує, що в реальному світі, особливо у світі бізнес-систем, каскадна модель не повинна застосовуватися. Специфіка таких систем в тому, що вимоги характеризуються високою динамікою коригування та уточнення, неможливо чітко і однозначно визначити вимоги до початку робіт з реалізації (особливо, для нових систем) і швидкої мінливості вимог у процесі експлуатації системи.

В каскадній моделі перехід від однієї фази проекту до іншої припускає повну коректність результату (виходу) попередньої фази. Однак, наприклад, неточність будь-якої вимоги або некоректна її інтерпретація призводить до того, що доводиться «відкочуватися» до ранньої фази проекту, а необхідна переробка не просто вибиває проектну команду з графіка, але призводить до якісного зростання витрат і, не виключено, до припинення проекту в тій формі, в якій він спочатку замислювався. Крім того, ця модель не здатна гарантувати необхідну швидкість відгуку і внесення відповідних змін у відповідь на швидко змінні потреби користувачів, для яких програмна система є одним з інструментів виконання бізнес-функцій.

До основних переваг каскадної моделі відносяться:

- стабільність вимог протягом усього життєвого циклу розробки;

- можливість послідовного усунення виникаючих складнощів;
- визначеність і зрозумілість кроків моделі і простота її застосування;
- спрощення можливості здійснення планування, контролю та управління проектом;
- доступність для розуміння замовниками;
- ефективність для проектів з чіткими і зрозумілими, але важко реалізованими вимогами;
- ефективність для проектів з високими вимогами до якості при відсутності жорстких обмежень витрат і графіка робіт.

Недоліки каскадної моделі життєвого циклу:

- складність чіткого формулювання вимог на початку життєвого циклу і неможливість їх динамічної зміни на його протязі;
- послідовність лінійної структури процесу розробки, в результаті повернення до попередніх кроків для вирішення виникаючих проблем призводить до збільшення витрат і порушення графіка робіт;
- непридатність проміжного продукту для використання;
- неможливість гнучкого моделювання систем, що не мають аналогів;
- пізнє виявлення проблем, пов'язаних зі складанням, у зв'язку з одночасною інтеграцією всіх результатів в кінці розробки;
- недостатня участь користувача у створенні системи – тільки на самому початку (при розробці вимог) і в кінці (під час приймальних випробувань);
- неможливість попередньої оцінки якості системи користувачем;
- проблемність фінансування проекту, пов'язана зі складністю одноразової розподілу великих грошових коштів.

Область застосування каскадної моделі:

- Обмеження області застосування каскадної моделі визначається її недоліками. Її використання найбільш ефективно в таких випадках:

- при розробці проектів з чіткими, незмінними протягом ЖЦ вимогами, зрозумілими реалізацією і технічними методиками;
- при розробці проекту, орієнтованого на побудову системи або продукту такого ж типу, як вже розроблялися розробниками раніше;
- при розробці проекту, пов'язаного зі створенням і випуском нової версії вже існуючого продукту або системи;
- при розробці проекту, пов'язаного з перенесенням вже існуючого продукту на нову платформу;
- при виконанні великих проектів, в яких задіяно декілька великих команд розробників [3].

*Ітераційна модель життєвого циклу ІС* (поетапна модель з проміжним контролем) характеризується тим, що розробка ІС ведеться ітераціями з циклами зворотного зв'язку між етапами. Міжетапні коригування дозволяють враховувати реально існуюче взаємовплив результатів розробки на різних етапах; час життя кожного з етапів розтягується на весь період розробки.

Однак і ця схема (рис. 1.3) не дозволяє оперативного враховувати виникаючі зміни і уточнення вимог до системи. Узгодження результатів розробки з користувачами проводиться тільки в точках, планованих після завершення кожного етапу робіт, а загальні вимоги до ІС зафіксовані у вигляді технічного завдання на весь час її створення. Таким чином, користувачі часто отримують систему, не задовольняє їх реальним потребам [1].



Рис. 1.3 Ітераційна модель ЖЦ ІС [1]

#### Основні переваги ітераційної моделі розробки

- зниження ризиків – раннє виявлення конфліктів між вимогами, моделями та реалізацією проекту; велике фокусування на основних завданнях; динамічне формування вимог і управління ними.
- організація ефективного зворотного зв'язку проектною командою зі споживачем, створення продукту, реально відповідає його потребам.
- швидкий випуск мінімально цінного продукту (MVP) і можливість вивести продукт на ринок і почати експлуатацію набагато раніше.

#### Основні недоліки ітераційної моделі розробки:

- Проблеми з архітектурою і накладні витрати – при роботі з хаотичними вимогами і без опрацьованого глобального плану архітектура додатка може постраждати, а на її приведення до адекватного стану можуть знадобитися додаткові ресурси.
- Немає фіксованого бюджету і термінів, а також потрібна сильна залученість замовника в процес – для деяких замовників це неприйнятні умови співпраці з розробником, їм краще підійде водоспадна модель.

Ця модель застосовується для систем, в яких найбільш важливими є функціональні можливості, і які необхідно швидко продемонструвати на CASE-засобах [3].

Заключною моделлю життєвого циклу інформаційної системи є спіральна, що запропонована для подолання основних проблем каскадної моделі. Це найбільш відомий і поширений варіант еволюційної моделі, що став вже фактично самостійною моделлю, має різні сценарії розвитку і деталізації.

Відмінною особливістю цієї моделі є спеціальна увага ризикам, що впливає на організацію життєвого циклу.

Головне досягнення спіральної моделі полягає в тому, що вона пропонує спектр можливостей адаптації вдалих аспектів існуючих моделей процесів життєвого циклу.

Дана модель життєвого циклу характерна при розробці новаторських (нетипових) систем. На початку роботи над проектом у замовника і розробника немає чіткого бачення підсумкового продукту (вимоги не можуть бути чітко визначені) або стовідсоткової впевненості в успішній реалізації проекту (ризики дуже великі). В зв'язку з цим приймається рішення розробки системи по частинах з можливістю зміни вимог або відмови від її подальшого розвитку. Як видно з оригінальної спіральної моделі ЖЦ, розробленої Боемом (рис. 1.4), розвиток проекту може бути завершено не тільки після стадії впровадження, але і після стадії аналізу ризику [3].



- отримання більш надійної і стійкої системи, так як помилки і слабкі місця виявляються і виправляються на кожній ітерації;

- вдосконалення процесу розробки від ітерації до ітерації.

Основна проблема спірального циклу - визначення моменту переходу на наступний етап. Для її вирішення необхідно ввести тимчасові обмеження на кожен з етапів життєвого циклу. Перехід здійснюється відповідно до плану, навіть якщо не вся запланована робота закінчена. План складається на основі статистичних даних, отриманих в попередніх проектах, і особистого досвіду розробників.

Кожна з цих стадій створення систем передбачає виконання певного обсягу робіт, які подаються у вигляді процесів ЖЦ. Процес визначає як сукупність взаємопов'язаних дій, що перетворюють вхідні дані у вихідні. Опис кожного процесу включає в себе перелік вирішуваних завдань, вихідних даних і результатів.

Також існують сучасні моделі, прикладом яких є об'єктно-орієнтована модель.

Дана методологія припускає конструювання програмного рішення з готових об'єктів, для яких визначаються правила їх взаємодії, що переводять об'єкти з одного стану в інший. Така модель, що передбачає повну відповідність процесу розробки положенням об'єктно-орієнтованої методології (об'єктно-орієнтований аналіз, проектування, програмування), ефективна у великих проектах, а також там, де застосовуються так звані засоби швидкої розробки (RAD, Rapid Application Development), засновані на цих технологіях і містять готові бібліотеки класів.

Застосовується переважно в дуже великих проектах, де приділяється належна увага етапам аналізу і проектування, а також жорстко контролюється дотримання розробниками встановлених правил [1].

Кожен етап побудови інформаційної системи має свою вагому цінність. Також правильно спроектована та реалізована інформаційна система забезпечує

користувачам доступ до потрібної інформації вчасно та ефективно. Це може значно полегшити їхню роботу та покращити якість рішень.

Отже, відбір моделей інформаційних систем, дослідження їх складових елементів та врахування вимог і зовнішнього вигляду істотно впливають на підвищення ефективності та конкурентоспроможності організації. Для реалізації завдань та досягнення мети дослідження, щодо розробки інформаційної системи обліку неформальної освіти НУВГП, було обрано спіральну модель ЖЦ.

## 1.2 Аналіз та вибір мови програмування

**Мова програмування** - це формальний засіб комунікації, який використовується програмістами для створення програмного забезпечення. Вона надає набір синтаксичних правил і символів, за допомогою яких можна створювати програми, які потім виконуються на комп'ютері або іншому обчислювальному пристрої.

Мови програмування можна розділити на декілька категорій в залежності від їх призначення, синтаксису та парадигми програмування, яку вони підтримують. Наприклад, деякі мови призначені для розв'язання конкретних завдань, таких як обробка даних, веб-розробка, наукове програмування тощо, тоді як інші є загальнопризначеними і можуть бути використані для різних типів завдань.

Кожна мова має свої переваги та недоліки, і вибір конкретної мови зазвичай залежить від вимог проекту, особистих вподобань програміста та екосистеми розробки, що супроводжує мову.

Всі існуючі мови програмування можна поділити на дві групи (рис. 1.5):

- мови низького рівня;
- мови високого рівня.



Рис. 1.5 Класифікація мов програмування [17]

Низькорівневі мови програмування використовуються для прямого керування апаратним забезпеченням. Ці мови найбільш близькі до машинного коду та асемблеру, які використовуються для безпосереднього управління процесором. Деякі приклади низькорівневих мов програмування включають C, C++ та асемблер.

Більшість програмістів при складанні програм користуються деякою мовою високого рівня. Вони дозволяють програмістам писати програми на більш абстрактному рівні. Ці мови мають вбудовані функції та бібліотеки, які дозволяють виконувати різні завдання без необхідності писати код на низькорівневому рівні. Деякі приклади високорівневих мов програмування включають Python, Java, C# та PHP.

В свою чергу машинно-незалежні мови програмування (високого рівня) діляться на процедурно-орієнтовані, проблемно-орієнтовані та декларативні (процедурні).

**Процедурно-орієнтоване програмування** – це парадигма програмування, в якій програма розглядається як послідовність процедур або функцій. Основними елементами є процедури (функції) і дані, а управління

потокот виконання програми зазвичай відбувається за допомогою послідовних команд умовних операторів (if-then-else) та циклів. Прикладами таких мов є сучасні мови, які підтримують процедурно-орієнтоване програмування, включають C, Pascal, Fortran (у деяких аспектах).

**Проблемно-орієнтоване програмування** – це парадигма, в якій програми структуруються як набір зв'язаних висловлень, які описують бажаний результат, а не послідовність кроків, як у процедурному програмуванні. Програми зазвичай описують "що" має бути зроблено, а не "як" це зробити (SQL – мова структурованих запитів, HTML/CSS – мови опису структури та вигляду веб-сторінок, Prolog – мова логічного програмування).

**Процедурні мови програмування** вищого рівня використовують для опису мов, які поєднують в собі елементи процедурно-орієнтованого та об'єктно-орієнтованого програмування. Вони надають зручний інтерфейс для програмістів, що дозволяє писати програми на високому рівні абстракції, але можуть мати елементи процедурної парадигми в їхній основі.

Приклади мов: C#, Java, Python - ці мови включають елементи процедурного програмування, але також підтримують об'єктно-орієнтоване програмування та інші парадигми.

Мова програмування C# підтримує кілька рівнів програмування, які дозволяють розробникам працювати на різних рівнях абстракції, є повністю об'єктно-орієнтованою та має потужну систему подій і делегатів, що задовільняє вимогам щодо створення інформаційної системи обліку неформальної освіти НУВГП.

В сучасному світі вибір мови програмування визначається не лише технічними аспектами, але й стратегічними рішеннями, що мають значний вплив на розвиток проекту. У цьому розділі ми ретельно проаналізуємо різноманітні аспекти вибору мови програмування для реалізації нашої інформаційної системи. Починаючи з вимог до функціональності та ефективності програмного забезпечення і закінчуючи зручністю розробки та

підтримки, ми розглянемо різні альтернативи, що можуть відповісти нашим потребам найкраще.

Крім того, ми дослідимо потенційні переваги та обмеження кожної мови програмування у контексті наших цілей та завдань проекту. Ретельний аналіз дозволить обрати оптимальне програмне середовище, яке сприятиме успішному втіленню наших ідей та досягненню поставлених цілей.

Здійснивши аналіз індексу ТІОВЕ за травень 2024 року, ми можемо побачити що в трійці найбільш популярних мов програмування зараз є Python, C та C++ (табл. 1.1, рис. 1.6).

Таблиця 1.1

Індекс ТІОВЕ за травень 2024 року [9]

05.2024	05.2023	Зміна	Мова програмування	Рейтинг	Зміна %
1	1		 Python	16,33%	+2,88%
2	2		 C	9,98%	-3,37%
3	4		 C++	9,53%	-2,43%
4	3		 Java	8,69%	-3,53%
5	5		 C#	6,49%	-0,94%

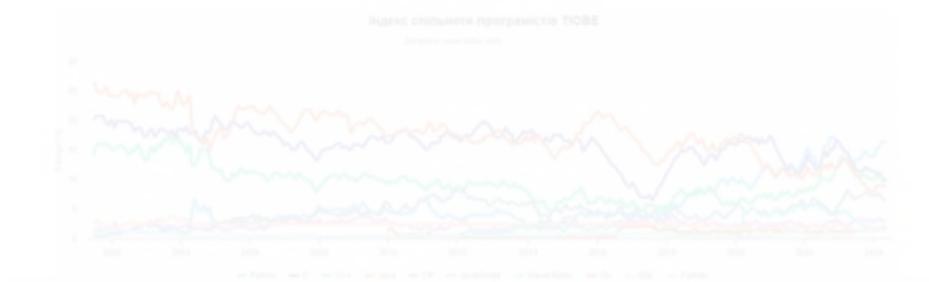


Рис. 1.6 Динаміка мов програмування за версією ТІОВ [9]

Також оцінивши рейтинг мов програмування «PYPL Popularity of Programming Language», ми бачимо інший список претендентів, що несуттєво відрізняється від ТІОВ (табл. 1.2).

Таблиця 1.2

## PYPL Popularity of Programming Language [10]

Worldwide, Jun 2024 :

<i>Rank</i>	<i>Language</i>	<i>Share</i>	<i>1-year trend</i>
1	Python	29.06 %	+1.4 %
2	Java	15.97 %	+0.2 %
3	JavaScript	8.7 %	-0.6 %
4	C#	6.73 %	-0.0 %

**Python** (рис. 1.7)— це високорівнева, універсальна мова програмування, яка визначається своєю простотою, читабельністю коду та багатством функціональних можливостей. Вона широко використовується у багатьох галузях, включаючи розробку веб-застосунків, наукові дослідження, штучний інтелект, аналіз даних, автоматизацію завдань та багато іншого.



Рис. 1.7 Логотип мови програмування Python

Однією з основних переваг Python є його простота та легкість вивчення. Синтаксис мови є дуже зрозумілим, що дозволяє швидко розпочати програмування навіть для початківців. Крім того, Python має величезну спільноту розробників, яка постійно поповнюється новими бібліотеками та фреймворками, що робить мову ще більш потужною та відповідальною на потреби сучасного програміста.

Іншою важливою характеристикою Python є його крос-платформенність. Код, написаний на Python, може бути запущений на будь-якій операційній системі без будь-яких змін, що робить його ідеальним вибором для проектів, що потребують переносимості.

Серед основних її переваг можна назвати такі:

- чистий синтаксис (для виділення блоків слід використовувати відступи);
- переносність програм (що властиве більшості інтерпретованих мов);
- стандартний дистрибутив має велику кількість корисних модулів (включно з модулем для розробки графічного інтерфейсу);
- можливість використання Python в діалоговому режимі (дуже корисне для експериментування та розв'язання простих задач);
- стандартний дистрибутив має просте, але разом із тим досить потужне середовище розробки, яке називається IDLE і яке написане мовою Python;
- зручний для розв'язання математичних проблем (має засоби роботи з комплексними числами, може оперувати з цілими числами довільної величини, у діалоговому режимі може використовуватися як потужний калькулятор);
- відкритий код (можливість редагувати його іншими користувачами).

Python має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис Python, динамічна обробка типів, а також те, що це інтерпретована мова, роблять її ідеальною для написання скриптів та швидкої розробки прикладних програм у багатьох галузях на більшості платформ [11].

C++ (рис. 1.8) — це потужна та універсальна мова програмування, яка використовується для розробки різноманітних програм, від низькорівневих системних додатків до великих програмних проектів. Вона є однією з найпопулярніших мов програмування і використовується в багатьох галузях,

таких як ігрова індустрія, фінансові технології, наукові дослідження, вбудовані системи та багато інших.



Рис. 1.8 Логотип мови програмування C++

Однією з головних переваг C++ є його висока ефективність. Вона надає прямий доступ до пам'яті комп'ютера та дозволяє ефективно управляти ресурсами системи, що робить її ідеальним вибором для розробки швидких та продуктивних програм.

Ще однією важливою особливістю C++ є його можливість використання об'єктно-орієнтованого програмування (ООП). Об'єктно-орієнтований підхід дозволяє розбити програму на невеликі, незалежні компоненти (об'єкти), що спрощує розробку та підтримку складних програмних систем.

**Java** (рис. 1.9) є однією з найбільш популярних та впливових мов програмування в світі сучасної програмної індустрії. Вона відома своєю простотою, надійністю та переносимістю, що робить її чудовим вибором для розробки широкого спектру програмних застосунків.

Однією з особливостей Java є її платформонезалежність. Код, написаний на Java, може бути запущений на будь-якій платформі, що підтримує відповідне віртуальне середовище Java (JVM).



Рис. 1.9 Логотип мови програмування Java

Ще однією перевагою Java є її велика екосистема та багатий набір бібліотек та фреймворків. Розробники можуть використовувати ці засоби для швидкої розробки програм та виконання різних завдань, від створення веб-додатків до розробки мобільних додатків та великих корпоративних систем [12].

Java також славиться своєю високою продуктивністю та масштабованістю. Вона дозволяє розробникам ефективно працювати з великими обсягами даних та високовимогливими додатками, забезпечуючи швидку та надійну роботу програм.

Загалом, ця мова програмування є потужним інструментом для розробки програмного забезпечення будь-якої складності та масштабу, і вона продовжує залишатися однією з найбільш важливих мов програмування у сучасній індустрії.

C# (вимовляється як "C sharp") (рис. 1.10) є потужною та високопродуктивною мовою програмування, розробленою компанією Microsoft. Вона відома своєю простотою використання, ефективністю та широким спектром застосувань, що робить її однією з найпопулярніших мов у світі програмування.



Рис. 1.10 Логотип мови програмування C#

Однією з переваг C# є її висока продуктивність та ефективність. Вона дозволяє розробникам швидко створювати якісне програмне забезпечення, забезпечуючи оптимальні рішення для різних завдань.

Обравши цю мову програмування для своєї дипломної роботи, я врахував кілька ключових факторів.

По-перше, C# є потужною та динамічно розвиваючою мовою програмування, яка має широкі можливості для створення різноманітних програмних рішень. Вона дозволяє швидко розробляти програми з високим рівнем продуктивності та ефективності.

Крім того, C# має розширену екосистему та велику спільноту розробників, що забезпечує доступ до багатьох корисних інструментів, бібліотек та фреймворків. Це дозволяє ефективно вирішувати різноманітні завдання та використовувати найновіші технології в проекті.

Також, враховуючи інтеграцію C# з платформою Windows, вибір цієї мови програмування стає особливо обґрунтованим, якщо проект спрямований на розробку додатків для операційних систем від Microsoft.

Отже, обравши C# для своєї дипломної роботи, я обрав потужну, ефективну та добре підтримувану мову програмування, яка дозволить реалізувати мої ідеї та цілі з високою якістю та ефективністю.

### 1.3 Аналіз та вибір СУБД

*Система управління базами даних* (СУБД, англ. Database Management System) — це комплекс програмних і лінгвістичних засобів, що призначені для створення, зберігання, управління та використання баз даних. Він забезпечує ефективне та надійне зберігання даних, а також надає користувачам зручний інтерфейс для роботи з ним [18].

Це програмне забезпечення дозволяє створювати, керувати та використовувати бази даних. Її основною функцією є забезпечити зручний та ефективний доступ до даних, а також забезпечити захист інформації від несанкціонованого доступу.

Системи керування базами даних класифікують за різними критеріями (рис. 1.11).

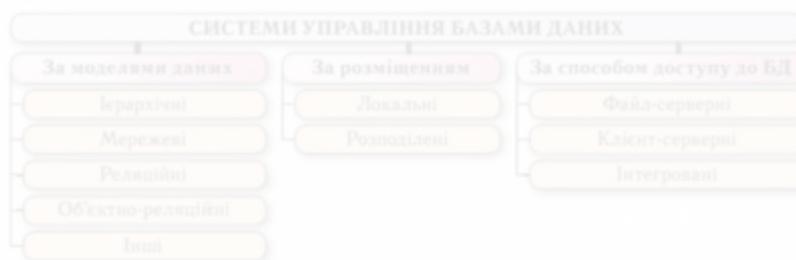


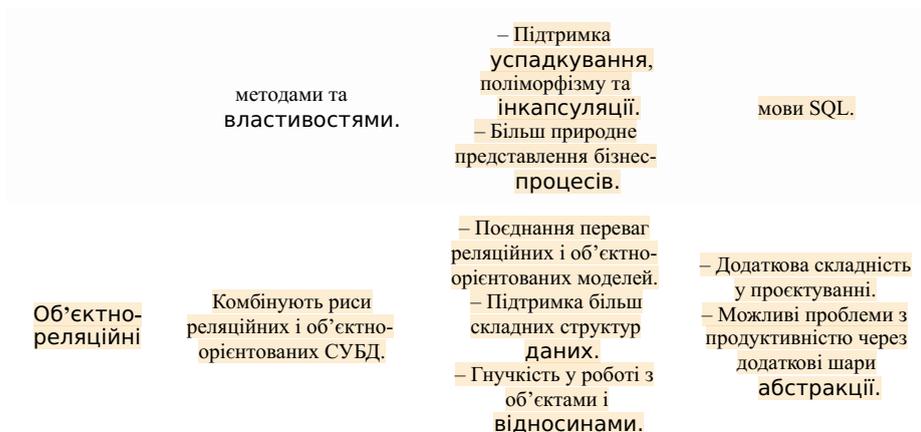
Рис. 1.11 Класифікація СУБД

Розглянемо основні СУБД за моделлю даних за допомогою таблиці 1.3. Розуміння переваг і недоліків кожного типу допомагає вибрати найкращий варіант залежно від конкретних потреб нашого проекту.

