

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

ПОГОДЖЕНО
Декан факультету

_____ **Ігор БОЛБОТ**
(підпис)

« ____ » _____ 2026 р.

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри Інформаційних
систем та технологій

_____ **Швиденко М.З.**
(підпис)

« ____ » _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: **«Розробка інформаційної системи обліку студентів та моніторингу
академічних досягнень»**

Спеціальність «122 Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні науки»

Гарант освітньої програми

д.е.н., професор

_____ **Роман РУДЕНСЬКИЙ**
(підпис)

**Керівник бакалаврської
кваліфікаційної роботи**

к.т.н., доцент

_____ **Олексій Кафтанников**
(підпис)

Виконав

_____ **Роман КАРПОВИЧ**
(підпис)

Київ-2026

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
Факультет інформаційних технологій**

ЗАТВЕРДЖУЮ
**Завідувач кафедри Інформаційних
систем та технологій**

к.т.н., доцент _____ Швиденко М.З.
(підпис)

« ____ » _____ 2026 р.

**ЗАВДАННЯ
ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧУ**

Карповичу Роману Петровичу
(прізвище, ім'я, по батькові)

Спеціальність «122 Комп'ютерні науки»
Освітньо-професійна програма «Комп'ютерні науки»

Тема бакалаврської кваліфікаційної роботи
«Розробка інформаційної системи обліку студентів та моніторингу академічних
досягнень»

затверджена наказом від «06» січня 2026 р. № 46 «С»

Термін подання завершеної роботи на кафедру «_22_» травня 2026 р.

Вихідні дані до роботи: Опис процесів обліку студентів, моніторингу академічних
досягнень та контролю навчального процесу; вимоги до інформаційної системи збору,
зберігання, обробки та візуалізації даних про студентів і результати їх навчання;
структура бази даних студентської інформації та академічних показників; технології MS
SQL Server, C#, Windows Forms та Microsoft Visual Studio 2019 для реалізації
інтерактивної інформаційної системи обліку студентів та моніторингу академічних
досягнень.

Перелік питань, що підлягають дослідженню: аналіз предметної області та постановка
задачі; проєктування системи та інформаційного забезпечення; розроблення програмного
забезпечення; тестування та впровадження системи

Перелік графічних документів (за необхідності).

Дата видачі завдання «_17_» грудня 2025 р.

**Керівник бакалаврської
кваліфікаційної роботи**

(підпис)

Олексій КАФТАННИКОВ

Завдання взяв до виконання

(підпис)

Роман КАРПОВИЧ

ЗМІСТ

ВСТУП.....	3
1 ОПИС ОБ'ЄКТУ ДОСЛІДЖЕННЯ.....	5
1.1 Аналіз предметної області.....	5
1.2 Бізнес-процеси обліку студентів та моніторингу академічних досягнень....	6
1.3 Формування вимог до автоматизації обліку студентів та моніторингу академічних досягнень.....	10
1.4 Огляд програм-аналогів.....	13
Висновки до 1 розділу.....	16
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ СТУДЕНТІВ ТА МОНІТОРИНГУ АКАДЕМІЧНИХ ДОСЯГНЕНЬ.....	17
2.1 Розробка архітектури інформаційної системи.....	17
2.2 Проектування бази даних.....	18
2.3 Вибір засобів розробки інформаційної системи.....	22
2.4 Визначення форм вхідних та вихідних даних.....	25
2.5 Навігація та взаємодія з користувачем.....	27
Висновки до 2 розділу.....	28
3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	30
3.1 Створення бази даних у MS SQL Server.....	30
3.2 Реалізація інтерфейсу інформаційної системи.....	32
3.3 Реалізація інтерфейсу адміністрування та розмежування прав доступу.....	39
3.4 Тестування розробленої інформаційної системи.....	40
Висновки до 3 розділу.....	42
Висновки.....	43
Перелік джерел посилань.....	44
ДОДАТОК А Скрипти створення бази даних та її таблиць.....	46

ВСТУП

У сучасних умовах розвитку інформаційного суспільства інформатизація освітніх процесів є одним із ключових напрямів підвищення ефективності діяльності закладів вищої освіти. Значні обсяги даних про студентів, результати їх навчання, відвідуваність та інші показники академічної діяльності потребують систематизації, швидкої обробки та зручного доступу для різних категорій користувачів.

Традиційні методи ведення обліку студентів, які базуються на паперовій документації або розрізнених електронних таблицях, мають низку недоліків: високу ймовірність помилок, складність актуалізації даних, обмежені можливості аналітики та формування звітності. Це ускладнює контроль за навчальним процесом, своєчасне виявлення проблем з успішністю студентів і прийняття обґрунтованих управлінських рішень.