Таблиця 1.3

Класифікація СУБД за моделлю даних [18]

Тип СУБД	Опис	Переваги	Недоліки
Ієрархічні	Організують дані у вигляді ієрархії, де кожен запис має одного "батька" і багато "нащадків".	<ul style="list-style-type: none"> <li>– Простота в моделюванні зв'язків.</li> <li>– Ефективний доступ до деревоподібних структур даних.</li> </ul>	<ul style="list-style-type: none"> <li>– Обмеження в гнучкості структури даних.</li> <li>– Утруднення у вираженні складних відносин.</li> </ul>
Мережеві	Дають змогу встановлювати складні зв'язки між даними, створюючи структуру, схожу на граф.	<ul style="list-style-type: none"> <li>– Підтримка складних і взаємопов'язаних даних.</li> <li>– Гнучкіша, ніж ієрархічні, щодо зв'язків.</li> <li>– Сприяють поданню складних бізнес-процесів.</li> </ul>	<ul style="list-style-type: none"> <li>– Складність в управлінні структурою і підтримці запитів.</li> <li>– Складнощі в підтримці та обслуговуванні.</li> </ul>
Реляційні	Засновані на теорії відносин і таблиць. Кожна таблиця представляє окреме відношення.	<ul style="list-style-type: none"> <li>– Простота у використанні та моделюванні даних.</li> <li>– Незалежність даних від фізичної структури зберігання.</li> <li>– Велика підтримка SQL для запитів і маніпуляції даними.</li> </ul>	<ul style="list-style-type: none"> <li>– Продуктивність може страждати за великих обсягів даних.</li> <li>– Складність у поданні складних ієрархій даних.</li> </ul>
Об'єктно-орієнтовані	Працюють з об'єктами, де дані представлено у вигляді об'єктів із	<ul style="list-style-type: none"> <li>– Зручність у моделюванні складних об'єктних структур.</li> </ul>	<ul style="list-style-type: none"> <li>– Складність у міграції з традиційних СУБД.</li> <li>– Обмежена підтримка</li> </ul>



Після детального аналізу типів СУБД, ми впевнилися, що найбільш зручною для виконання нашого завдання по створенню бази даних для інформаційної системи обліку неформальної освіти НУВГП є реляційна база даних, оскільки вона надає нам можливість легко організувати дані у вигляді таблиць з рядками та стовпцями, що спрощує роботу з даними та забезпечує їх структурованість і зручність у використанні. Крім того, реляційні моделі дозволяють здійснювати складні запити до даних, забезпечують цілісність даних за допомогою зовнішніх ключів та відносин між таблицями, а також підтримують транзакційні операції для забезпечення консистентності та надійності даних.

Аналіз та вибір системи управління базами даних є важливим етапом у процесі проєктування інформаційної системи. Цей вибір може вплинути на продуктивність, надійність та функціональність системи.

За даними індексу найкращих баз даних TOPDB, рейтинг складається на основі частоти пошуку імен баз даних у Google, що відображає їхню популярність та популярність серед користувачів (таблиця 1.4).

Таблиця 1.4

Рейтинг TOPDB [4]

Ранг	Зміна	База даних	Поділіться	1-річний тренд
------	-------	------------	------------	----------------

1		Oracle	28,74 %	+2,0 %
2		MySQL	16,91 %	-1,4 %
3		SQL Server	11,72 %	-0,6 %
6	↓↓	Microsoft Access	6,0 %	-1,1 %

Так само рейтинг DB-Engines рейтингує системи керування базами даних за їх популярністю. Рейтинг оновлюється щомісяця. Їхній топ показаний у таблиці 1.5.

Таблиця 1.5

Рейтинг DB-Engines [5]

06.2024	Ранг		СУБД	Модель бази даних
	05.2024	06.2024		
<b>1.</b>	1.	1.	Oracle	Реляційна , мультивмодельна
<b>2.</b>	2.	2.	MySQL	Реляційна , мультивмодельна
<b>3.</b>	3.	3.	Microsoft SQL Server	Реляційна , мультивмодельна
<b>11.</b>	11.	9.	Microsoft Access	Реляційний

Різниця між рейтингом DB-Engines та TOPDB може бути зумовлена кількома факторами:

1. Методологія рейтингування: DB-Engines та Індекс найкращих баз даних TOPDB можуть використовувати різні методи рейтингування.
2. Джерела даних: Рейтинги можуть базуватися на різних джерелах даних. Наприклад, DB-Engines може використовувати дані з власного внутрішнього дослідження, тоді як Індекс найкращих баз даних TOPDB може опиратися на дані від пошукових систем або інших джерел.
3. Критерії включення: Різні рейтинги можуть враховувати різні критерії для включення баз даних у свій список.

4. Актуальність даних: Рейтинги можуть оновлюватися з різною частотою, що може призвести до різниці у показниках популярності баз даних в різний час.

Загалом, різниця у рейтингах може бути зумовлена різними методами збору та обробки даних, а також різними критеріями включення баз даних у рейтинг.

**СУБД Oracle** (рис. 1.12) – це сучасна система управління реляційними базами даних, що підтримує роботу в різних операційних середовищах. Система Oracle реалізує найсучасніші технології і підтримує багато можливостей, що дозволяє характеризувати її як досить потужну систему.



Рис. 1.12 Логотип СУБД Oracle

Для забезпечення роботи користувачів з базою даних Oracle потрібні фахівці, відповідальні за роботу всієї системи – адміністратори бази даних.

Наприклад, у функції адміністратора бази даних входить:

1. Створення екземпляра Oracle і його запуск;
2. Створення початкового варіанта БД і планування її подальшого розширення;
3. Реєстрація користувачів в системі, призначення їм привілеїв, ролей і профілів;
4. Відстеження роботи БД та вжиття заходів щодо оптимізації її функціонування;
5. Створення резервних копій БД і відновлення БД після збоїв [7].

База даних Oracle є однією з найбільш потужних та надійних систем управління базами даних на сучасному ринку. Її широкі можливості, надійність,

масштабованість та високий рівень безпеки роблять її популярним вибором для підприємств та організацій у всьому світі.

Наступною на черзі БД є *MySQL* (рис. 1.13). Вона являє собою структуровану сукупність даних. Ці дані можуть бути будь-якими - від простого списку майбутніх покупок до переліку експонатів картинної галереї або величезної кількості інформації в корпоративній мережі. Для запису, вибірки і обробки даних, що зберігаються в комп'ютерній базі даних, необхідна система управління базою даних, якою і є ПЗ *MySQL*.



Рис. 1.13 Логотип MySQL

*MySQL* - це система керування базами даних, ПЗ з відкритим кодом. У реляційній базі даних дані зберігаються не всі скопом, а в окремих таблицях, завдяки чому досягається вигреш в швидкості і гнучкості. Таблиці зв'язуються між собою за допомогою відносин, завдяки чому забезпечується можливість об'єднувати при виконанні запиту дані з декількох таблиць.

Тепер розглянемо СУБД, що використана у даному проекті - *Microsoft Office Access* (рис. 1.14).



Рис. 1.14 Логотип MS Access

*MS Access* є інструментом для керування базами даних, що входить до складу пакету офісних програм *Microsoft Office*. Вона дозволяє користувачам

легко створювати та управляти базами даних без необхідності великих знань в області програмування або баз даних та володіє кількома перевагами, які роблять її привабливим вибором для невеликих та середніх проектів.

Одна з головних переваг MS Access полягає в її легкості використання. Інтерфейс MS Access досить інтуїтивно зрозумілий, що дозволяє користувачам швидко створювати та налаштовувати бази даних.

Ще одна важлива перевага MS Access - це швидкість розробки. Вона має вбудовані шаблони, які дозволяють швидко створювати готові бази даних для різних цілей, таких як ведення списку клієнтів або відстеження запасів. Крім того, MS Access має можливості автоматизації, такі як макроси та модулі VBA, які дозволяють користувачам створювати складні функції та процедури.

Однак важливо відзначити, що MS Access краще підходить для невеликих та середніх проектів. Вона може стати обмеженням для великих обсягів даних або високої навантаженості. Проте, усупереч цим обмеженням, MS Access залишається популярним інструментом для багатьох користувачів та організацій.

Обираючи MS Access як СУБД для даного проекту, ми врахували усі ключові фактори, що визначили його відповідність нашим потребам. Таким чином, обрання MS Access є важливим кроком у напрямку успішної реалізації нашого проекту та досягнення його поставлених цілей.

## РОЗДІЛ 2. ОПИС МОДЕЛІ ТА МЕТОДІВ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ НЕФОРМАЛЬНОЇ ОСВІТИ

### 2.1 Розробка концептуальної моделі

Існує в основному три різні типи моделей баз даних (рис. 2.1): концептуальні моделі даних, логічні моделі даних і фізичні моделі даних, і кожна з них має певне призначення. Моделі баз даних використовуються для представлення даних і того, як вони зберігаються в базі даних, а також для встановлення зв'язку між елементами даних.

- **Концептуальна модель даних:** Ця модель даних визначає **ЩО** система містить. Цю модель зазвичай створюють бізнес-стейкхолдери та дані **Архітектів**. Мета полягає в тому, щоб організувати, охопити та визначити бізнес concepts і правила.
- **Логічна модель даних:** Визначає **ЯК** систему слід впроваджувати незалежно від СУБД. Ця модель зазвичай створюється **Data Архітектів** та бізнес-аналітиків. Метою є розробка технічної карти правил і структур даних.
- **Фізична модель даних:** Ця модель даних описує **ЯК** система буде реалізована за допомогою конкретної системи СУБД. Ця модель зазвичай створюється адміністратором баз даних і розробниками. Мета – фактичне впровадження бази даних.



Рис. 2.1 Типи моделей даних [19]

До бази даних інформаційної системи обліку неформальної освіти входить інформація про студентів, їхні навчально-наукові інститути, групи, освітні рівні, платформи MOOC та сертифікати. Концептуальну модель показано на рисунку 2.2.

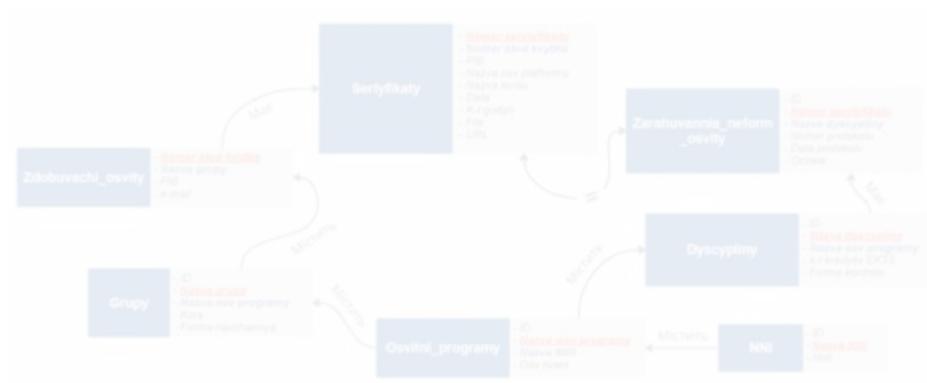


Рис. 2.2 Концептуальна модель бази даних інформаційної системи обліку неформальної освіти НУВГП

На рис. 2.2 написи червоним кольором позначають первинний ключ (Primary Key), написи синім кольором – зовнішній ключ (Foreign Key), написи чорним кольором – неключовий атрибут.

Аналіз підлеглості сутностей бази даних наступний:

1. ННІ:
  - Освітні програми - дочірня сутність;
2. Освітні програми:
  - ННІ - батьківська сутність;
  - Дисципліни - дочірня сутність;
  - Групи - дочірня сутність;
3. Дисципліни:

- Освітні програми - батьківська сутність;
- Зарахування неформальної освіти - дочірня сутність;

4. Групи (Groups):

- Освітні програми - батьківська сутність;
- Здобувачі освіти - дочірня сутність;

5. Здобувачі освіти:

- Групи - батьківська сутність;
- Сертифікати - дочірня сутність;

6. Сертифікати:

- Здобувачі освіти - батьківська сутність;
- Зарахування неформальної освіти - дочірня сутність;

7. Зарахування неформальної освіти:

- Сертифікати - батьківська сутність;
- Дисципліни - батьківська сутність.

Перейдемо до опису переліку сутностей та їх ролей в базі даних (табл. 2.1).

Таблиця 2.1

Перелік сутностей бази даних та їх ролей

Сутності (таблиці)	Її роль
<i>NNI</i> (Навчально-наукові інститути)	Зберігає інформацію про навчально-наукові інститути (ННІ), включаючи ID, назву та аббревіатуру ННІ, що допомагає структурувати організаційну інформацію навчального закладу, відслідковувати відповідальність за освітні програми та керувати адміністративною діяльністю.
<i>Osvitni_programy</i> (Освітні програми)	Зберігає інформацію про освітні програми, включаючи ID, назву освітньої програми, назву ННІ та освітній рівень, що допомагає структурувати навчальний процес.
<i>Dyscypliny</i> (Дисципліни)	Містить дані про дисципліни, такі як ID, назва дисципліни, назва освітньої програми, кількість кредитів ECTS та форма контролю.
<i>Grupy</i>	Містить дані про навчальні групи, такі як

(Групи)	ID, назва групи, назва освітньої програми, курс та форма навчання здобувача освіти, що дозволяє організовувати студентів за групами та освітніми програмами.
Zdobuvachi_oshvity (Студенти)	Зберігає інформацію про студентів, включаючи номер студентського квитка, назву групи, ПІБ та email. Це дозволяє відстежувати, які сертифікати мають конкретні студенти.
Sertyfikaty (Сертифікати)	Містить дані про сертифікати, включаючи номер сертифікату, номер студентського квитка, ПІБ здобувача освіти, назву платформи та назву курсу MOOC, дату, кількість годин, файл та URL, що дозволяє відстежувати інформацію про підтвержені здобутки шляхом неформальної освіти.
Zarahuvannya_neform_oshvity (Зарахування неформальної освіти)	Включає інформацію про зарахування неформальної освіти, такі як ID, назва дисципліни, номер протоколу, дата протоколу та оцінка, що дозволяє відстежувати зарахування досягнень неформальної освіти.

Проаналізуємо структуру кожної сутності детальніше, для визначення атрибутів та зв'язків.

Таблиця NNI (табл. 2.2) призначена для зберігання інформації про навчально-наукові інститути (ННІ) НУВГП, які є структурними підрозділами навчального закладу. Вона містить базову інформацію про ці інститути.

Таблиця 2.2

Атрибути та зв'язки сутності «NNI»

Атрибути	Зв'язки
<b>ID</b>	Зв'язок із таблицею
<b>Nazva NNI:</b> Назва навчально-наукового інституту – <b>Primary Key</b>	<b>Osvitni_programy</b> за допомогою поля "Назва ННІ" (дозволяє вказати, який інститут відповідає за ту чи іншу освітню програму).
<b>NNI:</b> аббревіатура навчально-наукових інститутів.	

Таблиця **Osvitni\_programy** (табл. 2.3) призначена для зберігання інформації про освітні програми, за якими здійснюється здобуття вищої освіти

в залежності від освітнього рівня. Вона містить основні дані про кожну програму, що дозволяє організувати і структурувати освітній процес.

Таблиця 2.3

## Атрибути та зв'язки сутності «Osvitni\_programy»

Атрибути	Зв'язки
<b>ID</b>	Зв'язок із таблицею <b>Dyscypliny</b> через атрибут "Nazva osv programy", що допомагає зрозуміти, які дисципліни входять до складу певної освітньої програми.
<b>Nazva osv programy</b> : Назва освітньої програми – <b>Primary Key</b>	Зв'язок із таблицею <b>Grupy</b> через поле "Nazva osv programy", що дозволяє відслідковувати, які групи належать до певної освітньої програми.
<b>Nazva NNI</b> : Назва навчально-наукового інституту (ННІ)	Зв'язок із таблицею <b>NNI</b> через поле "Nazva NNI", що дозволяє ідентифікувати інститут, відповідальний за освітню програму.
<b>Osv riven</b> : Освітній рівень програми (наприклад, бакалавр, магістр)	

Ця таблиця дозволяє структурувати інформацію про освітні програми, відповідні дисципліни, навчальні групи та інститути. Це допоможе адміністрації та викладачам контролювати навчальний процес, відстежувати виконання академічних вимог та забезпечувати якість освіти.

Таблиця **Dyscypliny** (табл. 2.4). Вона призначена для зберігання інформації про навчальні дисципліни, що викладаються в рамках освітніх програм. Вона містить основні дані про кожну дисципліну, що дозволяє організувати та керувати навчальним процесом.

Таблиця 2.4

## Атрибути та зв'язки сутності «Dyscypliny»

Атрибути	Зв'язки
<b>ID</b>	Зв'язок із таблицею <b>Zarahuvannia_neform_osvity</b> через поле "Nazva dyscypliny", що дозволяє відстежувати, які дисципліни були зараховані на основі неформальної освіти.
<b>Nazva dyscypliny</b> : Назва дисципліни – <b>Primary Key</b> .	
<b>Nazva osv programy</b> : Назва освітньої	Зв'язок із таблицею

програми, до якої належить дисципліна. <b>k-t kredytiv ECTS:</b> Кількість кредитів ECTS (Європейської кредитної трансферної та накопичувальної системи), що присвоюється за цю дисципліну. <b>Forma kontroly:</b> Форма контролю (залік або екзамен)	<b>Osvitni_programy</b> через поле "Nazva osv programy" , що дозволяє визначити, яка освітня програма включає конкретну дисципліну.
---	---

Таблиця Dyscypliny використовується для управління інформацією про навчальні дисципліни. Це включає планування навчального процесу, розподіл навчальних годин та кредитів ECTS, а також взаємодію з освітніми програмами та результатами неформальної освіти.

Таблиця **Grupy** (табл. 2.5) призначена для зберігання інформації про навчальні групи, що формуються в рамках освітніх програм. Вона містить основні дані про кожну групу, що дозволяє організовувати та керувати навчальним процесом для студентів.

Таблиця 2.5

## Атрибути та зв'язки сутності «Grupy»

Атрибути	Зв'язки
<b>ID</b>	Зв'язок із таблицею <b>Zdobuvachi_osvity</b> через поле "Nazva grupy", що дозволяє відстежувати склад студентів у кожній групі.
<b>Nazva grupy:</b> Назва групи – <b>Primary Key</b> .	
<b>Nazva osv programy:</b> Назва освітньої програми, до якої належить група. <b>Kurs:</b> Курс, на якому навчається група (наприклад, 1-й курс, 2-й курс і т.д.). <b>Forma navchannya:</b> Форма навчання (наприклад, денна, заочна).	Зв'язок із таблицею <b>Osvitni_programy</b> через поле "Nazva osv programy", що допомагає зрозуміти, яка освітня програма відповідає за конкретну групу.

Таблиця **Zdobuvachi\_osvity** (табл. 2.6) призначена для зберігання інформації про студентів (здобувачів освіти), які навчаються в навчальному закладі. Вона містить основні дані про кожного студента, що дозволяє організовувати та керувати їхнім навчальним процесом щодо неформальної освіти.

Таблиця 2.6

Атрибути та зв'язки сутності «Zdobuvachi\_osvity»

Атрибути	Зв'язки
<b>Nomer stud kvytka:</b> Унікальний номер студентського квитка – <b>Primary Key</b> .	Зв'язок із таблицею <b>Sertyfikaty</b> через поле "Nomer stud kvytka" , що дозволяє відстежувати подані кожним здобувачем освіти сертифікати для підтвердження досягнень шляхом неформальної освіти.
<b>Nazva grupy:</b> Назва групи, до якої належить студент.	Зв'язок із таблицею <b>Grupy</b> через поле "Nazva grupy" , що допомагає ідентифікувати студентів.
<b>PIB:</b> Прізвище, ім'я, по батькові студента.	
<b>e-mail:</b> Електронна пошта студента.	

Таблиця допомагає у веденні унікального обліку всіх студентів за допомогою номера студентського квитка. Зв'язок з таблицею груп дозволяє легко організовувати та керувати студентами за групами, спрощуючи адміністративні процеси щодо неформальної освіти. Завдяки зв'язку з таблицею сертифікатів можна легко відслідковувати академічні досягнення студентів.

Таблиця **Sertyfikaty** (табл. 2.7) призначена для зберігання інформації про сертифікати, які отримали студенти в рамках навчального процесу шляхом неформальної освіти. Вона містить основні дані про кожен сертифікат, що дозволяє відстежувати відповідні академічні досягнення студентів.

Таблиця 2.7

Атрибути та зв'язки сутності «Sertyfikatu»

Атрибути	Зв'язки
<b>Nomer serytyfikatu:</b> Унікальний номер сертифіката – <b>Primary Key</b> .	Зв'язок із таблицею <b>Zarakhuvannya_neform_osvit</b> у через поле "Nomer serytyfikatu" дозволяє співвідносити сертифікати з відповідними дисциплінами та протоколами.
<b>Nomer stud kvytka:</b> Номер студентського	Зв'язок із таблицею

квитка (зовнішній ключ), який ідентифікує студента, що отримав сертифікат.	<b>Zdobuvachi_osvity</b> через поле "Nomer stud kvytka" дозволяє відстежувати, які сертифікати отримали конкретні студенти, співвіднести сертифікати з відповідними дисциплінами та протоколами.
<b>PIB:</b> Прізвище, ім'я, по батькові студента.	
<b>Nazva osv platformy:</b> Назва освітньої платформи, через яку був отриманий сертифікат.	
<b>Nazva kursu:</b> Назва курсу, за який виданий сертифікат.	
<b>Data:</b> Дата видачі сертифіката.	
<b>K-t godyn:</b> Кількість годин навчання, проведених на курсі.	
<b>File:</b> Файл сертифіката (формат PNG).	
<b>URL:</b> URL-адреса, що веде до сертифіката або пов'язаної інформації.	

Ця таблиця використовується для управління інформацією про сертифікати, видані студентам. Це включає відстеження видачі сертифікатів, перевірку академічних досягнень, управління навчальними даними та створення звітів.

Таблиця **Zarahuvannia\_neform\_osvity** (табл. 2.8) призначена для зберігання інформації про зарахування результатів неформальної освіти студентів. Вона містить дані про оцінювання, дисципліни та інші аспекти неформальної освіти, що дозволяє організувати та керувати процесом зарахування таких результатів.

Таблиця 2.8

Атрибути та зв'язки сутності «Zarahuvannia\_neform\_osvity»

Атрибути	Зв'язки
<b>ID.</b>	Зв'язок із таблицею <b>Sertyfikaty</b> через поле "Nomer serytyfikatu", що дозволяє визначити, який сертифікат підтверджує проходження курсу неформальної освіти.
<b>Nomer serytyfikatu:</b> Номер сертифіката (зовнішній ключ), який підтверджує проходження курсу – <b>Primary Key.</b>	
<b>Nazva dyscypliny:</b> Назва дисципліни, яка була зарахована.	Зв'язок із таблицею <b>Dyscypliny</b> через поле "Nazva dyscypliny", що дозволяє відстежувати, які дисципліни були зараховані на основі здобутого сертифікату неформальної освіти.
<b>Nomer protokolu:</b> Номер протоколу зарахування.	
<b>Data protokolu:</b> Дата протоколу зарахування.	
<b>OCinka:</b> Оцінка, отримана за результатами неформальної освіти.	

Таблиця допомагає забезпечити документальне підтвердження зарахування та оцінювання неформальної освіти, що може бути важливо для академічного обліку та акредитації.

Після визначення зв'язків між сутностями, можемо ідентифікувати їхні типи:

○ **Дисципліни** < — >> **Освітні програми:** Один-до-багатьох (One-to-Many)

- Одна освітня програма може мати багато дисциплін, але кожна дисципліна належить тільки одній освітній програмі.

○ **Групи** < — >> **Освітні програми** : Один-до-багатьох (One-to-Many)

- Одна освітня програма може мати багато груп, але кожна група належить тільки одній освітній програмі.

○ **Освітні програми** < — >> **ННІ:** Один-до-багатьох (One-to-Many)

- Один інститут може мати багато освітніх програм, але кожна освітня програма належить тільки одному інституту.

○ **Сертифікати** < — >> **Здобувачі освіти:** Один-до-багатьох (One-to-Many)

- Один студент може мати багато сертифікатів, але кожен сертифікат належить тільки одному студенту.

○ **Зарахування неформальної освіти < — > Сертифікати:** Один-до-одного (One-to-One)

- Кожне зарахування неформальної освіти відповідає одному сертифікату.

○ **Зарахування неформальної освіти < — >> Дисципліни:** Один-до-багатьох (One-to-Many)

- Одна дисципліна може бути зарахована за багатьма сертифікатами, але кожне зарахування стосується однієї дисципліни.

○ **Здобувачі освіти < — >> Групи:** Один-до-багатьох (One-to-Many)

- Одна група може мати багато студентів, але кожен студент належить тільки одній групі.

В цілому, розроблена концептуальна модель надає основу для подальшого розроблення бази даних, яка дозволить збирати, зберігати та аналізувати дані про неформальну освіту в університеті, сприяючи покращенню управління цим процесом та підвищенню якості навчання.

## 2.2 ER-модель та нормалізація відношень

**Модель «сутність-зв'язок» (ER-модель)** — модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків. ER-модель — це мета-модель даних, тобто засіб опису моделей даних. Існує ряд моделей для представлення знань, але одним з найзручніших інструментів уніфікованого представлення даних, незалежного від програмного забезпечення, що його реалізує, є модель «сутність-зв'язок». Важливим є той факт, що з моделі «сутність-зв'язок» можуть бути породжені всі існуючі моделі даних (ієрархічна, мережева, реляційна, об'єктна), тому вона є найзагальнішою.

Модель сутність-зв'язок є результатом систематичного процесу, який описує та визначає деяку предметну область. Вона не визначає сам процес, а лише візуалізує його. Дані представлені у вигляді компонентів (сутностей), які пов'язані між собою певними зв'язками, які виражають залежності і вимоги між ними, такі як: *один студент може мати багато сертифікатів, проте кожен сертифікат належить тільки одному студенту*. Сутності можуть мати різні властивості (атрибути), які характеризують їх. **Діаграми**, створені для представлення цих сутностей, атрибутів і зв'язків графічно, називають сутність-зв'язок діаграмами.

ER-модель зазвичай реалізується в вигляді схем баз даних. У разі реляційної бази даних, в якій зберігаються дані в таблицях, кожен рядок кожної таблиці являє собою один екземпляр сутності. Деякі поля даних в цих таблицях вказують на індекси в інших таблицях. Такі поля є показниками фізичної реалізації зв'язків між сутностями [20].

**Нормальна форма** — властивість відношення в реляційній моделі даних, що характеризує його з погляду надмірності, що потенційно призводить до логічно помилкових результатів вибірки або зміни даних. Нормальна форма окреслюється сукупністю вимог, яким має задовольняти відношення (таблиць у базі даних).

Процес перетворення відношень бази даних до виду, що відповідає нормальним формам, називається нормалізацією. Нормалізація призначена для приведення структури БД до виду, що забезпечує мінімальну логічну надмірність, і не має на меті зменшення або збільшення продуктивності роботи або зменшення чи збільшення фізичного обсягу бази даних.

Кінцевою метою нормалізації є зменшення потенційної суперечливості, що зберігається в базі даних інформації. Загальне призначення процесу нормалізації полягає в наступному:

- виключення деяких типів надмірності;
- усунення деяких аномалій оновлення;

- розробка проекту бази даних, який є досить «якісним» поданням реального світу, інтуїтивно зрозумілий і може бути гарною основою для подальшого розширення;
- спрощення процедури застосування необхідних обмежень цілісності.

Існують такі рівні нормалізації: перша нормальна форма (1НФ), 2НФ, 3НФ, нормальна форма Бойса-Кодда (БКНФ), 4НФ, 5НФ. Але дотепер жодна з реляційних СКБД не надає належної підтримки усім п'яти нормальним формам. Це відбувається через жорсткі вимоги до продуктивності. Суть справи полягає в тому, що в повністю нормалізованій БД для виконання запиту треба з'єднати настільки багато таблиць, що продуктивність такої системи не зможе задовольнити користувачів. Тому на практиці використовують лише перші три рівня нормалізації – 1НФ, 2НФ, 3НФ [22].

*Атомарним відношенням* називається відношення, яке неможливо декомпонувати на незалежні проєкції. Далеко не завжди для неатомарних відношень потрібна декомпозиція на атомарні проєкції. При виборі способу декомпозиції необхідно прагнути до одержання незалежних проєкцій, але не обов'язково атомарних.

**Перша нормальна форма (1НФ)** — базова нормальна форма відношення у реляційній моделі даних.

*Змінна відношення знаходиться в першій нормальній формі тоді і тільки тоді, коли в будь-якому допустимому значенні цієї змінної кожен кортеж відношення містить лише одне значення для кожного з атрибутів.*

У реляційній моделі відношення завжди знаходиться в першій нормальній формі за визначенням поняття відношення.

Змінна відношення знаходиться в **другій нормальній формі** тоді і тільки тоді, коли вона знаходиться в першій нормальній формі та кожен неключовий атрибут невідомо залежить від (кожного) її потенційного ключа.

Незвідність означає, що у складі потенційного ключа відсутня менша підмножина атрибутів, від якої можна також вивести цю функціональну

залежність. Для функціональної залежності, що не наводиться, часто використовується еквівалентне поняття «повна функціональна залежність».

Якщо потенційний ключ є простим, тобто складається з єдиного атрибуту, то будь-яка функціональна залежність від нього є незвідною (повною). Якщо потенційний ключ є зіставним, то, згідно з визначенням другої нормальної форми, у відношенні не повинно бути неключових атрибутів, що залежать від частини зіставного потенційного ключа [20].

Відношення перебуває у **третій нормальній формі** (3NF, 3НФ), якщо воно перебуває в 2NF, і всі неключові атрибути відношення залежать тільки від первинного ключа. Іншими словами, 3NF вимагає, щоб відношення не містило транзитивних функціонального зв'язку неключових атрибутів від ключа.

Таким чином, процедура зведення відношення до 3NF складається у виконанні двох проєкцій: по правій і по лівій частині транзитивного функціонального зв'язку.

Зрозуміло, що в процесі нормалізації декомпозиція відношення на незалежні проєкції є кращою. Необхідні й достатні умови незалежності проєкцій відношення забезпечує теорема Ріссанена: проєкції  $r_1$  і  $r_2$  відношення  $r$  є незалежними тоді й тільки тоді, коли кожний зв'язок у відношенні  $r$  логічно виходить зі зв'язку у  $r_1$  і  $r_2$ ; загальні атрибути  $r_1$  і  $r_2$  утворюють можливий ключ хоча б для одного з цих відношень.

Змінна відношення перебуває в **нормальній формі Бойса-Кодда** (BCNF) у тому і тільки в тому випадку, коли будь-який виконуваний для цього змінного відношення нетривіальний і мінімальний функціональний зв'язок має як детермінант деякий можливий ключ даного відношення [23].

Перейдемо до аналізу схеми бази даних обліку неформальної освіти НУВГП (рис. 2.3), щодо відповідності вимогам умов нормалізації:

#### 1. ННІ:

- 1НФ: Всі атрибути є атомарними.
- 2НФ: Кожен неключовий атрибут повністю залежить від ключа (Назва ННІ).



- 3НФ: Відсутні транзитивні залежності між неключовими атрибутами.

#### 5. Здобувачі освіти:

- 1НФ: Всі атрибути є атомарними.
- 2НФ: Кожен неключовий атрибут повністю залежить від ключа (Номер студентського квитка).
- 3НФ: Відсутні транзитивні залежності між неключовими атрибутами.

#### 6. Сертифікати:

- 1НФ: Всі атрибути є атомарними.
- 2НФ: Кожен неключовий атрибут повністю залежить від ключа (Номер сертифікату).
- 3НФ: Прізвище студента можна виділити в окрему таблицю "Здобувачі освіти".

#### 7. Зарахування неформальної освіти:

- 1НФ: Всі атрибути є атомарними.
- 2НФ: Кожен неключовий атрибут повністю залежить від ключа (Номер сертифікату).
- 3НФ: Відсутні транзитивні залежності між неключовими атрибутами.

Отже, всі сутності бази даних "Неформальна освіта" відповідають 3НФ, що забезпечує відсутність аномалій вставки, видалення та оновлення, а також зменшення надмірності даних.

## РОЗДІЛ 3. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

### 3.1 Backend

Бекенд є ключовим компонентом програмної системи, відповідальним за обробку запитів, управління даними і забезпечення логіки застосунку. Він працює на серверній стороні і здійснює взаємодію між фронтендом та базою даних, гарантуючи надійність, масштабованість та безпеку системи. Основними завданнями бекенду є обробка бізнес-логіки, управління ресурсами та надання API для взаємодії з фронтендом та іншими сервісами.

Сутності з атрибутами були описані мовою C# (вибір якої було обгрунтовано у розділі 1), безпосередньо у програмному коді інформаційної системи, що наведений у Додатку А.

Перейдемо до аналізу запитів на основі таблиць, що були розглянуті раніше.

Основний простий запит «Зарахування неформальної освіти» дозволяє отримати зведену інформацію, яка містить відомості про здобувачів освіти (ННІ, Назва групи, Освітній рівень, ПІБ), отримані сертифікати (Назва освітньої платформи, Назва курсу) та результати зарахування цих сертифікатів у навчальному процесі - Назва дисципліни, Оцінка, № протоколу, Дата протоколу (рис. 3.1).



Простий запит «Зарахування неформальної освіти»

```
SELECT ННІ.НІ, Групи.Назва групи, Освітні програми.Освітній рівень, Здобувачі освіти.ПІБ, Сертифікати.Назва освітньої платформи, Сертифікати.Назва курсу, Зарахування неформальної освіти.Назва дисципліни, Зарахування неформальної освіти.Оцінка, Зарахування неформальної освіти.№ протоколу, Зарахування неформальної освіти.Дата протоколу FROM (SELECT ІМЕР JOIN Освітні програми ON ННІ.Назва ННІ = Освітні програми.Назва ННІ ІМЕР JOIN Групи ON Освітні програми.Назва освітньої програми = Групи.Назва освітньої програми ІМЕР JOIN Здобувачі освіти ON Групи.Назва групи = Здобувачі освіти.Назва групи ІМЕР JOIN Сертифікати ON Здобувачі освіти.ПІБ студентського квитка = Сертифікати.ПІБ студентського квитка ІМЕР JOIN Зарахування неформальної освіти ON Сертифікати.№ сертифікату = Зарахування неформальної освіти.№ сертифікату)
```

Рис. 3.1 Простий запит «Зарахування неформальної освіти»



Запит «Статистика дисципліни» здійснює підрахунок кількості записів для кожної дисципліни і повертає цю інформацію у вигляді таблиці з двома стовпцями: Назва дисципліни та Кількість (рис. 3.5).

```
Статистика Дисципліни  
SELECT (Зарядування неформальної освіти)[Назва дисципліни], Count([Зарядування неформальної освіти][Назва дисципліни]) AS Кількість  
FROM (Зарядування неформальної освіти)  
GROUP BY (Зарядування неформальної освіти)[Назва дисципліни];
```

Рис. 3.5 Запит «Статистика дисципліни»

Результат запиту «Статистика дисципліни», який містить інформацію щодо обліку кількості сертифікатів, які підтверджують здобутки неформальної освіти в розрізі дисциплін наведено на рис. 3.6.

Назва дисципліни	Кількість
Економіко-математичні методи і моделі	3
Інтелектуальний аналіз даних	4
Нереляційні бази даних	1
Основи цифрових технологій	5
Системи підтримки прийняття рішень	1
Теорія прийняття рішень	1

Рис. 3.6 Результат запити «Статистика дисципліни»

Запит «Статистика ННІ» здійснює підрахунок кількості записів для кожного ННІ і повертає цю інформацію у вигляді таблиці з двома стовпцями: ННІ та Кількість (рис. 3.7). Тобто даний запит дозволяє проаналізувати активність здобувачі у здобутті професійних компетентностей шляхом неформальної освіти в розрізі ННІ.

```
Статистика ННІ  
SELECT (Простий запит_Зарядування неформальної освіти)[ННІ], Count([Простий запит_Зарядування неформальної освіти][ННІ]) AS Кількість  
FROM (Простий запит_Зарядування неформальної освіти)  
GROUP BY (Простий запит_Зарядування неформальної освіти)[ННІ];
```

Рис. 3.7 Запит «Статистика ННІ»

Результат запиту «Статистика ННІ» наведено на рис. 3.8.



ННІ	Кількість
ННІЕАВГ	2
ННІЕМ	8
ННІКІТІ	4

Рис. 3.8 Результат запиту «Статистика ННІ»

Запит «Статистика Освітні платформи» здійснює підрахунок кількості записів для кожної освітньої платформи і повертає цю інформацію у вигляді таблиці з двома стовпцями: Назва освітньої платформи та Кількість (рис. 3.9). Тобто даний запит дозволяє проаналізувати популярність освітніх платформ для здобуття професійних компетентностей шляхом неформальної освіти.



```
SELECT Сертифікати.[Назва освітньої платформи], Count(Сертифікати.[Назва освітньої платформи]) AS Кількість
FROM Сертифікати
GROUP BY Сертифікати.[Назва освітньої платформи];
```

Рис. 3.9 Запит «Статистика Освітні платформи»

Результат запиту «Статистика Освітні платформи», в якому відображено зведені загальні кількості сертифікатів МООС в розрізі освітніх платформ наведено на рис. 3.10.



Назва освітньої платформи	Кількість
Courseya	8
Prometheus	2
Дія.Цифрова освіта	5

Рис. 3.10 Результат запиту «Статистика Освітні платформи»

Перехресний запит «Зарахування неформальної освіт \_ Освітні платформи» дозволяє отримати узагальнену інформацію щодо кількостей зарахованих сертифікатів, які підтверджують набуті компетентності здобувачів освіти шляхом неформальної освіти в залежності від обраних ними освітніх платформ та навчальних дисциплін за якими ці сертифікати є зарахованими (рис. 3.11).

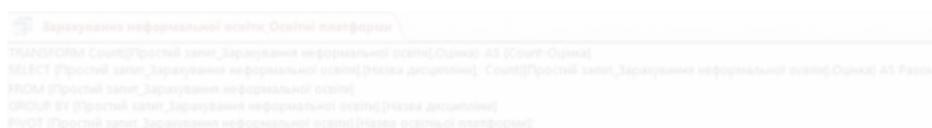


Рис. 3.11 Перехресний запит «Зарахування неформальної освіт \_ Освітні платформи»

Результати перехресного запиту «Зарахування неформальної освіт», в якому відображено результати загальних кількостей сертифікатів МООС в розрізі освітніх платформ наведено на рис. 3.12

Назва дисципліни	Разом	Coursera	Promethea	Дія. Цифрова освіта
Економіко-математичні методи і моделі	3	1	2	
Інтелектуальний аналіз даних	4	4		
Нереляційні бази даних	1	1		
Основи цифрових технологій	5			5
Системи підтримки прийняття рішень	1	1		
Теорія прийняття рішень	1	1		

Рис. 3.12 Результат перехресного запиту «Зарахування неформальної освіт»

Перехресний запит «Зарахування неформальної освіти \_ Освітній рівень» дозволяє отримати узагальнену інформацію щодо кількостей зарахованих сертифікатів, які підтверджують набуті компетентності здобувачів освіти шляхом неформальної освіти НУВГП в залежності від освітнього рівня та навчальних дисциплін за якими ці сертифікати є зарахованими (рис. 3.12).



Рис. 3.12 Перехресний запит «Зарахування неформальної освіти \_ Освітній рівень»

Результати перехресного запиту «Зарахування неформальної освіти \_ Освітній рівень», в якому відображено результати загальних кількостей сертифікатів МООС за відповідними дисциплінами в розрізі освітніх рівнів наведено на рис. 3.13

Рис. 3.13 Результат перехресного запиту «Зарахування неформальної освіти \_ Освітній рівень»

Таким чином, бекенд є невід'ємною частиною будь-якої програмної системи, забезпечуючи її функціональність, надійність та безпеку. Чіткий опис допомагає в проведенні тестування та валідації системи. Він дозволяє зрозуміти поточний стан системи та визначити, які зміни необхідні для досягнення нових функціональних або нефункціональних вимог. Також він допомагає визначити можливі точки покращення продуктивності та масштабованості системи. Це включає оптимізацію запитів до бази даних, поліпшення обробки запитів та збільшення ефективності використання ресурсів сервера.

### 3.2 Frontend

Ми використали елементи стилістики брендбуку Національного університету водного господарства та природокористування у нашій програмі для обліку неформальної освіти [24]. Використання сучасного та лаконічного дизайну, що включає стримані кольори, чіткі лінії та офіційну символіку університету, дозволяє забезпечити узгодженість і візуальну привабливість інтерфейсу користувача. Це сприяє створенню професійного враження від програмного забезпечення та підкреслює його зв'язок з університетом, зміцнюючи бренд та покращуючи досвід користувачів.

Запустивши інформаційну систему обліку неформальної освіти НУВГП, користувач побачить вікно входу в застосунок (рис. 3.14) та має пройти процедуру ідентифікації та аутентифікації.



Рис. 3.14 Вікно входу у ІС обліку неформальної освіти НУВГП

Кнопка «Очистити» призначена для очищення полів «Ім'я» та «Пароль», «Вихід» - для виходу до застосунку.

При успішному введенні логіну та пароля, користувачеві буде відкрите головне меню інформаційної системи (рис. 3.15).

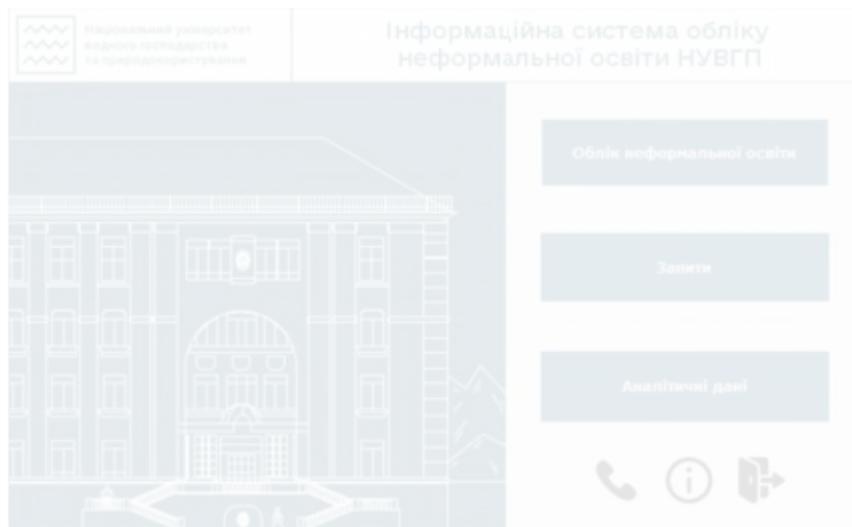


Рис. 3.15 Головне меню ІС обліку неформальної освіти НУВГП

Головне меню інформаційної системи включає такі кнопки:

- **«Облік неформальної освіти»;**  
Ця кнопка призначена для перегляду та внесення інформації у базу даних. Користувачі можуть використовувати її для додавання нових записів про неформальну освіту, оновлення наявних даних або перегляду існуючої інформації.
- **«Запити»;**  
Ця кнопка надає доступ до перегляду статистики інформаційної системи. Користувачі можуть виконувати різноманітні запити для отримання статистичних даних, що стосуються обліку неформальної освіти.
- **«Аналітичні дані»;**  
Ця кнопка призначена для аналізу даних. Користувачі можуть отримувати доступ до аналітичних інструментів, що дозволяють

проводити детальний аналіз зібраних даних, виявляти тенденції, готувати звіти та робити висновки на основі даних.

- **«Телефон» (інформація про розробника, рис. 3.16);**  
Кнопка із зображенням телефону призначена для отримання інформації про розробника застосунку. Користувач зможе дізнатись інформацію про компанію, команду або особу, яка розробила інформаційну систему.
- **«Інформація» (довідник застосунку, рис. 3.17);**  
Кнопка із зображенням інформаційного знаку відкриває довідник застосунку. Це розділ з інструкцією та допомогою для користувачів, де можна знайти відповіді на запитання, інструкції щодо користування системою.
- **Вихід (вихід з програми).**  
Кнопка із зображенням виходу призначена для виходу з програми. Користувачі можуть завершити свою роботу з інформаційною системою, безпечно заклавши програму.



Рис. 3.16 Вікно «Про розробника»

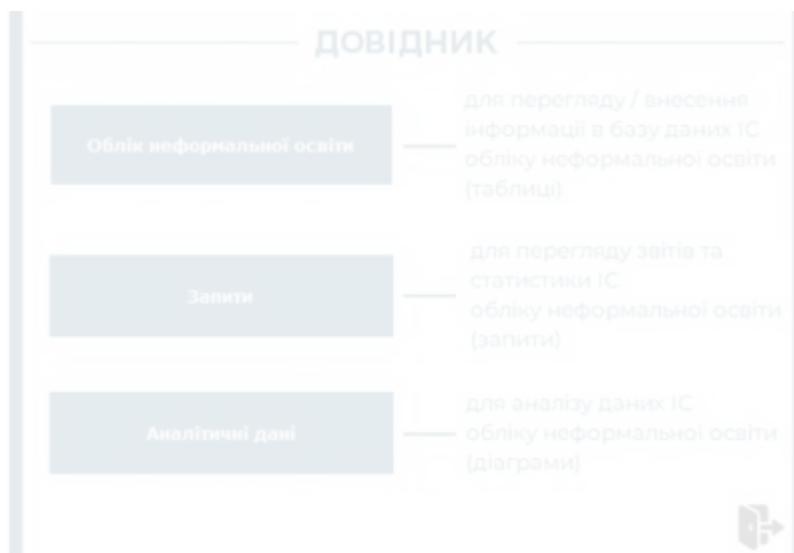


Рис. 1.17 Вікно «Інформація» (довідник ІС)

Як було вказано раніше, для перегляду, внесення чи редагування даних в інформаційній системі обліку неформальної освіти НУВГП потрібно перейти по кнопці «Облік неформальної освіти». У результаті, користувачеві відкриється наступне вікно з переліком сутностей бази даних (рис. 3.18).

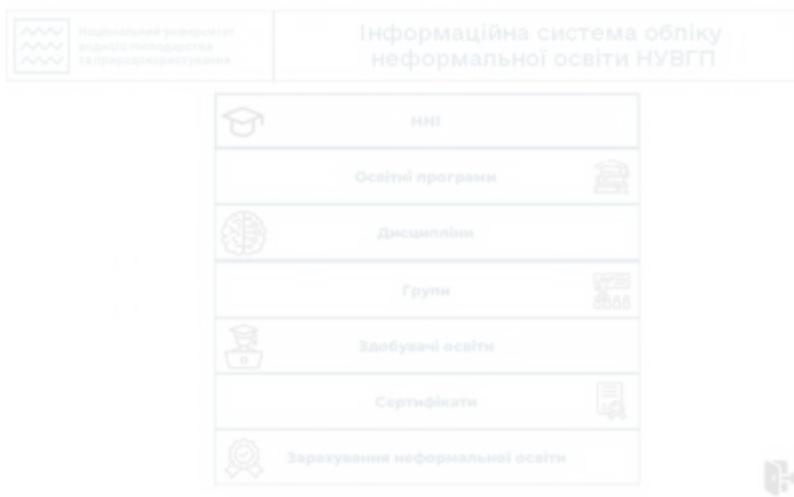


Рис. 3.18 Вікно «Облік неформальної освіти»

При роботі з інформаційною системою порядок внесення даних до неї визначається рівнем підлеглості сутностей бази даних (розділ 2).