У зв'язку з цим актуальною є розробка інформаційної системи обліку студентів та моніторингу академічних досягнень, яка забезпечить централізоване зберігання даних, автоматизацію процесів їх обробки, а також надасть інструменти для аналізу та оцінювання результатів навчання.

Метою роботи є розробка програмного забезпечення, яке дозволить автоматизувати облік студентів, ведення академічної успішності та здійснювати моніторинг навчальних досягнень.

Об'єктом дослідження є процеси обліку студентів та контролю їх академічної успішності у закладах освіти. Предметом дослідження виступають методи та засоби автоматизації цих процесів за допомогою інформаційних технологій.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз предметної області;
- визначити функціональні вимоги до системи;
- спроектувати структуру бази даних;
- розробити архітектуру інформаційної системи;
- реалізувати програмний продукт;

- провести тестування та оцінку ефективності системи.

Практичне значення роботи полягає у можливості впровадження розробленої системи у навчальний процес для підвищення якості управління освітньою діяльністю, зменшення трудомісткості обробки даних та забезпечення прозорості оцінювання результатів навчання студентів.

1 ОПИС ОБ'ЄКТУ ДОСЛІДЖЕННЯ

1.1 Аналіз предметної області

Предметна область розробки інформаційної системи обліку студентів та моніторингу академічних досягнень охоплює комплекс процесів, пов'язаних із організацією, веденням та контролем освітньої діяльності у закладах вищої освіти. У центрі цієї області знаходиться інформація про студентів, їх навчальний шлях, результати оцінювання знань, а також взаємодія між усіма учасниками освітнього процесу.

У сучасних умовах функціонування освітніх установ обсяг даних постійно зростає. Кожен студент має значну кількість пов'язаних характеристик: персональні дані, належність до певної академічної групи, спеціальності та курсу, історію навчання та результати складання дисциплін. Окрім цього, формується великий масив інформації про викладачів, навчальні дисципліни, навчальні плани, графіки навчального процесу, а також результати поточного і підсумкового контролю знань. Усе це створює складну інформаційну структуру, яка потребує впорядкування та ефективного управління.

Традиційні підходи до ведення обліку, які часто базуються на використанні паперових журналів або розрізнених електронних таблиць, вже не відповідають сучасним вимогам. Вони характеризуються низькою швидкістю обробки інформації, обмеженими можливостями пошуку та аналізу даних, а також високою ймовірністю виникнення помилок через людський фактор. Відсутність централізованої системи ускладнює доступ до актуальної інформації, призводить до дублювання даних та створює труднощі при формуванні звітності.

Особливе значення в межах даної предметної області має процес моніторингу академічних досягнень студентів. Його сутність полягає у постійному відстеженні результатів навчання, аналізі динаміки змін успішності та виявленні тенденцій у навчальному процесі. Такий підхід дозволяє своєчасно виявляти студентів із низьким рівнем знань, визначати проблемні дисципліни або етапи навчання, а

також оцінювати ефективність викладання. Моніторинг є важливим інструментом для прийняття управлінських рішень, спрямованих на підвищення якості освіти.

Інформаційна система, що розробляється, повинна враховувати потреби різних категорій користувачів. Викладачі зацікавлені у зручному внесенні та коригуванні оцінок, швидкому доступі до інформації про студентів та формуванні звітів по дисциплінах. Студенти потребують можливості оперативно отримувати інформацію про власні академічні результати, що сприяє підвищенню їх мотивації та відповідальності за навчання. Адміністрація навчального закладу, у свою чергу, використовує узагальнені дані для аналізу діяльності підрозділів, контролю якості освітнього процесу та планування подальшого розвитку.

Таким чином, аналіз предметної області свідчить про необхідність створення єдиної інформаційної системи, яка забезпечить централізоване зберігання даних, автоматизацію основних процесів обліку та ефективні інструменти для моніторингу академічних досягнень. Впровадження такої системи дозволить підвищити оперативність роботи з інформацією, зменшити кількість помилок, забезпечити прозорість оцінювання та покращити якість управління освітнім процесом загалом.

1.2 Бізнес-процеси обліку студентів та моніторингу академічних досягнень

Бізнес-процеси обліку студентів та моніторингу їх академічних досягнень є ключовими елементами управління освітнім процесом у закладах вищої освіти. Вони забезпечують формування єдиного інформаційного простору, необхідного для ефективної організації навчального процесу, контролю результатів навчання та прийняття управлінських рішень.