Натискаючи на кнопку «Дисципліни», користувач переходить на однойменну форму, що зображена на рисунку 3.19.

ID	Назва дисципліни	Назва освітньої програми	К-ть кредитів	Форма контролю
1	Економіко-математичні методи і моделі	Інформаційні системи	4	екзам
2	Методи глибокої аналізи даних	ІІС Інформаційні системи і технології	5	екзам
3	Методи бази даних	ІІС Інформаційні системи і технології (англ.)	3	екзам
4	Основи сучасних технологій	ІІС ІТ-менеджмент	3	екзам
5	Системи підтримки прийняття рішень	ІІС Бізнесова інформатика	5	екзам
6	Теорія прийняття рішень	ІІС ІТ-менеджмент (англ.)	4	екзам

Рис. 3.19 Вікно «Дисципліни»

Це вікно забезпечує зручний інтерфейс для управління інформацією про дисципліни, дозволяючи додавати, редагувати, видаляти та переглядати записи про дисципліни, що полегшує адміністративні процеси в освітній установі.

Назва дисципліни – це поле, де користувач може ввести або вибрати назву дисципліни.

Назва освітньої програми - поле з спадним списком, що дозволяє вибрати назву освітньої програми, до якої відноситься дисципліна.

Кількість кредитів - поле для введення кількості кредитів, які нараховуються за цю дисципліну.

Форма контролю - спадний список, де користувач може вибрати форму контролю (наприклад, залік, екзамен).

Кнопка "Добавити" - кнопка для додавання нової дисципліни до списку. Після заповнення всіх полів користувач натискає цю кнопку, щоб зберегти інформацію про нову дисципліну.

Кнопка "Редагувати" - для редагування вибраної дисципліни. Користувач може вибрати дисципліну зі списку нижче, змінити потрібну інформацію та зберегти зміни.

Кнопка "Очистити поля" - для очищення всіх полів введення. Це дозволяє користувачеві швидко очистити форму для введення нових даних.

Кнопка "Видалити" - для видалення вибраної дисципліни зі списку. Користувач може вибрати дисципліну і натиснути цю кнопку для її видалення з бази даних.

Таблиця дисциплін відображає список усіх дисциплін з такими колонками:

- ID: Ідентифікатор дисципліни.
- Назва дисципліни.
- Назва освітньої програми: Освітня програма, до якої належить дисципліна.
- К-ть кредитів: Кількість кредитів за дисципліну.
- Форма контролю (залік, екзамен).

Кнопка "Інформація" (рис. 3.20) - відкриває довідник застосунку, де користувач може знайти інструкції та допомогу з користування системою.

Кнопка "Вихід" – призначена для виходу з програми, дозволяє користувачеві завершити роботу з системою.

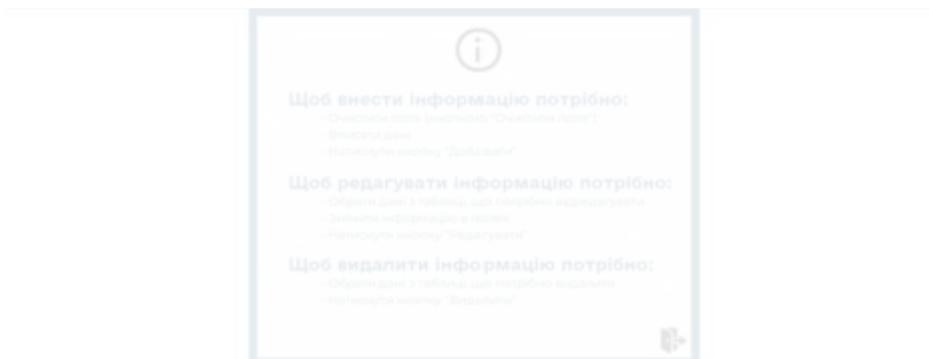


Рис. 3.20 Довідка для користування обліком неформальної освіти  
При виборі кнопки «Групи» відкриється наступне вікно (рис. 3.21).

ID	Назва групи	Назва освітньої програми	Курс	Форма навчання
1	EK-41	051 Економічна кібернетика	4	денна
2	КТ-25тг	126 Інформаційні системи і технології	4	денна
3	КТ-41	126 Інформаційні системи і технології	4	денна
4	ПБ-5тг	126 Інформаційні системи і технології (магістр)	5	денна
5	МБ-11	133 Галузеве машинобудування	1	денна
6	ОО-11	071 Облік і оподаткування	1	денна
7	ОО-31	071 Облік і оподаткування	3	денна
8	ПТБД-11	076 Підприємство, торгівля та біржова діяльність	1	денна
9	Фін-11тг	072 Фінанси, банківська справа та страхування	3	денна

3.21 Вікно «Групи»

Це вікно призначене для управління навчальними групами в рамках інформаційної системи обліку неформальної освіти.

Назва групи - поле для введення або вибору назви навчальної групи.

60

Назва освітньої програми – спадний список, який дозволяє вибрати назву освітньої програми, до якої належить група.

Курс - поле для введення або вибору курсу навчання для групи.

Форма навчання - спадний список, де користувач може вибрати форму навчання (наприклад, денна, заочна).

Кнопки "Добавити", "Редагувати", "Очистити поля", "Видалити", "Інформація" та "Вихід" мають такий самий функціонал, як і у вікні "Дисципліни", та будуть повторюватись у наступних полях для полегшення користування.

Якщо користувач вибере кнопку «ННІ», користувач перейде на вікно інформації про навчально-наукові інститути НУВГП (рис. 3.22).

ID	Назва навчально-наукового інституту	ННІ
1	НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АГРОЕКОЛОГІЇ ТА ЗЕМЛЕУС	ННАЗ
2	НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БУДІВНИЦТВА ТА АРХІТЕКТУРИ	ННБА
3	НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ ТА МЕНЕДЖМЕНТУ	ННЕМ
4	НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕНЕРГЕТИКИ, АВТОМАТИКИ ТА ВОДНОГО ГОСПОДАРСТВА	ННЕАВГ
5	НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРНЕТИКИ, ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ІНЖЕНЕРІ	ННІКІТІ
6	НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ОХОРОНИ ЗДОРОВ'Я	ННОЗ
7	НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ПРАВА	ННП
8	НАВЧАЛЬНО-НАУКОВИЙ МЕХАНІЧНИЙ ІНСТИТУТ	ННМ

Рис. 3.22 Вікно «ННІ»

Це вікно призначене для управління навчально-науковими інститутами в рамках інформаційної системи.

61

Назва навчально-наукового інституту - поле для введення або вибору назви навчально-наукового інституту.

ННІ - поле для введення скороченої назви навчально-наукового інституту.

Нижче знаходиться таблиця, яка відображає перелік усіх наявних навчально-наукових інститутів, які були додані в систему.

При виборі кнопки «Освітні програми», користувачеві буде доступне вікно зображене на рисунку 3.23.

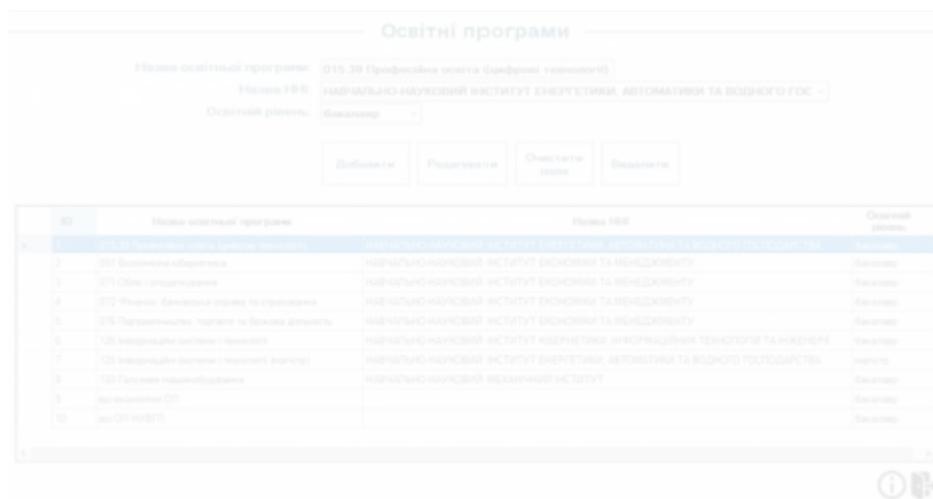


Рис. 3.23 Вікно «Освітні програми»

Дане вікно відображає перелік усіх наявних освітніх програм, які були додані в систему.

Назва освітньої програми - поле для введення або вибору назви освітньої програми.

Назва ННІ - спадний список, який дозволяє вибрати навчально-науковий інститут, до якого належить освітня програма.

Освітній рівень - спадний список, де користувач може вибрати освітній рівень (наприклад, бакалавр, магістр).

При натисканні на кнопці «Сертифікати», користувач побачить наступне вікно з новим інтерфейсом (рис. 3.24).

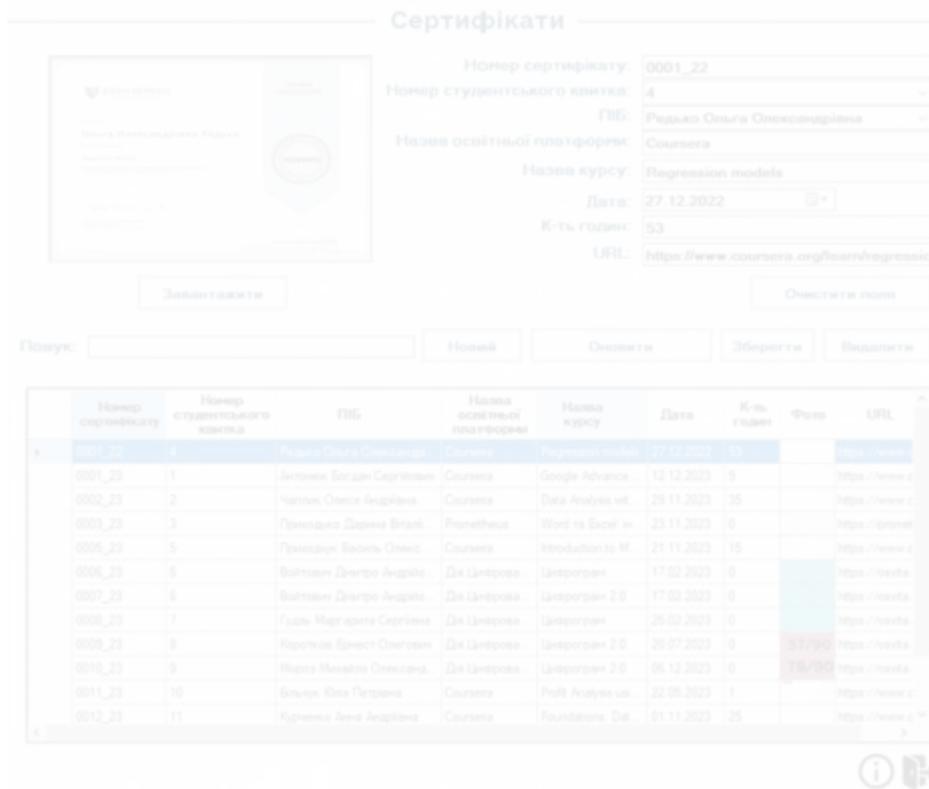


Рис. 3.24 Вікно для обліку сертифікатів

Це вікно забезпечує зручний інтерфейс для управління інформацією про сертифікати, дозволяючи додавати, редагувати, видалити та переглядати записи про сертифікати, що полегшує адміністративні процеси в освітній установі.

Номер сертифікату – поле для введення або вибору номера сертифікату.

Номер студентського квитка – поле для введення номера студентського квитка.

ПІБ – спадний список для вибору прізвища, імені та по батькові студента.

Назва освітньої платформи – поле для введення або вибору назви освітньої платформи, де було отримано сертифікат.

Назва курсу – поле для введення назви курсу, за яким було отримано сертифікат.

Дата – поле для введення дати отримання сертифікату.

К-ть годин – поле для введення кількості годин, витрачених на курс.

URL – поле для введення посилання на сертифікат або курс.

Файл сертифікату – поле для завантаження фотозображення сертифікату.

Кнопки:

- "Завантажити" – дозволяє завантажити зображення сертифікату (формату JPG).
- "Очистити файл" – для очищення поля файлу сертифікату.
- "Очистити поля" – очищує всі поля для введення, готуючи їх до введення нової інформації.
- "Новий" – додає нові дані у інформаційну систему.
- "Оновити" – дозволяє оновити вікно застосунку.
- "Зберегти" – зберігає інформацію про новий або відредагований сертифікат.
- "Видалити" – видаляє вибраний запис про сертифікат.

Нижче знаходиться таблиця, яка містить такі колонки:

- Номер сертифікату – унікальний ідентифікатор сертифікату.
- Номер студентського квитка – номер студентського квитка.
- ПІБ – прізвище, ім'я та по батькові студента.
- Назва освітньої платформи – назва платформи, де було отримано сертифікат.
- Назва курсу – назва курсу.
- Дата – дата отримання сертифікату.
- К-ть годин – кількість годин, витрачених на курс.
- Фото – зображення сертифікату.
- URL – посилання на сертифікат або курс.

Кнопки "Інформація" та "Вихід" повторюють попередній функціонал, проте довідник вікна «Сертифікати» дещо відрізняється, через зміну інтерфейсу (рис. 3.25).

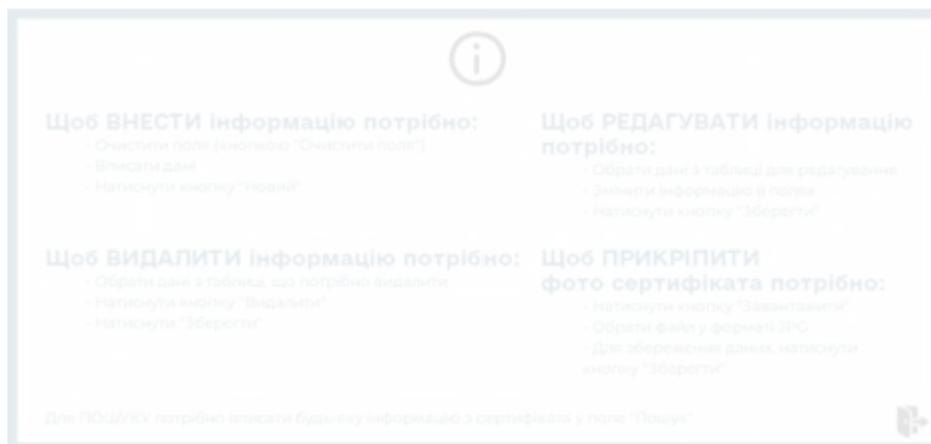


Рис. 3.25 Вікно довідки «Сертифікати»

При натисканні на кнопки «Зарахування неформальної освіти» чи «Здобувачі освіти» відкриваються наступні вікна, що схожі на перші, що ми розбирали (рис. 3.26, 3.27).

Перше вікно дозволяє додавати, редагувати, видаляти та переглядати записи про курси неформальної освіти, включаючи інформацію про сертифікати, дисципліни, протоколи та оцінки. Воно містить записи про зараховану неформальну освіту:

- Номер сертифікату;
- Назва дисципліни;
- Номер протоколу;
- Дата протоколу;
- Оцінка.

Вікно інформаційної системи для обліку здобувачів освіти дозволяє додавати, редагувати, видаляти та переглядати записи про здобувачів освіти,

включаючи інформацію про студентські квитки, групи, повні імена та електронні адреси.

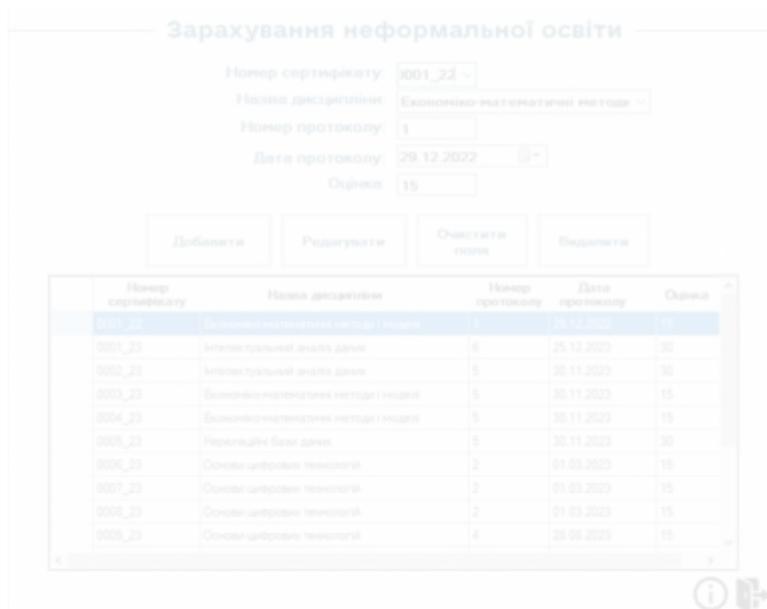


Рис. 3.26 Вікно «Зарахування неформальної освіти»

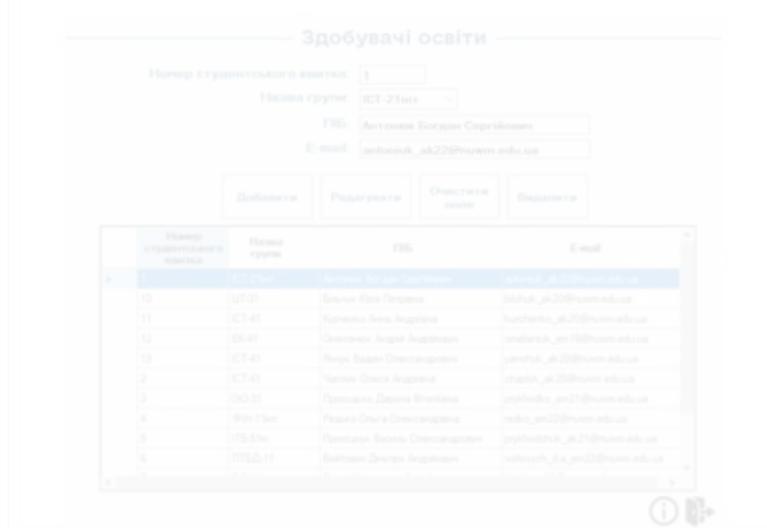


Рис. 3.27 Вікно «Здобувачі освіти»

Узагальнюючи, розроблені діалогові вікна форм інформаційної системи для обліку неформальної освіти НУВГП забезпечують високий рівень зручності та ефективності роботи з системою. Вони дозволяють користувачам швидко та точно додавати, редагувати та видаляти інформацію про сертифікати, а також зберігати фотозображення документів. Це сприяє більш ефективному управлінню даними та підвищенню надійності обліку неформальної освіти в університеті

### 3.3 Запити інформаційної системи

При виборі кнопки «Запити», на головному меню застосунку, з'явиться наступне вікно з вибором перегляду статистик інформаційної системи (рис. 3.28).

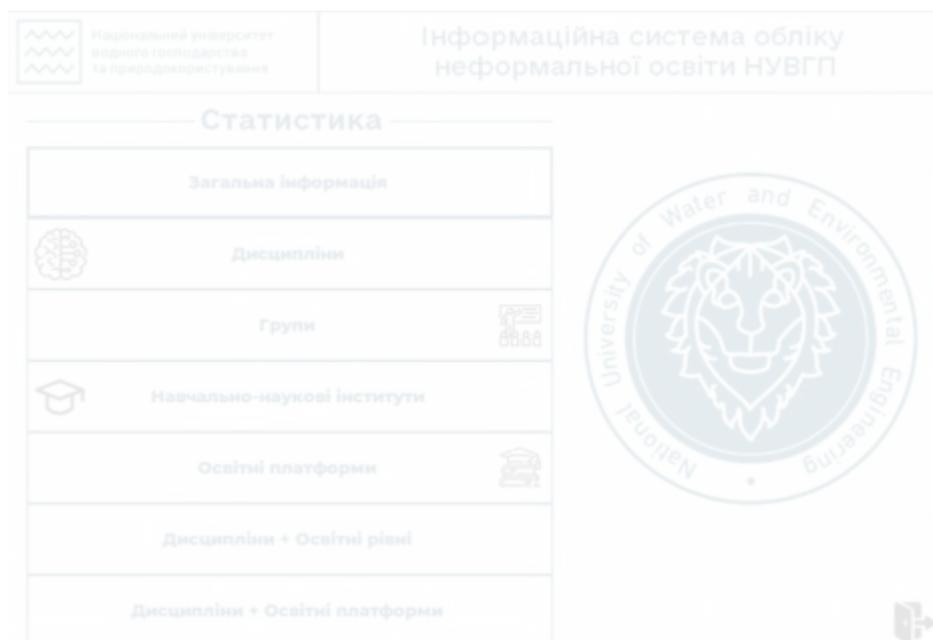


Рис. 3.28 Вікно «Запити»

При натисканні на кнопки «Загальна інформація» (рис. 3.29), з'явиться загальний запит інформації про сертифікати в ІС обліку неформальної освіти Національного університету водного господарства та природокористування (НУВГП).

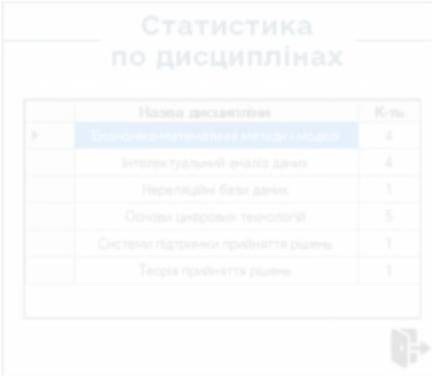
Таблиця містить інформацію про сертифікати неформальної освіти:

- ННІ (Навчально-науковий інститут.).
- Назва групи (Назва академічної групи студента).
- Освітній рівень (Рівень освітньої програми, яку здобуває студент).
- ПІБ (Повне ім'я студента).
- Освітня платформа (Назва платформи, на якій студент проходив курс неформальної освіти).
- Курс (Назва курсу, який пройшов студент на обраній платформі).
- Дисципліна (Назва дисципліни, що відповідає курсу та в якій зараховано результати навчання.).
- Оцінка (Оцінка, яку отримав студент за курс.).
- № протоколу (Номер протоколу, в якому зафіксовано результати зарахування курсу.).
- Дата (Дата протоколу, в який було внесено результати.).

ID	Назва курсу	Назва академічної групи	ННІ	Статус	Назва	Опис	Оцінка	№ протоколу	Дата
100001	100001	100001	100001	100001	100001	100001	100001	100001	10.10.2024
100002	100002	100002	100002	100002	100002	100002	100002	100002	10.10.2024
100003	100003	100003	100003	100003	100003	100003	100003	100003	10.10.2024
100004	100004	100004	100004	100004	100004	100004	100004	100004	10.10.2024
100005	100005	100005	100005	100005	100005	100005	100005	100005	10.10.2024
100006	100006	100006	100006	100006	100006	100006	100006	100006	10.10.2024
100007	100007	100007	100007	100007	100007	100007	100007	100007	10.10.2024
100008	100008	100008	100008	100008	100008	100008	100008	100008	10.10.2024
100009	100009	100009	100009	100009	100009	100009	100009	100009	10.10.2024
100010	100010	100010	100010	100010	100010	100010	100010	100010	10.10.2024

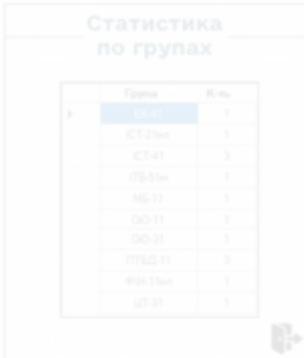
Рис. 3.29 Вікно запити «Загальна інформація»

При виборі перегляду статистики дисциплін, груп, ННІ чи освітніх платформ, відкриються наступні вікна, що зображені на рисунках 3.30, 3.31, 3.32 та 3.33, що призначені тільки для перегляду статистики.



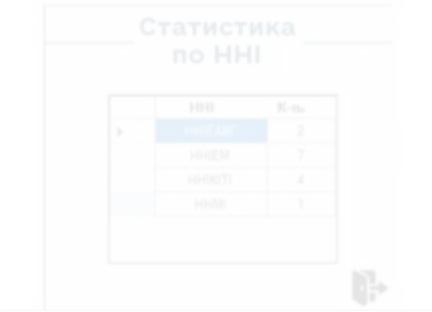
	Назва дисципліни	К-ть
№	Основи математичних систем керування	4
	Інтелектуальний аналіз даних	4
	Нереляційні бази даних	1
	Основи цифрових технологій	5
	Системи підтримки прийняття рішень	1
	Теорія прийняття рішень	1

Рис. 3.30 Вікно статистики по дисциплінах



	Група	К-ть
№	СТ-11	1
	СТ-211	1
	СТ-41	3
	СТ-51	1
	WS-11	1
	DO-11	1
	DO-31	1
	ПІІІ-11	3
	WS-111	1
	СТ-31	1

Рис. 3.31 Вікно статистики по групах



	ННІ	К-ть
№	ННІ-11	2
	ННІ-EM	3
	ННІ-OT	4
	ННІ-0	1

Рис. 3.32 Вікно статистики по навчально-наукових інститутах

Освітня платформа	К-ть
Coursera	3
Proxeheus	1
Дія Цифрова освіта	5

Рис. 3.33 Вікно статистики по освітнім платформам

Запити «Дисципліни + Освітні рівні» та «Дисципліни + Освітні платформи» призначені для порівняння даних (рис. 3.34, рис. 3.35).

Дисципліна	Усього	Бакалаври	Магістри
Інтелектуальний аналіз даних	2	2	
Інтелектуальний аналіз даних	4	4	
Перекладні бази даних	1		1
Основи цифрової технології	5	5	
Система підтримки прийнятті рішень	1	1	
Теорія прийняття рішень	1	1	

Рис. 3.34 Вікно запиту «Дисципліни + Освітні рівні»

Дисципліна	Усього	Course	Platform	Дія. Цілісна освіта
Система управління інформаційними ресурсами	2	1	1	
Інформаційний аналіз даних	4	4		
Нарешадки бази даних	1	1		
Основи цифрових технологій	5			5
Система підтримки прийняття рішень	1	1		
Теорія прийняття рішень	1	1		

Рис. 3.35 Вікно запиту «Дисципліни + Освітні платформи»

У рамках розробки інформаційної системи обліку неформальної освіти планується впровадження вкладки "Аналітичні дані". Ця вкладка буде призначена для візуалізації та аналізу даних за допомогою динамічних діаграм. В даний момент ця вкладка ще знаходиться у стадії розробки (приклад аналітичних даних зображено на рисунку 3.36).



Рис. 3.36 Приклад діаграм ІС обліку неформальної освіти НУВГП

Розроблений Frontend забезпечує зручну та ефективну взаємодію користувачів з інформаційною системою, сприяючи підвищенню продуктивності та точності обліку неформальної освіти. Інтуїтивно зрозумілий інтерфейс дозволяє користувачам швидко освоїти систему та ефективно виконувати необхідні операції. Подальший розвиток вкладки "Аналітичні дані"

забезпечить додаткові можливості для аналізу та візуалізації інформації, що значно підвищить цінність системи для освітніх установ.

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи на здобуття ступеня «бакалавр» створено застосунок обліку неформальної освіти Національного університету водного господарства та природокористування. Він дозволяє зібрати всі дані про неформальну освіту студентів в одному місці, спрощує їх пошук і управління, що значно полегшує роботу викладачів та адміністрації, зменшуючи час на обробку даних і паперову роботу. Викладачі та адміністрація можуть аналізувати дані про неформальну освіту, щоб визначити, які курси є найбільш популярними та корисними; аналізувати здобувачі яких курсів, спеціальностей та рівнів освіти є активними у здобутті неформальної освіти; відслідковувати показники неформальної освіти в динаміці тощо.

Технологічний прогрес освіти вимагає впровадження нових інструментів для управління навчальним процесом. Тому створення такого застосунку відповідає сучасним тенденціям цифровізації та інновацій в освіті. Розроблений продукт допомагає університету виконувати свою місію щодо підготовки висококваліфікованих спеціалістів та сприяє розвитку інноваційних підходів у навчанні.

Було проаналізовано різні моделі життєвого циклу розробки програмного забезпечення та обрано найбільш підходящу для даного проекту - спіральну модель ЖЦ ІС.

Також було проведено глибокий аналіз предметної області, який включав вивчення потреб користувачів, вимог до функціональності системи та особливостей обліку неформальної освіти. Це дозволило визначити ключові функції застосунку, необхідні для ефективного управління даними про неформальну освіту студентів.

Для розробки ІС було обрано мову програмування C# завдяки її потужним можливостям та інтеграції з .NET платформою. Середовище розробки Visual Studio 2019 забезпечує ефективну розробку та відлагодження

коду. Для управління базою даних було обрано Microsoft Access, що пропонує простоту використання та інтеграцію з іншими продуктами Microsoft.

Кожен з цих компонентів був обраний з урахуванням специфічних вимог проекту, що забезпечило високу якість кінцевого продукту.

З метою успішного функціонування бази даних інформаційної системи обліку неформальної освіти моделювання було відібрано 7 сутностей, що відображають основні компоненти та процеси, пов'язані з обліком неформальної освіти: ННІ, Освітні програми, Дисципліни, Групи, Здобувачі освіти, Сертифікати, Зарахування неформальної освіти (з'єднані між собою ключами). Відібрані сутності забезпечують комплексний підхід до управління даними про неформальну освіту. Вони дозволяють структуровано зберігати інформацію, ефективно координувати навчальний процес та офіційно визнавати досягнення студентів у сфері неформальної освіти.

Для ефективної організації даних було визначено ієрархію та підлеглисть сутностей. Для кожної сутності були визначені основні атрибути, включаючи ключові атрибути, що забезпечують унікальну ідентифікацію кожного запису та створення структурованої та ефективної моделі даних.

На основі концептуальної моделі було розроблено ER-модель бази даних, яка задовольняє вимоги другої (2НФ) та третьої нормальної форми (3НФ). Це забезпечило ефективну організацію даних, уникнення надлишковості та аномалій при обробці інформації.

Застосунок має інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам легко взаємодіяти з системою без попереднього навчання. Завдяки чіткій структурі та логічній організації елементів користувачі можуть швидко знаходити необхідні функції та виконувати завдання.

Також, однією з ключових можливостей програми є функція додавання файлів сертифікатів. Це забезпечує зручний спосіб зберігання та перегляду документів, та спрощує процес обліку та верифікації документів, що підтверджують досягнення у неформальній освіті

Інтерфейс програми розроблено із використанням елементів стилістики брендбуку Національного університету водного господарства та природокористування. Використання фірмових кольорів, логотипів та шрифтів забезпечує цілісність візуального стилю та підкреслює зв'язок з університетом, створюючи професійне та привабливе враження для користувачів.

На завершальному етапі було проведено всебічне тестування ІС для виявлення та виправлення можливих помилок. Ми провели функціональне, інтеграційне та системне тестування, щоб переконатися у коректній роботі всіх компонентів системи. Ефективність роботи ІС була оцінена на основі швидкості обробки даних, зручності користування та точності обліку неформальної освіти.

Розроблена ІС для обліку неформальної освіти НУВГП повністю відповідає визначеним вимогам та забезпечує ефективне управління даними. Вона покращує точність і швидкість обліку неформальної освіти, зменшує адміністративне навантаження і підвищує загальну якість освітнього процесу в університеті.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ременяк Л.В. Конспект лекцій з навчальної дисципліни "Проектування інформаційних систем". *Вісник Одеського державного екологічного університету. Комп'ютерні науки*. Одеса, 2016. С.10. URL: <http://surl.li/ulvvl> (дата звернення: 02.06.2024).
2. Життєвий цикл розробки інформаційної системи. URL: <https://www.a18.pp.ua/VFP/pdf/lecobdz6.pdf> (дата звернення: 02.06.2024).
3. Моделі і методи проектування інформаційних систем. Проектування інформаційних систем. URL: <http://surl.li/ulvul> (дата звернення: 02.06.2024).
4. TOPDB Top Database index. URL: <https://pypl.github.io/DB.html> (дата звернення: 02.06.2024).
5. DB-Engines Ranking. URL: <https://db-engines.com/en/ranking> (дата звернення: 02.06.2024).
6. Microsoft Access. Матеріал з Вікіпедії. URL: [https://uk.wikipedia.org/wiki/Microsoft\\_Access](https://uk.wikipedia.org/wiki/Microsoft_Access) (дата звернення: 02.06.2024).
7. Моделювання даних та маніпулювання даними. Основи роботи в СУБД Oracle. URL: <http://surl.li/ulvsl> (дата звернення: 02.06.2024).
8. База даних MySQL. URL: <https://promoter.net.ua/articles/baza-danix-mysql.html> (дата звернення: 02.06.2024).
9. TIOBE Index for June 2024. URL: <https://www.tiobe.com/tiobe-index/> (дата звернення: 02.06.2024).
10. PYPL Popularity of Programming Language. URL: <https://pypl.github.io/PYPL.html> (дата звернення: 02.06.2024).
11. Python. Матеріал з Вікіпедії. URL: <https://uk.wikipedia.org/wiki/Python> (дата звернення: 02.06.2024).

12. Java. Матеріал з Вікіпедії. URL: <https://uk.wikipedia.org/wiki/Java> (дата звернення: 02.06.2024).
13. Про мову програмування С#. URL: <https://foxminded.ua/prohramuvannia-na-si/> (дата звернення: 02.06.2024).
14. Соціалізація молоді з особливими освітніми потребами: кращі практики: метод. посіб. / Базиленко А. та ін.; За загальною редакцією А. Базиленко. К.: Талком, 2020. 128 с.
15. Центр неформальної освіти НУВГП. URL: <https://nuwm.edu.ua/strukturni-pidrozdili/centr-neformaljnoji-osviti> (дата звернення: 04.06.2024).
16. Положення про неформальну та інформаційну освіту в Національному університеті водного господарства та природокористування. URL: <http://surl.li/ulvnu> (дата звернення: 04.06.2024).
17. Класифікація баз даних. URL: <https://www.step.org.ua/konspekt/program/tema2> (дата звернення: 04.06.2024).
18. Що таке СУБД і для чого вони потрібні. URL: <https://foxminded.ua/systema-upravlinnia-bazamy-danykh/> (дата звернення: 04.06.2024).
19. Що таке моделювання даних? Типи (концептуальний, логічний, фізичний). URL: <https://www.guru99.com/uk/data-modelling-conceptual-logical.html> (дата звернення: 07.06.2024).
20. Модель «сутність-зв'язок». URL: <http://surl.li/uiwnc> (дата звернення: 07.06.2024).
21. Нормальні форми бази даних. URL: <https://javarush.com/ua/quests/lectures/ua.questhibernate.level17.lecture02> (дата звернення: 07.06.2024).
22. Нормалізація відношень при проектуванні БД. URL: [https://rdb.dp.ua/uk/chapter\\_03](https://rdb.dp.ua/uk/chapter_03) (дата звернення: 07.06.2024).

23. Організація баз даних. Тема 6 - Теорія нормалізації реляційної моделі даних. URL: <http://surl.li/uiwvc> (дата звернення: 07.06.2024).
24. Брендбук Національного університету водного господарства та природокористування. URL: <https://nuwm.edu.ua/publicna-informacija/brendbuk> (дата звернення: 12.06.2024).

## ДОДАТКИ

Додаток А

Код головного вікна застосунку

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WFAApp_NFE
{
    public partial class MainMenu : Form
    {
        public MainMenu()
        {
            InitializeComponent();
        }

        private void pictureBox4_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            MenuOblikNefSystemy newForm = new MenuOblikNefSystemy();
            newForm.ShowDialog();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            MenuZapyty newForm = new MenuZapyty();
            newForm.ShowDialog();
        }

        private void pictureBox3_Click(object sender, EventArgs e)
        {
            Info newForm = new Info();
            newForm.ShowDialog();
        }

        private void pictureBox5_Click(object sender, EventArgs e)
        {
            Dovidnyk_Main newForm = new Dovidnyk_Main();
            newForm.ShowDialog();
        }
    }
}
```

Код вікна «Облік неформальної освіти»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
```

```
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WfApp_NFE
{
    public partial class MenuOblikNefSystemy : Form
    {
        public MenuOblikNefSystemy()
        {
            InitializeComponent();
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btn_NNI_Click(object sender, EventArgs e)
        {
            NNI newForm = new NNI();
            newForm.ShowDialog();
        }

        private void btn_OsvGrupy_Click(object sender, EventArgs e)
        {
            Osvitni_programy newForm = new Osvitni_programy();
            newForm.ShowDialog();
        }

        private void btn_Dyscypliny_Click(object sender, EventArgs e)
        {
            Dyscypliny newForm = new Dyscypliny();
            newForm.ShowDialog();
        }

        private void btn_Grupy_Click(object sender, EventArgs e)
        {
            Grupy newForm = new Grupy();
            newForm.ShowDialog();
        }

        private void btnZdobuvOsv_Click(object sender, EventArgs e)
        {
            Zdobuvachi_osvity newForm = new Zdobuvachi_osvity();
            newForm.ShowDialog();
        }

        private void btn_Sertyfikaty_Click(object sender, EventArgs e)
        {
            Sertyficaty newForm = new Sertyficaty();
            newForm.ShowDialog();
        }

        private void btn_ZarahNef0sv_Click(object sender, EventArgs e)
        {
            Zarahuvannia_neform_osvity newForm = new Zarahuvannia_neform_osvity();
            newForm.ShowDialog();
        }
    }
}
```

## Код вікна «Запити»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
```

```
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WFAApp_NFE
{
    public partial class MenuZapyty : Form
    {
        public MenuZapyty()
        {
            InitializeComponent();
        }

        private void pictureBox5_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnZagalnaInfo_Click(object sender, EventArgs e)
        {
            zapyt_Zagalna_info newForm = new zapyt_Zagalna_info();
            newForm.ShowDialog();
        }

        private void btnDyscypliny_Click(object sender, EventArgs e)
        {
            zapyt_Dyscypliny newForm = new zapyt_Dyscypliny();
            newForm.ShowDialog();
        }

        private void btnGrupy_Click(object sender, EventArgs e)
        {
            zapyt_Grupy newForm = new zapyt_Grupy();
            newForm.ShowDialog();
        }

        private void btnNNI_Click(object sender, EventArgs e)
        {
            zapyt_NNI newForm = new zapyt_NNI();
            newForm.ShowDialog();
        }

        private void btnOsv_Plat_Click(object sender, EventArgs e)
        {
            zapyt_Osv_platformy newForm = new zapyt_Osv_platformy();
            newForm.ShowDialog();
        }

        private void btn_Disc_OsvPlat_Click(object sender, EventArgs e)
        {
            zapyt_Disc_OsvRiven newForm = new zapyt_Disc_OsvRiven();
            newForm.ShowDialog();
        }

        private void btnDyst_OsvRiven_Click(object sender, EventArgs e)
        {
            zapyt_Disc_OsvPlatf newForm = new zapyt_Disc_OsvPlatf();
            newForm.ShowDialog();
        }
    }
}
```

Додаток Б

## Код вікна «ННІ»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WFAApp_NFE
{
    public partial class NNI : Form
    {
        OleDbConnection conn;
        OleDbCommand cmd;
        OleDbDataAdapter adapter;
        DataTable dt;

        public NNI()
        {
            InitializeComponent();
        }

        void GetNNI()
        {
            conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb");
            dt = new DataTable();
            adapter = new OleDbDataAdapter("SELECT *FROM NNI", conn);
            conn.Open();
            adapter.Fill(dt);
            dataGridView1.DataSource = dt;
            conn.Close();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            GetNNI();
            dataGridView1.Columns[0].HeaderText = "ID";
            dataGridView1.Columns[1].HeaderText = "Назва навчально-наукового інституту";
            dataGridView1.Columns[2].HeaderText = "ННІ";
            dataGridView1.Columns[0].Width = 50;
            dataGridView1.Columns[1].Width = 550;
            dataGridView1.Columns[2].Width = 100;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            string query = "INSERT INTO NNI([Nazva NNI], NNI) VALUES(@nazvaNNI, @nni)";
            using (OleDbCommand cmd = new OleDbCommand(query, conn))
            {
                cmd.Parameters.AddWithValue("@nazvaNNI", textBox1.Text);
                cmd.Parameters.AddWithValue("@nni", textBox2.Text);

                try
                {
                    conn.Open();
                    cmd.ExecuteNonQuery();
                    MessageBox.Show("Навчально-науковий інститут додано.");
                    GetNNI();
                }
                catch (Exception ex)
                {
                }
            }
        }
    }
}
```

```
        MessageBox.Show("Помилка: " + ex.Message);
    }
    finally
    {
        conn.Close();
    }
}

private void button2_Click(object sender, EventArgs e)
{
    string query = "UPDATE NNI SET [Nazva NNI] = @nazvaNNI, NNI = @nni WHERE ID = @id";
    using (OleDbCommand cmd = new OleDbCommand(query, conn))
    {
        cmd.Parameters.AddWithValue("@nazvaNNI", textBox1.Text);
        cmd.Parameters.AddWithValue("@nni", textBox2.Text);
        cmd.Parameters.AddWithValue("@id",
Convert.ToInt32(dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["ID"].Value));

        try
        {
            conn.Open();
            cmd.ExecuteNonQuery();
            MessageBox.Show("Навчально-науковий інститут оновлено.");
            GetNNI();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Помилка: " + ex.Message);
        }
        finally
        {
            conn.Close();
        }
    }
}

private void dataGridView1_CellEnter(object sender, DataGridViewCellEventArgs e)
{
    textBox1.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
    textBox2.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
}

private void button3_Click(object sender, EventArgs e)
{
    string query = "DELETE FROM NNI WHERE ID = @id";
    using (OleDbCommand cmd = new OleDbCommand(query, conn))
    {
        cmd.Parameters.AddWithValue("@id",
Convert.ToInt32(dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["ID"].Value));

        try
        {
            conn.Open();
            cmd.ExecuteNonQuery();
            MessageBox.Show("Навчально-науковий інститут видалено.");
            GetNNI();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Помилка: " + ex.Message);
        }
        finally
        {
            conn.Close();
        }
    }
}
```

```
private void button4_Click(object sender, EventArgs e)
{
    textBox1.Text = "";
    textBox2.Text = "";
}

private void pictureBox5_Click(object sender, EventArgs e)
{
    this.Close();
}

private void pictureBox2_Click(object sender, EventArgs e)
{
    Dovidnyk_Oblik1 newForm = new Dovidnyk_Oblik1();
    newForm.ShowDialog();
}
}
```

### Код вікна «Освітні програми»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WFApp_NFE
{
    public partial class Osvitni_programy : Form
    {
        OleDbConnection conn;
        OleDbCommand cmd;
        OleDbDataAdapter adapter;
        DataTable dt;

        public Osvitni_programy()
        {
            InitializeComponent();
            FillComboBox();
        }

        void GetOsvProgramy()
        {
            conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb");
            dt = new DataTable();
            adapter = new OleDbDataAdapter("SELECT *FROM Osvitni_programy", conn);
            conn.Open();
            adapter.Fill(dt);
            dataGridView1.DataSource = dt;
            conn.Close();
        }

        private void FillComboBox()
        {
            string connectionString = @"Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb";
            using (OleDbConnection connection = new OleDbConnection(connectionString))
            {
                string query = "SELECT * FROM NNI";
            }
        }
    }
}
```

```
DataSet dataset = new DataSet();
using (OleDbDataAdapter adapter = new OleDbDataAdapter(query, connection))
{
    adapter.Fill(dataset);
}

foreach (DataRow row in dataset.Tables[0].Rows)
{
    comboBox1.Items.Add(row["Nazva NNI"].ToString());
}

}

private void Osvitni_programy_Load(object sender, EventArgs e)
{
    GetOsvProgramy();
    dataGridView1.Columns[0].HeaderText = "ID";
    dataGridView1.Columns[0].Width = 50;
    dataGridView1.Columns[1].HeaderText = "Назва освітньої програми";
    dataGridView1.Columns[1].Width = 300;
    dataGridView1.Columns[2].HeaderText = "Назва ННІ";
    dataGridView1.Columns[2].Width = 550;
    dataGridView1.Columns[3].HeaderText = "Освітній рівень";
    dataGridView1.Columns[3].Width = 100;
}

private void button1_Click(object sender, EventArgs e)
{
    string query = "INSERT INTO Osvitni_programy([Nazva osv programy], [Nazva NNI], [Osv
riven]) VALUES(@nazvaOsvProg, @nazvaNNI, @osvriven)";
    using (OleDbCommand cmd = new OleDbCommand(query, conn))
    {
        cmd.Parameters.AddWithValue("@nazvaOsvProg", textBox1.Text);
        cmd.Parameters.AddWithValue("@nazvaNNI", comboBox1.Text);
        cmd.Parameters.AddWithValue("@osvriven", comboBox2.Text);

        try
        {
            conn.Open();
            cmd.ExecuteNonQuery();
            MessageBox.Show("Навчально-науковий інститут додано.");
            GetOsvProgramy();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Помилка: " + ex.Message);
        }
        finally
        {
            conn.Close();
        }
    }
}

private void button2_Click(object sender, EventArgs e)
{
    string query = "UPDATE Osvitni_programy SET [Nazva osv programy] = @nazvaOsvProg,
[Nazva NNI] = @nazvaNNI, " +
"[Osv riven] = @osvriven WHERE ID = @id";
    using (OleDbCommand cmd = new OleDbCommand(query, conn))
    {
        cmd.Parameters.AddWithValue("@nazvaOsvProg", textBox1.Text);
        cmd.Parameters.AddWithValue("@nazvaNNI", comboBox1.Text);
        cmd.Parameters.AddWithValue("@osvriven", comboBox2.Text);
        cmd.Parameters.AddWithValue("@id",
Convert.ToInt32(dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["ID"].Value));
    }
}
```