Облік студентів являє собою сукупність організаційних та інформаційних процедур, спрямованих на реєстрацію, зберігання та актуалізацію даних про студентів протягом усього періоду їх навчання.

На етапі зарахування здійснюється первинне формування інформації про

студента, що включає персональні дані, відомості про освітню програму, форму навчання, групу та інші характеристики. Далі відбувається створення електронної особової справи, яка є основним джерелом інформації про студента.

У процесі навчання система обліку забезпечує:

- ведення історії змін статусу студента (переведення, академічні відпустки, відрахування);
- актуалізацію даних про навчальні групи та спеціальності;
- зберігання інформації про індивідуальні освітні траєкторії;
- облік контингенту студентів у розрізі факультетів та кафедр.

Автоматизація даного процесу дозволяє значно зменшити трудомісткість обробки інформації, підвищити її достовірність та забезпечити швидкий доступ до необхідних даних.

Моніторинг академічних досягнень є безперервним процесом збору, обробки та аналізу інформації про результати навчальної діяльності студентів. Його основною метою є оцінка рівня засвоєння навчального матеріалу та виявлення тенденцій у навчанні.

Даний процес включає:

- реєстрацію результатів поточного контролю (модульні роботи, тести, практичні завдання);
- фіксацію результатів підсумкового контролю (заліки, іспити);
- розрахунок середнього балу та рейтингових показників;
- формування відомостей успішності;
- аналіз динаміки змін навчальних результатів;
- виявлення студентів із ризиком академічної неуспішності.

Таблиця 1.1 – Входи та виходи бізнес-процесів обліку студентів та моніторингу академічних досягнень

Бізнес-процес	Входи	Виходи	Відповідальні
Облік студентів	Заяви абітурієнтів; персональні дані; накази про зарахування	Електронні особові справи студентів; база даних контингенту; списки груп	Деканат, приймальна комісія, ІТ- відділ

Кінець таблиці 1.1

Бізнес-процес	Входи	Виходи	Відповідальні
Актуалізація даних студентів	Накази про переведення, відрахування, академічні відпустки	Оновлені дані в інформаційній системі; актуальний статус студентів	Деканат, відділ кадрів
Облік навчального процесу	Навчальні плани; розклад занять; перелік дисциплін	Структурована інформація про навчальний процес у системі	Навчальний відділ, кафедри
Введення результатів навчання	Результати контрольних заходів (тести, заліки, іспити)	База даних оцінок; відомості успішності	Викладачі, кафедри
Моніторинг успішності	Дані про оцінки студентів; навчальні плани	Аналітичні звіти; рейтинги студентів; середній бал	Деканат, адміністрація
Аналіз академічних досягнень	Історія успішності; статистичні дані	Виявлення проблемних студентів; рекомендації щодо покращення навчання	Адміністрація, куратори груп
Формування звітності	Дані з інформаційної системи	Звіти для керівництва, державної статистики, внутрішнього контролю	Деканат, навчальний відділ

У межах розглянутих бізнес-процесів відбувається активний обмін інформацією між різними учасниками освітнього процесу (рисунк 1.1):

- адміністрацією закладу освіти;
- деканатами та кафедрами;
- викладачами;
- студентами.

Викладачі здійснюють внесення даних про успішність, деканати контролюють повноту та коректність інформації, а адміністрація використовує узагальнені дані для аналізу та прийняття рішень. Студенти, у свою чергу, отримують доступ до власних результатів навчання.