```
try
{
    conn.Open();
    cmd.ExecuteNonQuery();
    MessageBox.Show("Навчально-науковий інститут оновлено.");
    GetOsvProgramy();
}
catch (Exception ex)
{
    MessageBox.Show("Помилка: " + ex.Message);
}
finally
{
    conn.Close();
}
}

private void dataGridView1_CellEnter(object sender, DataGridViewCellEventArgs e)
{
    textBox1.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
    comboBox1.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
    comboBox2.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
}

private void button3_Click(object sender, EventArgs e)
{
    string query = "DELETE FROM Osvitni_programy WHERE ID = @id";
    using (OleDbCommand cmd = new OleDbCommand(query, conn))
    {
        cmd.Parameters.AddWithValue("@id",
Convert.ToInt32(dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["ID"].Value));

        try
        {
            conn.Open();
            cmd.ExecuteNonQuery();
            MessageBox.Show("Навчально-науковий інститут видалено.");
            GetOsvProgramy();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Помилка: " + ex.Message);
        }
        finally
        {
            conn.Close();
        }
    }
}

private void button4_Click(object sender, EventArgs e)
{
    textBox1.Text = "";
    comboBox1.Text = "";
    comboBox2.Text = "";
}

private void pictureBox5_Click(object sender, EventArgs e)
{
    this.Close();
}

private void pictureBox2_Click(object sender, EventArgs e)
{
    Dovidnyk_Oblik1 newForm = new Dovidnyk_Oblik1();
    newForm.ShowDialog();
}
```

```
}  
}
```

## Код меню «Дисципліни»

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using System.Data.OleDb;  
  
namespace WfApp_NFE  
{  
    public partial class Dyscypliny : Form  
    {  
        OleDbConnection conn;  
        OleDbCommand cmd;  
        OleDbDataAdapter adapter;  
        DataTable dt;  
  
        public Dyscypliny()  
        {  
            InitializeComponent();  
            FillComboBox();  
        }  
  
        void GetDyscypliny()  
        {  
            conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data  
Source=NFE_db.accdb");  
            dt = new DataTable();  
            adapter = new OleDbDataAdapter("SELECT *FROM Dyscypliny", conn);  
            conn.Open();  
            adapter.Fill(dt);  
            dataGridView1.DataSource = dt;  
            conn.Close();  
        }  
        private void FillComboBox()  
        {  
            string connectionString = @"Provider=Microsoft.ACE.OLEDB.12.0; Data  
Source=NFE_db.accdb";  
            using (OleDbConnection connection = new OleDbConnection(connectionString))  
            {  
                string query = "SELECT * FROM Osvitni_programy";  
  
                DataSet dataset = new DataSet();  
                using (OleDbDataAdapter adapter = new OleDbDataAdapter(query, connection))  
                {  
                    adapter.Fill(dataset);  
                }  
  
                foreach (DataRow row in dataset.Tables[0].Rows)  
                {  
                    comboBox1.Items.Add(row["Nazva osv programy"].ToString());  
                }  
            }  
        }  
  
        private void Dyscypliny_Load(object sender, EventArgs e)  
        {  

```

```
GetDyscypliny();
dataGridView1.Columns[0].HeaderText = "ID";
dataGridView1.Columns[0].Width = 50;
dataGridView1.Columns[1].HeaderText = "Назва дисципліни";
dataGridView1.Columns[1].Width = 250;
dataGridView1.Columns[2].HeaderText = "Назва освітньої програми";
dataGridView1.Columns[2].Width = 250;
dataGridView1.Columns[3].HeaderText = "К-ть кредитів";
dataGridView1.Columns[3].Width = 75;
dataGridView1.Columns[4].HeaderText = "Форма контролю";
dataGridView1.Columns[4].Width = 100;
}

private void button1_Click(object sender, EventArgs e)
{
    string query = "INSERT INTO Dyscypliny([Nazva dyscypliny], [Nazva osv programy], [k-t
kredytiv ECTS], [Forma kontroly]) VALUES(@nazvaDyscypliny, " +
"@nazvaOsvProg, @ktKredytiv, @formaKontroly)";
    using (OleDbCommand cmd = new OleDbCommand(query, conn))
    {
        cmd.Parameters.AddWithValue("@nazvaDyscypliny", textBox1.Text);
        cmd.Parameters.AddWithValue("@nazvaOsvProg", comboBox1.Text);
        cmd.Parameters.AddWithValue("@ktKredytiv", textBox2.Text);
        cmd.Parameters.AddWithValue("@formaKontroly", comboBox2.Text);

        try
        {
            conn.Open();
            cmd.ExecuteNonQuery();
            MessageBox.Show("Дисципліну додано.");
            GetDyscypliny();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Помилка: " + ex.Message);
        }
        finally
        {
            conn.Close();
        }
    }
}

private void button2_Click(object sender, EventArgs e)
{
    string query = "UPDATE Dyscypliny SET [Nazva dyscypliny] = @nazvaDyscypliny, [Nazva
osv programy] = @nazvaOsvProg, " +
"[k-t kredytiv ECTS] = @ktKredytiv, [Forma kontroly] = @formaKontroly WHERE ID =
@id";
    using (OleDbCommand cmd = new OleDbCommand(query, conn))
    {
        cmd.Parameters.AddWithValue("@nazvaDyscypliny", textBox1.Text);
        cmd.Parameters.AddWithValue("@nazvaOsvProg", comboBox1.Text);
        cmd.Parameters.AddWithValue("@ktKredytiv", textBox2.Text);
        cmd.Parameters.AddWithValue("@formaKontroly", comboBox2.Text);
        cmd.Parameters.AddWithValue("@id",
Convert.ToInt32(dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["ID"].Value));

        try
        {
            conn.Open();
            cmd.ExecuteNonQuery();
            MessageBox.Show("Дисципліну оновлено.");
            GetDyscypliny();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Помилка: " + ex.Message);
        }
    }
}
```

```
    }  
    finally  
    {  
        conn.Close();  
    }  
}  
  
private void dataGridView1_CellEnter(object sender, DataGridViewCellEventArgs e)  
{  
    textBox1.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();  
    comboBox1.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();  
    textBox2.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();  
    comboBox2.Text = dataGridView1.CurrentRow.Cells[4].Value.ToString();  
}  
  
private void button3_Click(object sender, EventArgs e)  
{  
    string query = "DELETE FROM Dyscipliny WHERE ID = @id";  
    using (OleDbCommand cmd = new OleDbCommand(query, conn))  
    {  
        cmd.Parameters.AddWithValue("@id",  
Convert.ToInt32(dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["ID"].Value));  
  
        try  
        {  
            conn.Open();  
            cmd.ExecuteNonQuery();  
            MessageBox.Show("Дисципліну видалено.");  
            GetDyscipliny();  
        }  
        catch (Exception ex)  
        {  
            MessageBox.Show("Помилка: " + ex.Message);  
        }  
        finally  
        {  
            conn.Close();  
        }  
    }  
}  
  
private void button4_Click(object sender, EventArgs e)  
{  
    textBox1.Text = "";  
    comboBox1.Text = "";  
    textBox2.Text = "";  
    comboBox2.Text = "";  
}  
  
private void pictureBox5_Click_1(object sender, EventArgs e)  
{  
    this.Close();  
}  
  
private void pictureBox2_Click(object sender, EventArgs e)  
{  
    Dovidnyk_Oblik1 newForm = new Dovidnyk_Oblik1();  
    newForm.ShowDialog();  
}  
}
```

## Код вікна «Групи»

```
using System;  
using System.Collections.Generic;
```

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WFAApp_NFE
{
    public partial class Grupy : Form
    {
        OleDbConnection conn;
        OleDbCommand cmd;
        OleDbDataAdapter adapter;
        DataTable dt;

        public Grupy()
        {
            InitializeComponent();
            FillComboBox();
        }

        void GetGrupy()
        {
            conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb");
            dt = new DataTable();
            adapter = new OleDbDataAdapter("SELECT *FROM Grupy", conn);
            conn.Open();
            adapter.Fill(dt);
            dataGridView1.DataSource = dt;
            conn.Close();
        }

        private void FillComboBox()
        {
            string connectionString = @"Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb";
            using (OleDbConnection connection = new OleDbConnection(connectionString))
            {
                string query = "SELECT * FROM Osvitni_programy";

                DataSet dataset = new DataSet();
                using (OleDbDataAdapter adapter = new OleDbDataAdapter(query, connection))
                {
                    adapter.Fill(dataset);
                }

                foreach (DataRow row in dataset.Tables[0].Rows)
                {
                    comboBox1.Items.Add(row["Nazva osv programy"].ToString());
                }
            }
        }

        private void Grupy_Load(object sender, EventArgs e)
        {
            GetGrupy();
            dataGridView1.Columns[0].HeaderText = "ID";
            dataGridView1.Columns[0].Width = 50;
            dataGridView1.Columns[1].HeaderText = "Назва групи";
            dataGridView1.Columns[1].Width = 100;
            dataGridView1.Columns[2].HeaderText = "Назва освітньої програми";
            dataGridView1.Columns[2].Width = 300;
            dataGridView1.Columns[3].HeaderText = "Курс";
            dataGridView1.Columns[3].Width = 50;
            dataGridView1.Columns[4].HeaderText = "Форма навчання";
        }
    }
}
```

```
        dataGridView1.Columns[4].Width = 100;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        string query = "INSERT INTO Grupy([Nazva grupy], [Nazva osv programy], Kurs, [Forma
navchannia]) VALUES(@nazvaGrupy, " +
"@nazvaOsvProg, @kurs, @formaNavchannia)";
        using (OleDbCommand cmd = new OleDbCommand(query, conn))
        {
            cmd.Parameters.AddWithValue("@nazvaGrupy", textBox1.Text);
            cmd.Parameters.AddWithValue("@nazvaOsvProg", comboBox1.Text);
            cmd.Parameters.AddWithValue("@kurs", textBox2.Text);
            cmd.Parameters.AddWithValue("@formaNavchannia", comboBox2.Text);

            try
            {
                conn.Open();
                cmd.ExecuteNonQuery();
                MessageBox.Show("Групу додано.");
                GetGrupy();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Помилка: " + ex.Message);
            }
            finally
            {
                conn.Close();
            }
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        string query = "UPDATE Grupy SET [Nazva grupy] = @nazvaGrupy, [Nazva osv programy] =
@nazvaOsvProg, " +
"@kurs = @kurs, [Forma navchannia] = @formaNavchannia WHERE ID = @id";
        using (OleDbCommand cmd = new OleDbCommand(query, conn))
        {
            cmd.Parameters.AddWithValue("@nazvaGrupy", textBox1.Text);
            cmd.Parameters.AddWithValue("@nazvaOsvProg", comboBox1.Text);
            cmd.Parameters.AddWithValue("@kurs", textBox2.Text);
            cmd.Parameters.AddWithValue("@formaNavchannia", comboBox2.Text);
            cmd.Parameters.AddWithValue("@id",
Convert.ToInt32(dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["ID"].Value));

            try
            {
                conn.Open();
                cmd.ExecuteNonQuery();
                MessageBox.Show("Групу оновлено.");
                GetGrupy();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Помилка: " + ex.Message);
            }
            finally
            {
                conn.Close();
            }
        }
    }

    private void dataGridView1_CellEnter(object sender, DataGridViewCellEventArgs e)
    {
        textBox1.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
    }
}
```

```
comboBox1.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
textBox2.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
comboBox2.Text = dataGridView1.CurrentRow.Cells[4].Value.ToString();
}

private void button3_Click(object sender, EventArgs e)
{
    string query = "DELETE FROM Grupy WHERE ID = @id";
    using (OleDbCommand cmd = new OleDbCommand(query, conn))
    {
        cmd.Parameters.AddWithValue("@id",
Convert.ToInt32(dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["ID"].Value));

        try
        {
            conn.Open();
            cmd.ExecuteNonQuery();
            MessageBox.Show("Групу видалено.");
            GetGrupy();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Помилка: " + ex.Message);
        }
        finally
        {
            conn.Close();
        }
    }
}

private void button4_Click(object sender, EventArgs e)
{
    textBox1.Text = "";
    comboBox1.Text = "";
    textBox2.Text = "";
    comboBox2.Text = "";
}

private void pictureBox5_Click(object sender, EventArgs e)
{
    this.Close();
}

private void pictureBox2_Click(object sender, EventArgs e)
{
    Dovidnyk_Oblik1 newForm = new Dovidnyk_Oblik1();
    newForm.ShowDialog();
}
}
```

## Код вікна «Здобувачі освіти»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WFApp_NFE
{
```

```
public partial class Zdobuvachi_osvity : Form
{
    OleDbConnection conn;
    OleDbCommand cmd;
    OleDbDataAdapter adapter;
    DataTable dt;

    public Zdobuvachi_osvity()
    {
        InitializeComponent();
        FillComboBox();
    }

    void GetZdobuvachiOsvity()
    {
        conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb");
        dt = new DataTable();
        adapter = new OleDbDataAdapter("SELECT *FROM Zdobuvachi_osvity", conn);
        conn.Open();
        adapter.Fill(dt);
        dataGridView1.DataSource = dt;
        conn.Close();
    }

    private void FillComboBox()
    {
        string connectionString = @"Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb";
        using (OleDbConnection connection = new OleDbConnection(connectionString))
        {
            string query = "SELECT * FROM Grupy";

            DataSet dataset = new DataSet();
            using (OleDbDataAdapter adapter = new OleDbDataAdapter(query, connection))
            {
                adapter.Fill(dataset);
            }

            foreach (DataRow row in dataset.Tables[0].Rows)
            {
                comboBox1.Items.Add(row["Nazva grupy"].ToString());
            }
        }
    }

    private void Zdobuvachi_osvity_Load(object sender, EventArgs e)
    {
        GetZdobuvachiOsvity();
        dataGridView1.Columns[0].HeaderText = "Номер студентського квитка";
        dataGridView1.Columns[0].Width = 100;
        dataGridView1.Columns[1].HeaderText = "Назва групи";
        dataGridView1.Columns[1].Width = 100;
        dataGridView1.Columns[2].HeaderText = "ПІБ";
        dataGridView1.Columns[2].Width = 200;
        dataGridView1.Columns[3].HeaderText = "E-mail";
        dataGridView1.Columns[3].Width = 200;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        string query = "INSERT INTO Zdobuvachi_osvity([Nomer stud kvytka], [Nazva grupy],
PIB, [e-mail]) " +
"VALUES(@nomerStudKvytka, @nazvaGrupy, @pib, @email)";
        using (OleDbCommand cmd = new OleDbCommand(query, conn))
        {
            cmd.Parameters.AddWithValue("@nomerStudKvytka", textBox1.Text);
            cmd.Parameters.AddWithValue("@nazvaGrupy", comboBox1.Text);
            cmd.Parameters.AddWithValue("@pib", textBox2.Text);
        }
    }
}
```

```
cmd.Parameters.AddWithValue("@email", textBox3.Text);

try
{
    conn.Open();
    cmd.ExecuteNonQuery();
    MessageBox.Show("Дані про студента додано.");
    GetZdobuvachiOsvity();
}
catch (Exception ex)
{
    MessageBox.Show("Помилка: " + ex.Message);
}
finally
{
    conn.Close();
}
}

private void button2_Click(object sender, EventArgs e)
{
    string query = "UPDATE Zdobuvachi_osvity SET [Nomer stud kvytka] = @nomerStudKvytka,
[Nazva grupy] = @nazvaGrupy, " +
    "PIB = @pib, [e-mail] = @email WHERE [Nomer stud kvytka] = @nomerStudKvytka";
    using (OleDbCommand cmd = new OleDbCommand(query, conn))
    {
        cmd.Parameters.AddWithValue("@nomerStudKvytka", textBox1.Text);
        cmd.Parameters.AddWithValue("@nazvaGrupy", comboBox1.Text);
        cmd.Parameters.AddWithValue("@pib", textBox2.Text);
        cmd.Parameters.AddWithValue("@email", textBox3.Text);

        try
        {
            conn.Open();
            cmd.ExecuteNonQuery();
            MessageBox.Show("Дані студента оновлено.");
            GetZdobuvachiOsvity();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Помилка: " + ex.Message);
        }
        finally
        {
            conn.Close();
        }
    }
}

private void dataGridView1_CellEnter(object sender, DataGridViewCellEventArgs e)
{
    textBox1.Text = dataGridView1.CurrentRow.Cells[0].Value.ToString();
    comboBox1.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
    textBox2.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
    textBox3.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
}

private void button3_Click(object sender, EventArgs e)
{
    string query = "DELETE FROM Zdobuvachi_osvity WHERE [Nomer stud kvytka] =
@nomerStudKvytka";
    using (OleDbCommand cmd = new OleDbCommand(query, conn))
    {
        cmd.Parameters.AddWithValue("@nomerStudKvytka",
Convert.ToInt32(dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["Nomer stud
kvytka"].Value));
    }
}
```

```
        try
        {
            conn.Open();
            cmd.ExecuteNonQuery();
            MessageBox.Show("Дані студента видалено.");
            GetZdobuvachi0svity();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Помилка: " + ex.Message);
        }
        finally
        {
            conn.Close();
        }
    }
}

private void button4_Click(object sender, EventArgs e)
{
    textBox1.Text = "";
    comboBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
}

private void pictureBox5_Click(object sender, EventArgs e)
{
    this.Close();
}

private void pictureBox4_Click(object sender, EventArgs e)
{
    Dovidnyk_Oblik1 newForm = new Dovidnyk_Oblik1();
    newForm.ShowDialog();
}
}
}
```

#### Код вікна «Сертифікати»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WFApp_NFE
{
    public partial class Sertyficaty : Form
    {
        OleDbConnection conn;
        OleDbCommand cmd;
        OleDbDataAdapter adapter;
        DataTable dt;

        public Sertyficaty()
        {
            InitializeComponent();
            FillComboBox();
        }
    }
}
```

```
void GetSertyficaty()
{
    conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb");
    dt = new DataTable();
    adapter = new OleDbDataAdapter("SELECT *FROM Sertyfikaty", conn);
    conn.Open();
    adapter.Fill(dt);
    conn.Close();
}

private void FillComboBox()
{
    string connectionString = @"Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb";
    using (OleDbConnection connection = new OleDbConnection(connectionString))
    {
        string query = "SELECT * FROM Zdobuvachi_osvity";

        DataSet dataset = new DataSet();
        using (OleDbDataAdapter adapter = new OleDbDataAdapter(query, connection))
        {
            adapter.Fill(dataset);

        }

        foreach (DataRow row in dataset.Tables[0].Rows)
        {
            comboBox1.Items.Add(row["Nomer stud kvytka"].ToString());
        }

        DataSet dataset2 = new DataSet();
        using (OleDbDataAdapter adapter = new OleDbDataAdapter(query, connection))
        {
            adapter.Fill(dataset2);

        }

        foreach (DataRow row in dataset2.Tables[0].Rows)
        {
            comboBox2.Items.Add(row["PIB"].ToString());
        }
    }
}

private void Sertyficaty2_Load(object sender, EventArgs e)
{
    this.sertyfikatyTableAdapter.Fill(this.nFE_dbDataSet.Sertyfikaty);
    sertyfikatyBindingSource.DataSource = this.nFE_dbDataSet.Sertyfikaty;
    GetSertyficaty();
}

private void btnSave_Click(object sender, EventArgs e)
{
    try
    {
        sertyfikatyBindingSource.EndEdit();
        sertyfikatyTableAdapter.Update(this.nFE_dbDataSet.Sertyfikaty);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Message", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        sertyfikatyBindingSource.ResetBindings(false);
    }
}

private void btnBrowse_Click(object sender, EventArgs e)
```

```

    {
        try
        {
            using (OpenFileDialog openFileDialog = new OpenFileDialog() { Filter = "JPEG files|
*.jpg", ValidateNames = true, Multiselect = false })
            {
                if (fileDialog.ShowDialog() == DialogResult.OK)
                {
                    pictureBox1.Image = Image.FromFile(fileDialog.FileName);
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Помилка: " + ex.Message);
        }
    }

    private void btnDelete_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("Ви справді хочете видалити сертифікат?", "Message",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            sertyfikatyBindingSource.RemoveCurrent();
    }

    private void txtSearch_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == (char)13)
        {
            if (string.IsNullOrEmpty(txtSearch.Text))
                dataGridView1.DataSource = sertyfikatyBindingSource;
            else
            {
                var query = from o in this.nFE_dbDataSet.Sertyfikaty
                where o.PIB.Contains(txtSearch.Text) ||
                o.Nomer_sertyficatu.Contains(txtSearch.Text)
                select o;
                dataGridView1.DataSource = query.ToList();
            }
        }
    }

    private void btnClear_Click(object sender, EventArgs e)
    {
        textBox1.Text = "";
        comboBox1.Text = "";
        comboBox2.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        textBox4.Text = "";
        textBox5.Text = "";
        dateTimePicker1.Text = "";
        pictureBox1.Image = null;
    }

    private void txtSearch_TextChanged(object sender, EventArgs e)
    {
        DataView dv = dt.DefaultView;
        dv.RowFilter = $"[Nomer_sertyficatu] LIKE '{txtSearch.Text}%' OR " +
        $"[Nomer_stud_kvytka] LIKE '{txtSearch.Text}%' OR " +
        $"[PIB] LIKE '{txtSearch.Text}%' OR " +
        $"[Nazva_osv_platfomy] LIKE '{txtSearch.Text}%' OR " +
        $"[Nazva_kursu] LIKE '{txtSearch.Text}%' OR " +
        $"CONVERT([Data], 'System.String') LIKE '{txtSearch.Text}%' OR " +
        $"CONVERT([Nomer_sertyficatu], 'System.String') LIKE '{txtSearch.Text}%' OR " +
        $"[URL] LIKE '{txtSearch.Text}%'";
        dataGridView1.DataSource = dv;
    }

```

```
    }

    private void btnNew_Click(object sender, EventArgs e)
    {
        try
        {
            panel1.Enabled = true;
            textBox1.Focus();

            string query = "INSERT INTO Sertyfikaty([Nomer sertyficatu])
VALUES(@nomerSertyficatu)";
            using (OleDbCommand cmd = new OleDbCommand(query, conn))
            {
                cmd.Parameters.AddWithValue("@nomerSertyficatu", textBox1.Text);
                try
                {
                    conn.Open();
                    cmd.ExecuteNonQuery();
                    MessageBox.Show("Новий номер сертифікату додано.");
                    this.Hide();
                }
                catch (Exception ex)
                {
                    MessageBox.Show("Помилка: " + ex.Message);
                    this.Hide();
                }
            }
            finally
            {
                conn.Close();
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Message", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
            sertyfikatyBindingSource.ResetBindings(false);
        }

        Sertyficaty form = new Sertyficaty();
        form.Show();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        this.Hide();
        Sertyficaty form = new Sertyficaty();
        form.Show();
    }

    private void pictureBox5_Click_1(object sender, EventArgs e)
    {
        this.Close();
    }

    private void pictureBox4_Click(object sender, EventArgs e)
    {
        Dovidnyk_Sertyfikaty form = new Dovidnyk_Sertyfikaty();
        form.Show();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        pictureBox1.Image = null;
    }
}
}
```

## Код вікна «Зарахування неформальної освіти»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WfApp_NFE
{
    public partial class Zarahuvannia_neform_osvity : Form
    {
        OleDbConnection conn;
        OleDbCommand cmd;
        OleDbDataAdapter adapter;
        DataTable dt;

        public Zarahuvannia_neform_osvity()
        {
            InitializeComponent();
            FillComboBox();
        }

        void GetZarahuvanniaNeformOsvity()
        {
            conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb");
            dt = new DataTable();
            adapter = new OleDbDataAdapter("SELECT *FROM Zarahuvannia_neform_osvity", conn);
            conn.Open();
            adapter.Fill(dt);
            dataGridView1.DataSource = dt;
            conn.Close();
        }

        private void FillComboBox()
        {
            string connectionString = @"Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb";
            using (OleDbConnection connection = new OleDbConnection(connectionString))
            {
                string query = "SELECT * FROM Sertyfikaty";

                DataSet dataset = new DataSet();
                using (OleDbDataAdapter adapter = new OleDbDataAdapter(query, connection))
                {
                    adapter.Fill(dataset);
                }

                foreach (DataRow row in dataset.Tables[0].Rows)
                {
                    comboBox1.Items.Add(row["Nomer_sertyficatu"].ToString());
                }

                string query2 = "SELECT * FROM Dyscypliny";

                DataSet dataset2 = new DataSet();
                using (OleDbDataAdapter adapter = new OleDbDataAdapter(query2, connection))
                {
                    adapter.Fill(dataset2);
                }
            }
        }
    }
}
```

```
        foreach (DataRow row in dataset2.Tables[0].Rows)
        {
            comboBox2.Items.Add(row["Nazva dyscypliny"].ToString());
        }
    }

    private void Zarahuvannia_neform_osvity_Load(object sender, EventArgs e)
    {
        GetZarahuvanniaNeformOsvity();
        foreach (DataGridViewColumn column in dataGridView1.Columns)
        {
            column.DefaultCellStyle.Font = new Font(dataGridView1.Font, FontStyle.Regular);
        }
        dataGridView1.Columns[0].Visible = false;
        dataGridView1.Columns[1].HeaderText = "Номер сертифікату";
        dataGridView1.Columns[1].Width = 100;
        dataGridView1.Columns[2].HeaderText = "Назва дисципліни";
        dataGridView1.Columns[2].Width = 250;
        dataGridView1.Columns[3].HeaderText = "Номер протоколу";
        dataGridView1.Columns[3].Width = 75;
        dataGridView1.Columns[4].HeaderText = "Дата протоколу";
        dataGridView1.Columns[4].Width = 100;
        dataGridView1.Columns[5].HeaderText = "Оцінка";
        dataGridView1.Columns[5].Width = 75;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        string query = "INSERT INTO Zarahuvannia_neform_osvity([Nomer sertyficatu], [Nazva
dyscypliny], [Nomer protokolu], [Data protokolu], Ocinka) " +
            "VALUES(@nomerSertyfikatu, @nazvaDyscypliny, @nomerProtokolu, @dataProtokolu,
@ocinka)";
        using (OleDbCommand cmd = new OleDbCommand(query, conn))
        {
            cmd.Parameters.AddWithValue("@nomerSertyfikatu", comboBox1.Text);
            cmd.Parameters.AddWithValue("@nazvaDyscypliny", comboBox2.Text);
            cmd.Parameters.AddWithValue("@nomerProtokolu", textBox1.Text);
            cmd.Parameters.AddWithValue("@dataProtokolu", dateTimePicker1.Value);
            cmd.Parameters.AddWithValue("@ocinka", textBox2.Text);

            try
            {
                conn.Open();
                cmd.ExecuteNonQuery();
                MessageBox.Show("Дані про зарахування додано.");
                GetZarahuvanniaNeformOsvity();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Помилка: " + ex.Message);
            }
            finally
            {
                conn.Close();
            }
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        string query = "UPDATE Zarahuvannia_neform_osvity SET [Nomer sertyficatu] =
@nomerSertyfikatu, [Nazva dyscypliny] = @nazvaDyscypliny, " +
            "[Nomer protokolu] = @nomerProtokolu, [Data protokolu] = @dataProtokolu, [Ocinka]
= @ocinka WHERE ID = @id";
        using (OleDbCommand cmd = new OleDbCommand(query, conn))
        {
            cmd.Parameters.AddWithValue("@nomerSertyfikatu", comboBox1.Text);
```

```
cmd.Parameters.AddWithValue("@nazvaDyscypliny", comboBox2.Text);
cmd.Parameters.AddWithValue("@nomerProtokolu", textBox1.Text);
cmd.Parameters.AddWithValue("@dataProtokolu", dateTimePicker1.Value);
cmd.Parameters.AddWithValue("@ocinka", textBox2.Text);
cmd.Parameters.AddWithValue("@id",
Convert.ToInt32(dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["ID"].Value));

try
{
    conn.Open();
    cmd.ExecuteNonQuery();
    MessageBox.Show("Дані про зарахування оновлено.");
    GetZarahunanniaNeformOsvity();
}
catch (Exception ex)
{
    MessageBox.Show("Помилка: " + ex.Message);
}
finally
{
    conn.Close();
}
}
}

private void button4_Click(object sender, EventArgs e)
{
    comboBox1.Text = "";
    comboBox2.Text = "";
    textBox1.Text = "";
    dateTimePicker1.Text = "";
    textBox2.Text = "";
}

private void dataGridView1_CellEnter(object sender, DataGridViewCellEventArgs e)
{
    comboBox1.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
    comboBox2.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
    textBox1.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
    dateTimePicker1.Text = dataGridView1.CurrentRow.Cells[4].Value.ToString();
    textBox2.Text = dataGridView1.CurrentRow.Cells[5].Value.ToString();
}

private void button3_Click(object sender, EventArgs e)
{
    string query = "DELETE FROM Zarahunannia_neform_osvity WHERE ID = @id";
    using (OleDbCommand cmd = new OleDbCommand(query, conn))
    {
        cmd.Parameters.AddWithValue("@id",
Convert.ToInt32(dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["ID"].Value));

try
{
    conn.Open();
    cmd.ExecuteNonQuery();
    MessageBox.Show("Зарахування видалено.");
    GetZarahunanniaNeformOsvity();
}
catch (Exception ex)
{
    MessageBox.Show("Помилка: " + ex.Message);
}
finally
{
    conn.Close();
}
}
}
```

```
private void pictureBox5_Click(object sender, EventArgs e)
{
    this.Close();
}

private void pictureBox4_Click(object sender, EventArgs e)
{
    Dovidnyk_Oblik1 newForm = new Dovidnyk_Oblik1();
    newForm.ShowDialog();
}
}
```

Додаток В

## Код вікна запиту «Загальна інформація»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WApp_NFE
{
    public partial class zapyt_Zagalna_info : Form
    {
        OleDbConnection conn;
        OleDbCommand cmd;
        OleDbDataAdapter adapter;
        DataTable dt;

        public zapyt_Zagalna_info()
        {
            InitializeComponent();
        }

        void GetSimple_Summary_Info()
        {
            conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb");
            dt = new DataTable();
            adapter = new OleDbDataAdapter("SELECT *FROM Simple_Summary_Info", conn);
            conn.Open();
            adapter.Fill(dt);
            dataGridView1.DataSource = dt;
            conn.Close();
        }

        private void zapyt_Zagalna_info_Load(object sender, EventArgs e)
        {
            GetSimple_Summary_Info();
            dataGridView1.Columns[0].HeaderText = "HHI";
            dataGridView1.Columns[0].Width = 70;
            dataGridView1.Columns[1].HeaderText = "Назва групи";
            dataGridView1.Columns[1].Width = 70;
            dataGridView1.Columns[2].HeaderText = "Освітній рівень";
            dataGridView1.Columns[2].Width = 70;
            dataGridView1.Columns[3].HeaderText = "ПІБ";
            dataGridView1.Columns[3].Width = 200;
            dataGridView1.Columns[4].HeaderText = "Освітня платформа";
            dataGridView1.Columns[4].Width = 120;
            dataGridView1.Columns[5].HeaderText = "Курс";
        }
    }
}
```

```

dataGridView1.Columns[5].Width = 280;
dataGridView1.Columns[6].HeaderText = "Дисципліна";
dataGridView1.Columns[6].Width = 225;
dataGridView1.Columns[7].HeaderText = "Оцінка";
dataGridView1.Columns[7].Width = 50;
dataGridView1.Columns[8].HeaderText = "№ протоколу";
dataGridView1.Columns[8].Width = 70;
dataGridView1.Columns[9].HeaderText = "Дата";
dataGridView1.Columns[9].Width = 75;
}

private void pictureBox5_Click(object sender, EventArgs e)
{
    this.Close();
}
}
}
}

```

## Код вікна запиту «Дисципліни»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WFAApp_NFE
{
    public partial class zapyt_Dyscypliny : Form
    {
        OleDbConnection conn;
        OleDbCommand cmd;
        OleDbDataAdapter adapter;
        DataTable dt;

        public zapyt_Dyscypliny()
        {
            InitializeComponent();
        }

        void GetDyscypliny_Stat()
        {
            conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb");
            dt = new DataTable();
            adapter = new OleDbDataAdapter("SELECT *FROM Dyscypliny_Stat", conn);
            conn.Open();
            adapter.Fill(dt);
            dataGridView1.DataSource = dt;
            conn.Close();
        }

        private void Dyscypliny_Stat_Load(object sender, EventArgs e)
        {
            GetDyscypliny_Stat();
            dataGridView1.Columns[0].HeaderText = "Назва дисципліни";
            dataGridView1.Columns[0].Width = 230;
            dataGridView1.Columns[1].HeaderText = "К-ть";
            dataGridView1.Columns[1].Width = 50;
        }

        private void pictureBox5_Click(object sender, EventArgs e)

```

```
    {  
        this.Close();  
    }  
} }
```

## Код вікна запиту «Групи»

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using System.Data.OleDb;  
  
namespace WFApp_NFE  
{  
    public partial class zapyt_Grupy : Form  
    {  
        OleDbConnection conn;  
        OleDbCommand cmd;  
        OleDbDataAdapter adapter;  
        DataTable dt;  
  
        public zapyt_Grupy()  
        {  
            InitializeComponent();  
        }  
  
        void GetGrupy_Stat()  
        {  
            conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data  
Source=NFE_db.accdb");  
            dt = new DataTable();  
            adapter = new OleDbDataAdapter("SELECT *FROM Grupy_Stat", conn);  
            conn.Open();  
            adapter.Fill(dt);  
            dataGridView1.DataSource = dt;  
            conn.Close();  
        }  
  
        private void zapyt_Grupy_Load(object sender, EventArgs e)  
        {  
            GetGrupy_Stat();  
            dataGridView1.Columns[0].HeaderText = "Група";  
            dataGridView1.Columns[0].Width = 100;  
            dataGridView1.Columns[1].HeaderText = "К-ть";  
            dataGridView1.Columns[1].Width = 60;  
        }  
  
        private void pictureBox5_Click(object sender, EventArgs e)  
        {  
            this.Close();  
        }  
    }  
}
```

## Код вікна запиту «ННІ»

```
using System;  
using System.Collections.Generic;
```

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WFAApp_NFE
{
    public partial class zapyt_NNI : Form
    {
        OleDbConnection conn;
        OleDbCommand cmd;
        OleDbDataAdapter adapter;
        DataTable dt;

        public zapyt_NNI()
        {
            InitializeComponent();
        }

        void GetNNI_Stat()
        {
            conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb");
            dt = new DataTable();
            adapter = new OleDbDataAdapter("SELECT *FROM NNI_Stat", conn);
            conn.Open();
            adapter.Fill(dt);
            dataGridView1.DataSource = dt;
            conn.Close();
        }

        private void pictureBox5_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void zapyt_NNI_Load(object sender, EventArgs e)
        {
            GetNNI_Stat();
            dataGridView1.Columns[0].HeaderText = "HHI";
            dataGridView1.Columns[0].Width = 100;
            dataGridView1.Columns[1].HeaderText = "К-ть";
            dataGridView1.Columns[1].Width = 60;
        }
    }
}
```

## Код вікна запиту «Освітні платформи»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WFAApp_NFE
{
    public partial class zapyt_Osv_platformy : Form
```

```
{
    OleDbConnection conn;
    OleDbCommand cmd;
    OleDbDataAdapter adapter;
    DataTable dt;

    public zapyt_Osv_platformy()
    {
        InitializeComponent();
    }

    void GetOsv_Plat_Stat()
    {
        conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb");
        dt = new DataTable();
        adapter = new OleDbDataAdapter("SELECT *FROM OcvPlatformy_Stat", conn);
        conn.Open();
        adapter.Fill(dt);
        dataGridView1.DataSource = dt;
        conn.Close();
    }

    private void zapyt_Osv_platformy_Load(object sender, EventArgs e)
    {
        GetOsv_Plat_Stat();
        dataGridView1.Columns[0].HeaderText = "Освітня платформа";
        dataGridView1.Columns[0].Width = 150;
        dataGridView1.Columns[1].HeaderText = "К-ть";
        dataGridView1.Columns[1].Width = 60;
    }

    private void pictureBox5_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}
}
```

Код вікна запиту «Статистика по дисциплінах та освітніх рівнях»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WFApp_NFE
{
    public partial class zapyt_Dysc_OsvRiven : Form
    {
        OleDbConnection conn;
        OleDbCommand cmd;
        OleDbDataAdapter adapter;
        DataTable dt;

        public zapyt_Dysc_OsvRiven()
        {
            InitializeComponent();
        }
    }
}
```

```
void GetDysc_OsvRiven()
{
    conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb");
    dt = new DataTable();
    adapter = new OleDbDataAdapter("SELECT *FROM Cross_Osv_riven1", conn);
    conn.Open();
    adapter.Fill(dt);
    dataGridView1.DataSource = dt;
    conn.Close();
}

private void zapyt_Dysc_OsvPlat_Load(object sender, EventArgs e)
{
    GetDysc_OsvRiven();
    dataGridView1.Columns[0].HeaderText = "Дисципліна";
    dataGridView1.Columns[0].Width = 250;
    dataGridView1.Columns[1].HeaderText = "Усього";
    dataGridView1.Columns[1].Width = 60;
    dataGridView1.Columns[2].HeaderText = "Бакалаври";
    dataGridView1.Columns[2].Width = 80;
    dataGridView1.Columns[3].HeaderText = "Магістри";
    dataGridView1.Columns[3].Width = 80;
}

private void pictureBox5_Click(object sender, EventArgs e)
{
    this.Close();
}
}
```

#### Код вікна запиту «Статистика по дисциплінах та освітніх платформах»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WfApp_NFE
{
    public partial class zapyt_Dysc_OsvPlatf : Form
    {
        OleDbConnection conn;
        OleDbCommand cmd;
        OleDbDataAdapter adapter;
        DataTable dt;

        public zapyt_Dysc_OsvPlatf()
        {
            InitializeComponent();
        }

        void GetDysc_OsvPlat()
        {
            conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=NFE_db.accdb");
            dt = new DataTable();
            adapter = new OleDbDataAdapter("SELECT *FROM Cross_Osv_platf1", conn);
            conn.Open();
        }
    }
}
```

```
        adapter.Fill(dt);
        dataGridView1.DataSource = dt;
        conn.Close();
    }

    private void zapyt_Dysc_OsvPlatf_Load(object sender, EventArgs e)
    {
        GetDysc_OsvPlatf();
        dataGridView1.Columns[0].HeaderText = "Дисципліна";
        dataGridView1.Columns[0].Width = 250;
        dataGridView1.Columns[1].HeaderText = "Усього";
        dataGridView1.Columns[1].Width = 60;
        dataGridView1.Columns[2].Width = 80;
        dataGridView1.Columns[3].Width = 80;
        dataGridView1.Columns[4].Width = 100;
    }
    private void pictureBox5_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}
}
```

Додаток Г

## Код вікна «Login»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WFApp_NFE
{
    public partial class Login : Form
    {
        private OleDbConnection conn = new OleDbConnection();

        public Login()
        {
            InitializeComponent();

            conn.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\HYBГП\4
КУРС\2 сем\ДИПЛОМКА\NFE app project\WFApp_NFE\bin\Debug\NFE_db.accdb;Persist Security
Info=False";
        }

        private void button1_Click(object sender, EventArgs e)
        {
            conn.Open();
            OleDbCommand cmd = new OleDbCommand();
            cmd.Connection = conn;
            cmd.CommandText = "SELECT *FROM Users WHERE Username = '" + textBox1.Text + "' AND
Password = '" + textBox2.Text + "'";
            OleDbDataReader or = cmd.ExecuteReader();

            int count = 0;
            while (or.Read())
```

```
        {
            count = count + 1;
        }
        if (count == 1)
        {
            this.Hide();
            MainMenu newForm = new MainMenu();
            newForm.ShowDialog();
        }
        else
        {
            MessageBox.Show("Логін чи пароль не вірний.", "Повідомлення",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
            textBox1.Text = "";
            textBox2.Text = "";
            textBox1.Focus();
        }
        conn.Close();
    }

    private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == (char)13)
        {
            textBox2.Focus();
        }
    }

    private void textBox2_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == (char)13)
        {
            button1.PerformClick();
        }
    }

    private void label4_Click(object sender, EventArgs e)
    {
        textBox1.Text = "";
        textBox2.Text = "";
        textBox1.Focus();
    }

    private void label5_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

    private void Login_Load(object sender, EventArgs e)
    {
    }
}
}
```

## Код вікна «Інформація»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
namespace WApp_NFE
{
    public partial class Info : Form
    {
        public Info()
        {
            InitializeComponent();
        }

        private void pictureBox5_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

## Код вікна «Довідник ГОЛОВНОГО МЕНЮ»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WApp_NFE
{
    public partial class Dovidnyk_Main : Form
    {
        public Dovidnyk_Main()
        {
            InitializeComponent();
        }

        private void pictureBox5_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

## Код вікна «Довідник обліку неформальної освіти»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WApp_NFE
{
    public partial class Dovidnyk_Oblik1 : Form
    {
        public Dovidnyk_Oblik1()
        {
            InitializeComponent();
        }
    }
}
```