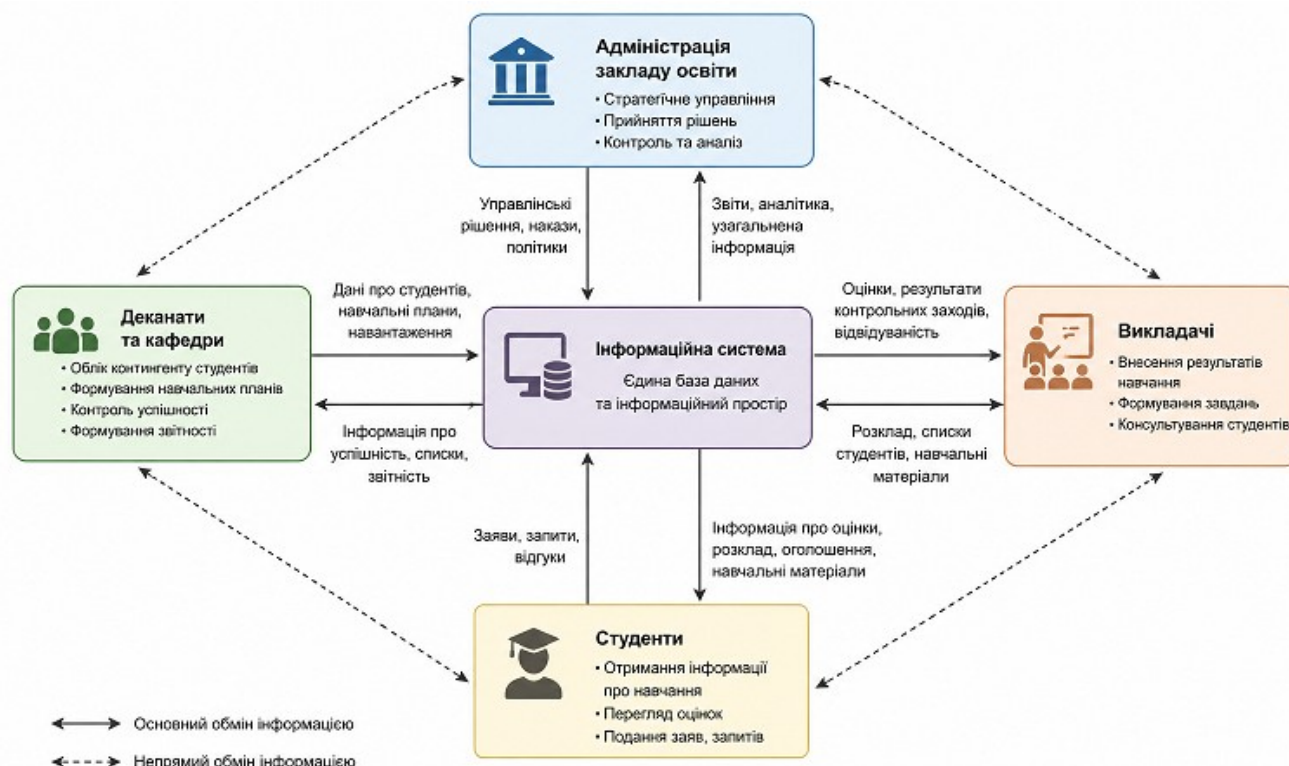


Рисунок 1.1 – Схема обміну інформацією між учасниками освітнього процесу

Процеси обліку студентів та моніторингу академічних досягнень є тісно інтегрованими. Дані про студентів, що формуються в процесі обліку, використовуються для ведення успішності, формування звітності та проведення аналітики.

Інтеграція цих процесів в єдину інформаційну систему забезпечує:

- централізоване зберігання інформації;
- узгодженість та актуальність даних;
- підвищення ефективності управління навчальним процесом;
- можливість оперативного реагування на проблеми в успішності студентів.

Таким чином, автоматизація бізнес-процесів обліку студентів та моніторингу їх академічних досягнень є важливою умовою підвищення якості освітніх послуг та ефективності функціонування закладу освіти.

Інформаційна система дозволить автоматизувати розрахунки, мінімізувати вплив людського фактору та забезпечить прозорість оцінювання. Крім того, вона надасть можливість формувати різноманітні аналітичні звіти для викладачів,

деканатів та адміністрації.

1.3 Формування вимог до автоматизації обліку студентів та моніторингу академічних досягнень

Формування вимог до автоматизації обліку студентів та моніторингу їх академічних досягнень є важливим етапом розробки та впровадження інформаційної системи. Від повноти та обґрунтованості вимог залежить ефективність функціонування системи та її відповідність потребам користувачів [1].

На першому етапі здійснюється визначення потреб основних користувачів системи, до яких належать:

- адміністрація закладу освіти;
- деканати та кафедри;
- викладачі;
- студенти.

Кожна група користувачів має власні вимоги до функціональності системи. Наприклад, адміністрація зацікавлена в отриманні аналітичної інформації, деканати – в оперативному веденні обліку контингенту, викладачі – у зручному введенні результатів навчання, а студенти – у доступі до власних даних.

У процесі формування вимог до автоматизації обліку студентів та моніторингу академічних досягнень необхідно визначити ключові функціональні та нефункціональні характеристики майбутньої інформаційної системи. Це дозволяє забезпечити ефективне управління навчальним процесом, підвищити точність обліку даних про студентів, а також спростити аналіз їхніх навчальних результатів.

Нижче наведено узагальнені вимоги до автоматизованої системи, які враховують потреби користувачів, особливості предметної області та принципи побудови сучасних інформаційних систем (таблиця 1.2).

Для забезпечення коректної роботи програмного додатку необхідно

відповідне технічне та програмне забезпечення клієнтської та серверної частини системи [2].