```
private void pictureBox6_Click(object sender, EventArgs e)
{
    this.Close();
}
}
```

## Код вікна «Довідка вікна «Сертифікати»»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WFApp_NFE
{
    public partial class Dovidnyk_Sertyfikaty : Form
    {
        public Dovidnyk_Sertyfikaty()
        {
            InitializeComponent();
        }

        private void pictureBox6_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

## Схожість

Джерела з Бібліотеки

1000

1	Студентська робота	ID файлу: 1004050487	Навчальний заклад: National University Ostroh Academy	47 Джерело	6.1%
2	Студентська робота	ID файлу: 1000095837	Навчальний заклад: Lviv Polytechnic National University	38 Джерело	5.58%
3	Студентська робота	ID файлу: 1000813670	Навчальний заклад: National University of Water Manag	13 Джерело	5.58%
4	Студентська робота	ID файлу: 1008286302	Навчальний заклад: National Aviation University	4 Джерело	5.39%
5	Студентська робота	ID файлу: 1004030560	Навчальний заклад: Lviv Polytechnic National University	6 Джерело	5.28%
6	Студентська робота	ID файлу: 1004042528	Навчальний заклад: National University of Life and Environmenta...		5.28%
7	Студентська робота	ID файлу: 1000034455	Навчальний заклад: National University of Life and Envir	15 Джерело	5.2%
8	Студентська робота	ID файлу: 1003748248	Навчальний заклад: National University of Water Manag	13 Джерело	5.19%
9	Студентська робота	ID файлу: 1008284815	Навчальний заклад: Lviv Polytechnic National University	3 Джерело	5.17%
10	Студентська робота	ID файлу: 1003613981	Навчальний заклад: National University of Water Manag	20 Джерело	5.13%
11	Студентська робота	ID файлу: 1000934429	Навчальний заклад: National Aviation University	2 Джерело	4.95%
12	Студентська робота	ID файлу: 1000983780	Навчальний заклад: National Aviation University	3 Джерело	4.91%
13	Студентська робота	ID файлу: 1003958080	Навчальний заклад: National Aviation University	5 Джерело	4.86%
14	Студентська робота	ID файлу: 1004210841	Навчальний заклад: National University of Life and Environmenta...		4.84%
15	Студентська робота	ID файлу: 1003949307	Навчальний заклад: National Technical University of Ukr	3 Джерело	4.82%
16	Студентська робота	ID файлу: 1004075637	Навчальний заклад: National University Ostroh Academy		4.81%
17	Студентська робота	ID файлу: 1004049173	Навчальний заклад: National University of Life and Envir	4 Джерело	4.62%
18	Студентська робота	ID файлу: 1000991534	Навчальний заклад: National Aviation University	2 Джерело	4.49%
19	Студентська робота	ID файлу: 1003991190	Навчальний заклад: Lviv Polytechnic National University	11 Джерело	4.45%
20	Студентська робота	ID файлу: 1003991287	Навчальний заклад: Lviv Polytechnic National University	8 Джерело	3.96%

21	Студентська робота	ID файлу: 1004046833	Навчальний заклад: National Technical University of Ukraine "Kyj...	2.82%
22	Студентська робота	ID файлу: 1008233765	Навчальний заклад: Lviv Polytechnic National University	2.42%
23	Студентська робота	ID файлу: 1000759578	Навчальний заклад: National Technical University of Ukraine "Kyj...	2.21%
24	Студентська робота	ID файлу: 1008355871	Навчальний заклад: National Aviation University	6 Джерело 2.03%
25	Студентська робота	ID файлу: 1000094036	Навчальний заклад: National Technical University of Ukraine "Kyj...	1.96%
26	Студентська робота	ID файлу: 1016130155	Навчальний заклад: Lutsk National Technical University	46 Джерело 1.92%
27	Студентська робота	ID файлу: 1011345847	Навчальний заклад: National Aviation University	1.83%
28	Студентська робота	ID файлу: 1000084746	Навчальний заклад: National University Ostroh Academy	1.79%
29	Студентська робота	ID файлу: 3557942	Навчальний заклад: Lviv Polytechnic National University	1.49%
30	Студентська робота	ID файлу: 1011480448	Навчальний заклад: National University of Water Manag	21 Джерело 1.48%
31	Студентська робота	ID файлу: 102262	Навчальний заклад: Lviv Polytechnic National University	2 Джерело 1.46%
32	Студентська робота	ID файлу: 1008413111	Навчальний заклад: National University of Water Manag	3 Джерело 1.43%
33	Студентська робота	ID файлу: 1016130133	Навчальний заклад: Lutsk National Technical University	4 Джерело 1.42%
34	Студентська робота	ID файлу: 1015253905	Навчальний заклад: National University Ostroh Academy	2 Джерело 1.37%
35	Студентська робота	ID файлу: 5995139	Навчальний заклад: National University of Water Management and N...	1.36%
36	Студентська робота	ID файлу: 5951932	Навчальний заклад: National University of Life and Environ	23 Джерело 1.25%
37	Студентська робота	ID файлу: 1009120485	Навчальний заклад: National Technical University of Ukraine "Kyj...	1.25%
38	Студентська робота	ID файлу: 1015291331	Навчальний заклад: National University of Water Manag	8 Джерело 1.24%
39	Студентська робота	ID файлу: 1004070570	Навчальний заклад: Donetsk National Technical Univers	10 Джерело 1.23%
40	Студентська робота	ID файлу: 1003829849	Навчальний заклад: National University of Life and Environmenta...	1.2%
41	Студентська робота	ID файлу: 5954924	Навчальний заклад: National University Ostroh Academy	5 Джерело 1.19%
42	Студентська робота	ID файлу: 1008397333	Навчальний заклад: Lviv Polytechnic National University	4 Джерело 1.18%

43	Студентська робота	ID файлу: 1008408920	Навчальний заклад: National University of Water Manage	3 Джерело	1.07%
44	Студентська робота	ID файлу: 1000045740	Навчальний заклад: National University of Life and Environmenta...		1.07%
45	Студентська робота	ID файлу: 1015223419	Навчальний заклад: Lviv Polytechnic National University	2 Джерело	1.05%
46	Студентська робота	ID файлу: 11262217	Навчальний заклад: Lviv Polytechnic National University		1.03%
47	Студентська робота	ID файлу: 1004028936	Навчальний заклад: National Technical University of Ukraine "Kyj...		1.01%
48	Студентська робота	ID файлу: 2049778	Навчальний заклад: National University of Life and Environm	3 Джерело	0.93%
49	Студентська робота	ID файлу: 1008251817	Навчальний заклад: National Technical University of Ukraine "Kyj...		0.91%
50	Студентська робота	ID файлу: 1011414468	Навчальний заклад: National University of Water Manage	6 Джерело	0.9%
51	Студентська робота	ID файлу: 107033	Навчальний заклад: National University of Life and Environm	33 Джерело	0.9%
52	Студентська робота	ID файлу: 1011485072	Навчальний заклад: Lviv Polytechnic National University		0.86%
53	Студентська робота	ID файлу: 1008410507	Навчальний заклад: National University of Water Manage	43 Джерело	0.86%
54	Студентська робота	ID файлу: 1015304285	Навчальний заклад: National University of Water Manage	2 Джерело	0.85%
55	Студентська робота	ID файлу: 1005797294	Навчальний заклад: National Technical University of Ukr	13 Джерело	0.84%
56	Студентська робота	ID файлу: 1009689716	Навчальний заклад: National Aviation University		0.84%
57	Студентська робота	ID файлу: 1015747559	Навчальний заклад: National Technical University of Ukraine "Kyj...		0.84%
58	Студентська робота	ID файлу: 1011509165	Навчальний заклад: Uzhhorod National University	2 Джерело	0.8%
59	Студентська робота	ID файлу: 1015714771	Навчальний заклад: Vasyl Stus Donetsk National Universi	8 Джерело	0.73%
60	Студентська робота	ID файлу: 1010764690	Навчальний заклад: Yuriy Fedkovych Chernivtsi National University		0.76%
61	Студентська робота	ID файлу: 50605	Навчальний заклад: Lviv Polytechnic National University	3 Джерело	0.76%
62	Студентська робота	ID файлу: 5975478	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...		0.75%
63	Студентська робота	ID файлу: 1011490679	Навчальний заклад: National University of Life and Envir	6 Джерело	0.75%
64	Студентська робота	ID файлу: 1008329572	Навчальний заклад: National Aviation University		0.69%

65	Студентська робота	ID файлу: 1015080978	Навчальний заклад: National University of Life and Environmenta...	0.7%
66	Студентська робота	ID файлу: 1011319577	Навчальний заклад: V.I. Vernadsky Taurida National Univ <a href="#">6 Джерело</a>	0.7%
67	Студентська робота	ID файлу: 1004104911	Навчальний заклад: National Technical University of Ukra <a href="#">4 Джерело</a>	0.7%
68	Студентська робота	ID файлу: 1004036162	Навчальний заклад: National University of Life and Envir <a href="#">3 Джерело</a>	0.7%
69	Студентська робота	ID файлу: 5882689	Навчальний заклад: National Technical University of Ukraine "Kyiv Po...	0.69%
70	Студентська робота	ID файлу: 1015298928	Навчальний заклад: National University Ostroh Academy <a href="#">2 Джерело</a>	0.67%
71	Студентська робота	ID файлу: 1968622	Навчальний заклад: Lviv Polytechnic National University	0.66%
72	Студентська робота	ID файлу: 1016151242	Навчальний заклад: Yuriy Fedkovych Chernivtsi National University	0.62%
73	Студентська робота	ID файлу: 1008312576	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.64%
74	Студентська робота	ID файлу: 1015677418	Навчальний заклад: National University of Water Management an...	0.63%
75	Студентська робота	ID файлу: 1000096191	Навчальний заклад: Lviv Polytechnic National University <a href="#">4 Джерело</a>	0.6%
76	Студентська робота	ID файлу: 1016130118	Навчальний заклад: Lutsk National Technical University <a href="#">7 Джерело</a>	0.56%
77	Студентська робота	ID файлу: 1011568391	Навчальний заклад: Vasyl Stus Donetsk National University	0.55%
78	Студентська робота	ID файлу: 1008333057	Навчальний заклад: National Aviation University <a href="#">2 Джерело</a>	0.54%
79	Студентська робота	ID файлу: 1003992050	Навчальний заклад: Lviv Polytechnic National University <a href="#">12 Джерело</a>	0.53%
80	Студентська робота	ID файлу: 1005783125	Навчальний заклад: Lutsk National Technical University <a href="#">9 Джерело</a>	0.52%
81	Студентська робота	ID файлу: 1016093025	Навчальний заклад: National University of Life and Environmenta...	0.48%
82	Студентська робота	ID файлу: 5490295	Навчальний заклад: National Technical University of Ukraine <a href="#">6 Джерело</a>	0.49%
83	Студентська робота	ID файлу: 1011195645	Навчальний заклад: Taras Shevchenko National Universi <a href="#">19 Джерело</a>	0.49%
84	Студентська робота	ID файлу: 1015719414	Навчальний заклад: National University of Water Manage <a href="#">7 Джерело</a>	0.48%
85	Студентська робота	ID файлу: 47051	Навчальний заклад: Lviv Polytechnic National University <a href="#">2 Джерело</a>	0.48%
86	Студентська робота	ID файлу: 1015366374	Навчальний заклад: Vasyl Stus Donetsk National University	0.42%

87	Студентська робота	ID файлу: 1016138248	Навчальний заклад: National Aviation University	8 Джерело	0.44%
88	Студентська робота	ID файлу: 1015906236	Навчальний заклад: Taras Shevchenko National University of Kyiv		0.46%
89	Студентська робота	ID файлу: 1005797268	Навчальний заклад: National Technical University of Ukraine "Kyiv..."		0.43%
90	Студентська робота	ID файлу: 1000869986	Навчальний заклад: National Technical University of Ukraine	2 Джерело	0.43%
91	Студентська робота	ID файлу: 1013075171	Навчальний заклад: Poltava National Technical Yuri Kondratyuk U...		0.42%
92	Студентська робота	ID файлу: 1001002616	Навчальний заклад: National Aviation University		0.36%
93	Студентська робота	ID файлу: 1005797290	Навчальний заклад: National Technical University of Ukraine	2 Джерело	0.41%
94	Студентська робота	ID файлу: 1009414609	Навчальний заклад: National Technical University of Ukraine	4 Джерело	0.4%
95	Студентська робота	ID файлу: 1007499039	Навчальний заклад: Taras Shevchenko National University	2 Джерело	0.4%
96	Студентська робота	ID файлу: 113120	Навчальний заклад: Lviv Polytechnic National University		0.37%
97	Студентська робота	ID файлу: 1009704931	Навчальний заклад: National Aviation University	4 Джерело	0.37%
98	Студентська робота	ID файлу: 1016036285	Навчальний заклад: Lviv Polytechnic National University		0.37%
99	Студентська робота	ID файлу: 1008146824	Навчальний заклад: Yuriy Fedkovych Chernivtsi National	2 Джерело	0.36%
100	Студентська робота	ID файлу: 5615373	Навчальний заклад: National Technical University of Ukraine	2 Джерело	0.35%
101	Студентська робота	ID файлу: 5946816	Навчальний заклад: Lviv Polytechnic National University	2 Джерело	0.34%
102	Студентська робота	ID файлу: 1015205355	Навчальний заклад: National University Ostroh Academy		0.33%
103	Студентська робота	ID файлу: 1089500	Навчальний заклад: Lviv Polytechnic National University	6 Джерело	0.32%
104	Студентська робота	ID файлу: 1004053582	Навчальний заклад: National Technical University of Ukraine "Kyiv..."		0.32%
105	Студентська робота	ID файлу: 1011399593	Навчальний заклад: National Aviation University		0.32%
106	Студентська робота	ID файлу: 1015242982	Навчальний заклад: National University Ostroh Academy		0.32%
107	Студентська робота	ID файлу: 1015732829	Навчальний заклад: Poltava National Technical Yuri Kondratyuk U...	3 Джерело	0.31%
108	Студентська робота	ID файлу: 106457	Навчальний заклад: National University of Life and Environmental Sciences		0.31%

109	Студентська робота	ID файлу: 1011481030	Навчальний заклад: National University of Water Management an...	0.3%
110	Студентська робота	ID файлу: 1015225470	Навчальний заклад: National University Ostroh Academy <a href="#">3 Джерело</a>	0.3%
111	Студентська робота	ID файлу: 5899950	Навчальний заклад: Lviv Polytechnic National University	0.29%
112	Студентська робота	ID файлу: 5952969	Навчальний заклад: National University Ostroh Academy	0.29%
113	Студентська робота	ID файлу: 1008422749	Навчальний заклад: National University of Life and Envir <a href="#">2 Джерело</a>	0.28%
114	Студентська робота	ID файлу: 5951098	Навчальний заклад: National University Ostroh Academy	0.27%
115	Студентська робота	ID файлу: 1016130131	Навчальний заклад: Lutsk National Technical University	0.27%
116	Студентська робота	ID файлу: 1011483520	Навчальний заклад: Yuriy Fedkovych Chernivtsi National University	0.27%
117	Студентська робота	ID файлу: 1010761276	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.27%
118	Студентська робота	ID файлу: 1014423474	Навчальний заклад: Lviv Polytechnic National University	0.27%
119	Студентська робота	ID файлу: 1011385454	Навчальний заклад: National University of Water Manag <a href="#">41 Джерело</a>	0.26%
120	Студентська робота	ID файлу: 1016168758	Навчальний заклад: National University of Water Manag <a href="#">16 Джерело</a>	0.24%
121	Студентська робота	ID файлу: 1007017100	Навчальний заклад: Lviv Polytechnic National University	0.24%
122	Студентська робота	ID файлу: 1016122747	Навчальний заклад: National Aviation University <a href="#">15 Джерело</a>	0.22%
123	Студентська робота	ID файлу: 1008395256	Навчальний заклад: National University of Water Manag <a href="#">2 Джерело</a>	0.22%
124	Студентська робота	ID файлу: 1005797259	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.21%
125	Студентська робота	ID файлу: 1005797280	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.21%
126	Студентська робота	ID файлу: 1015032076	Навчальний заклад: Yuriy Fedkovych Chernivtsi National <a href="#">2 Джерело</a>	0.2%
127	Студентська робота	ID файлу: 1015770414	Навчальний заклад: National Technical University of Ukr <a href="#">8 Джерело</a>	0.19%
128	Студентська робота	ID файлу: 1015726122	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.17%
129	Студентська робота	ID файлу: 1015727045	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.17%
130	Студентська робота	ID файлу: 1011321786	Навчальний заклад: Cherkasy State Technological University	0.16%

131	Студентська робота	ID файлу: 1016419	Навчальний заклад: National University of Life and Environmental Sc...	0.15%
132	Студентська робота	ID файлу: 1006041551	Навчальний заклад: National University of Water Manag <a href="#">26 Джерело</a>	0.15%
133	Студентська робота	ID файлу: 1008373156	Навчальний заклад: National University of Water Manag <a href="#">3 Джерело</a>	0.15%
134	Студентська робота	ID файлу: 1013123630	Навчальний заклад: National Technical University of Ukr <a href="#">4 Джерело</a>	0.15%
135	Студентська робота	ID файлу: 5866910	Навчальний заклад: National University of Water Management and N...	0.14%
136	Студентська робота	ID файлу: 1008362186	Навчальний заклад: National University of Water Manag <a href="#">3 Джерело</a>	0.14%
137	Студентська робота	ID файлу: 1011379562	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.14%
138	Студентська робота	ID файлу: 1015272208	Навчальний заклад: Uzhhorod National University	0.14%
139	Студентська робота	ID файлу: 1015726136	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.14%
140	Студентська робота	ID файлу: 1000093283	Навчальний заклад: National University of Water Manag <a href="#">31 Джерело</a>	0.14%
141	Студентська робота	ID файлу: 1011333406	Навчальний заклад: Ukrainian Catholic University <a href="#">3 Джерело</a>	0.14%
142	Студентська робота	ID файлу: 1015329480	Навчальний заклад: National University of Water Management an...	0.12%
143	Студентська робота	ID файлу: 1008276330	Навчальний заклад: National Aviation University	0.09%
144	Студентська робота	ID файлу: 1006481237	Навчальний заклад: National University of Water Manag <a href="#">5 Джерело</a>	0.11%
145	Студентська робота	ID файлу: 1016121213	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.11%
146	Студентська робота	ID файлу: 1008361624	Навчальний заклад: National University of Water Manag <a href="#">2 Джерело</a>	0.11%
147	Студентська робота	ID файлу: 1016167143	Навчальний заклад: Lutsk National Technical University	0.11%
148	Студентська робота	ID файлу: 1003979269	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.1%
149	Студентська робота	ID файлу: 1003918442	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.1%
150	Студентська робота	ID файлу: 1016132014	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.1%
151	Студентська робота	ID файлу: 1016130112	Навчальний заклад: Lutsk National Technical University	0.1%
152	Студентська робота	ID файлу: 1009771053	Навчальний заклад: Ukrainian Catholic University <a href="#">8 Джерело</a>	0.1%

153	Студентська робота	ID файлу: 1015231440	Навчальний заклад: National University Ostroh Academy	0.1%
154	Студентська робота	ID файлу: 46421	Навчальний заклад: Lviv Polytechnic National University	0.1%
155	Студентська робота	ID файлу: 1011445068	Навчальний заклад: National University Ostroh Academy	0.09%
156	Студентська робота	ID файлу: 1000668400	Навчальний заклад: National University of Life and Environ <a href="#">3 Джерело</a>	0.09%
157	Студентська робота	ID файлу: 1016142778	Навчальний заклад: National University of Life and Environta...	0.06%
158	Студентська робота	ID файлу: 1004020849	Навчальний заклад: National University of Water Management an...	0.09%
159	Студентська робота	ID файлу: 1005798105	Навчальний заклад: National University of Water Management an...	0.09%
160	Студентська робота	ID файлу: 1016159363	Навчальний заклад: Lviv Polytechnic National University	0.09%
161	Студентська робота	ID файлу: 1015721444	Навчальний заклад: National Technical University of Ukr <a href="#">4 Джерело</a>	0.09%
162	Студентська робота	ID файлу: 1016147619	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.09%
163	Студентська робота	ID файлу: 1015321047	Навчальний заклад: National Technical University of Ukr <a href="#">8 Джерело</a>	0.07%
164	Студентська робота	ID файлу: 1008257042	Навчальний заклад: National University of Life and Environ <a href="#">3 Джерело</a>	0.07%
165	Студентська робота	ID файлу: 1013105265	Навчальний заклад: Vasyl Stus Donetsk National Universi <a href="#">2 Джерело</a>	0.07%
166	Студентська робота	ID файлу: 1015721264	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.07%
167	Студентська робота	ID файлу: 1004032947	Навчальний заклад: National University of Life and Environ <a href="#">4 Джерело</a>	0.06%
168	Студентська робота	ID файлу: 1014705378	Навчальний заклад: Lviv Polytechnic National University <a href="#">7 Джерело</a>	0.06%
169	Студентська робота	ID файлу: 1015561634	Навчальний заклад: Cherkasy State Technological Univer <a href="#">2 Джерело</a>	0.06%
170	Студентська робота	ID файлу: 1016162681	Навчальний заклад: National Aviation University <a href="#">6 Джерело</a>	0.06%
171	Студентська робота	ID файлу: 1016125861	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.06%
172	Студентська робота	ID файлу: 1008252409	Навчальний заклад: Ukrainian Catholic University	0.06%
173	Студентська робота	ID файлу: 1005781000	Навчальний заклад: Lviv Polytechnic National University	0.06%
174	Студентська робота	ID файлу: 1015226961	Навчальний заклад: National Technical University of Ukraine "Kyj...	0.06%

175	Студентська робота	ID файлу: 1045649	Навчальний заклад: National University of Life and Environ	7 Джерело	0.06%
176	Студентська робота	ID файлу: 1014796185	Навчальний заклад: Taras Shevchenko National Universi	15 Джерело	0.06%
177	Студентська робота	ID файлу: 1008355970	Навчальний заклад: Yuriy Fedkovych Chernivtsi Nationa	10 Джерело	0.06%
178	Студентська робота	ID файлу: 1015216286	Навчальний заклад: National Aviation University		0.06%
179	Студентська робота	ID файлу: 1016089077	Навчальний заклад: Cherkasy State Technological University		0.05%
180	Студентська робота	ID файлу: 1016165759	Навчальний заклад: Zaporizhzhia National University	30 Джерело	0.05%
181	Студентська робота	ID файлу: 1015020624	Навчальний заклад: Taras Shevchenko National University of Kyiv		0.05%
182	Студентська робота	ID файлу: 1015742676	Навчальний заклад: National Technical University of Ukraine "Kyj...		0.05%
183	Студентська робота	ID файлу: 1016133765	Навчальний заклад: Lviv Polytechnic National University	8 Джерело	0.05%
184	Студентська робота	ID файлу: 1013105260	Навчальний заклад: Vasyl Stus Donetsk National Universi	4 Джерело	0.05%
185	Студентська робота	ID файлу: 1011026715	Навчальний заклад: Yuriy Fedkovych Chernivtsi Nationa	2 Джерело	0.05%
186	Студентська робота	ID файлу: 1016143257	Навчальний заклад: National Technical University of Ukr	3 Джерело	0.05%
187	Студентська робота	ID файлу: 1015234730	Навчальний заклад: Lutsk National Technical University	4 Джерело	0.05%
188	Студентська робота	ID файлу: 1011568637	Навчальний заклад: Vasyl Stus Donetsk National University		0.05%
189	Студентська робота	ID файлу: 1016156761	Навчальний заклад: National Technical University of Ukraine "Kyj...		0.05%
190	Студентська робота	ID файлу: 1015245683	Навчальний заклад: National Technical University of Ukraine "Kyj...		0.05%
191	Студентська робота	ID файлу: 1015669775	Навчальний заклад: Lviv Polytechnic National University		0.05%
192	Студентська робота	ID файлу: 1016144232	Навчальний заклад: National Aviation University		0.05%
193	Студентська робота	ID файлу: 1011569990	Навчальний заклад: Vasyl Stus Donetsk National University		0.05%
194	Студентська робота	ID файлу: 1016137276	Навчальний заклад: National Technical University of Ukr	3 Джерело	0.05%
195	Студентська робота	ID файлу: 1016134118	Навчальний заклад: Lviv Polytechnic National University		0.05%
196	Студентська робота	ID файлу: 1015317504	Навчальний заклад: Donetsk National Technical Universi	4 Джерело	0.05%

197	Студентська робота	ID файлу: 1015282981	Навчальний заклад: Lviv Polytechnic National University	0.05%
198	Студентська робота	ID файлу: 1015692604	Навчальний заклад: Lviv Polytechnic National University <a href="#">2 Джерело</a>	0.05%
199	Студентська робота	ID файлу: 1016108029	Навчальний заклад: Lviv Polytechnic National University	0.05%
200	Студентська робота	ID файлу: 1015661269	Навчальний заклад: Lviv Polytechnic National University	0.05%
201	Студентська робота	ID файлу: 1015366367	Навчальний заклад: Vasyl Stus Donetsk National University <a href="#">4 Джерело</a>	0.05%
202	Студентська робота	ID файлу: 1005982118	Навчальний заклад: State University Kyiv National Economic University <a href="#">10 Джерело</a>	0.05%
203	Студентська робота	ID файлу: 1015597807	Навчальний заклад: Lviv Polytechnic National University <a href="#">3 Джерело</a>	0.05%
204	Студентська робота	ID файлу: 1016092082	Навчальний заклад: National University of Life and Environmental Sciences <a href="#">2 Джерело</a>	0.05%
205	Студентська робота	ID файлу: 1015649443	Навчальний заклад: National Aviation University	0.05%
206	Студентська робота	ID файлу: 1016089245	Навчальний заклад: Cherkasy State Technological University <a href="#">2 Джерело</a>	0.05%
207	Студентська робота	ID файлу: 1015291736	Навчальний заклад: Lviv Polytechnic National University	0.05%

## Цитати

Цитати

20

- 1 2.2 **написи червоним кольором позначають первинний ключ (Primary Key), написи синім кольором – зовнішній ключ (Foreign Key), написи чорним кольором – неключовий атрибут.**
- 2 Osvitni\_programy (Освітні програми) Зберігає інформацію про освітні програми, включаючи ID, назву освітньої програми, назву ННІ та освітній рівень, що допомагає структурувати навчальний процес.
- 3 Dyscypliny (Дисципліни) Містить дані про дисципліни, такі як ID, назва дисципліни, назва освітньої програми, кількість кредитів ECTS та форма контролю.
- 4 Grypy Містить дані про навчальні групи, такі як 35 (Групи) ID, назва групи, назва освітньої програми, курс та форма навчання здобувача освіти, що дозволяє організувати студентів за групами та освітніми програмами.
- 5 Zdobuvachi\_osvity (Студенти) Зберігає інформацію про студентів, включаючи номер студентського квитка, назву групи, ПІБ та email.
- 6 Sertyfikaty (Сертифікати) Містить дані про сертифікати, включаючи номер сертифікату, номер студентського квитка, ПІБ здобувача освіти, назву платформи та назву курсу MOOC, дату, кількість годин, файл та URL, що дозволяє відстежувати інформацію про підтверджені здобутки шляхом неформальної освіти.
- 7 Zarahuvannya\_neform\_osvity (Зарахування неформальної освіти) Включає інформацію про зарахування неформальної освіти, такі як ID, назва дисципліни, номер протоколу, дата протоколу та оцінка, що дозволяє відстежувати зарахування досягнень неформальної освіти.
- 8 `Fill(dataset); foreach (DataRow row in dataset.Tables[0].Rows) { comboBox1.`
- 9 `Fill(dataset); foreach (DataRow row in dataset.Tables[0].Rows) { comboBox1.`
- 10 `Fill(dataset); foreach (DataRow row in dataset.Tables[0].Rows) { comboBox1.`
- 11 `Fill(dataset); foreach (DataRow row in dataset.Tables[0].Rows) { comboBox1.`
- 12 `Fill(dataset); foreach (DataRow row in dataset.Tables[0].Rows) { comboBox1.`
- 13 `Fill(dataset2); foreach (DataRow row in dataset2.Tables[0].Rows) { comboBox2.`
- 14 `Sertyfikaty); } catch (Exception ex) { MessageBox.`
- 15 `FileName); } } } catch (Exception ex) { MessageBox.`
- 16 `Hide(); } catch (Exception ex) { MessageBox.`
- 17 `Close(); } } } catch (Exception ex) { MessageBox.`
- 18 `Fill(dataset); foreach (DataRow row in dataset.Tables[0].Rows) { comboBox1.`
- 19 `Fill(dataset2); } 99 foreach (DataRow row in dataset2.Tables[0].Rows) { comboBox2.`

```
20 ToString()); } } } private void Zarahuvannia_neform_osvity_Load(object sender, EventArgs e) { GetZarahuvanniaNeformOsvity(); foreach  
(DataGridViewColumn column in dataGridView1.Columns) { column.
```