Таблиця 1.2 – Вимоги до автоматизації обліку студентів та моніторингу академічних досягнень

Тип вимог	Опис вимоги
Функціональна	Ведення електронного обліку студентів (створення, редагування, видалення записів)
	Формування та зберігання електронних особових справ студентів
	Облік навчальних груп, спеціальностей та форм навчання
	Введення результатів поточного та підсумкового контролю
	Автоматичний розрахунок середнього балу та рейтингу студентів
	Формування відомостей успішності та аналітичних звітів
	Пошук, фільтрація та сортування даних
Нефункціональна	Забезпечення надійності та безперервної роботи системи
	Захист персональних даних та розмежування доступу
	Висока продуктивність при обробці великих обсягів даних
	Зручний та інтуїтивно зрозумілий інтерфейс
	Можливість масштабування системи
Інформаційна	Централізоване зберігання даних
	Забезпечення актуальності та достовірності інформації
	Ведення історії змін даних
	Підтримка формування звітності
Організаційна	Розподіл ролей і прав доступу користувачів
	Регламентація роботи в системі
	Організація навчання користувачів
	Забезпечення технічної підтримки
Інтерфейсна	Забезпечення зручної навігації в системі
	Швидкий доступ до основних функцій
	Адаптивність інтерфейсу для різних пристроїв

Для роботи клієнтського додатку рекомендується персональний комп'ютер із такими мінімальними характеристиками:

- Процесор: не нижче Intel Core i3 або аналогічний за продуктивністю.
- Оперативна пам'ять: не менше 8 ГБ.
- Дискова система: SSD об'ємом не менше 256 ГБ.
- Монітор з роздільною здатністю не нижче 1366×768.

Рекомендована конфігурація (для комфортної роботи при великій кількості

даних):

- Процесор: Intel Core i5 або вище.
- Оперативна пам'ять: 16 ГБ.
- SSD об'ємом 512 ГБ або більше.

Вимоги до програмного забезпечення клієнтського ПК:

- Операційна система: Microsoft Windows 10.
- Платформа .NET (версія не нижче .NET 6 / .NET Framework 4.8).
- Середовище розробки (для підтримки та модифікації): Microsoft Visual

Studio 2019 або новіше.

- Драйвери доступу до бази даних (SQL Server Native Client).

У разі використання окремого сервера бази даних рекомендовані такі характеристики:

Процесор: не нижче 4-ядерного з частотою від 2.5 ГГц.

- Оперативна пам'ять: не менше 16 ГБ.
- Дискова підсистема: SSD від 512 ГБ з можливістю резервного копіювання.

Вимоги до серверного програмного забезпечення:

- Операційна система: Windows Server 2019 або новіше.
- Система керування базами даних: Microsoft SQL Server 2019 або новіше.
- Налаштовані механізми резервного копіювання бази даних.
- Засоби захисту доступу (аутентифікація користувачів, розмежування прав доступу).

Дотримання зазначених вимог забезпечить стабільну роботу програмного продукту, швидкий доступ до бази даних, надійне зберігання інформації та безпечну взаємодію користувачів із системою.

Проведена систематизація вимог до автоматизації обліку студентів та моніторингу академічних досягнень дозволяє визначити ключові напрями розробки та впровадження інформаційної системи. Представлені у таблиці вимоги охоплюють як функціональні можливості системи, так і її якісні характеристики та

технічні вимоги, що забезпечують ефективність та надійність її використання.

Комплексний підхід до формування вимог сприяє створенню інформаційної системи, яка відповідає потребам усіх учасників освітнього процесу та забезпечує підтримку основних бізнес-процесів закладу освіти. Це, у свою чергу, дозволяє підвищити рівень автоматизації, якість обробки даних та ефективність управління навчальною діяльністю.

Таким чином, сформовані вимоги є основою для подальшого проектування, реалізації та впровадження інформаційної системи обліку студентів і моніторингу їх академічних досягнень.

1.4 Огляд програм-аналогів

Для обґрунтування доцільності розробки власної інформаційної системи обліку студентів та моніторингу академічних досягнень необхідно провести аналіз існуючих програмних рішень, які використовуються у сфері освіти. Сучасний ринок програмного забезпечення пропонує широкий спектр систем, що частково або повністю реалізують функції обліку студентів, управління навчальним процесом та контролю успішності.

Одним із найбільш поширених класів таких рішень є системи управління навчанням (LMS), які забезпечують підтримку освітнього процесу в електронному середовищі. До них належать такі системи, як Moodle, Google Classroom та інші. Вони надають можливості для розміщення навчальних матеріалів, проведення тестування, оцінювання знань та комунікації між викладачами і студентами. Проте їх основний акцент зроблений на організацію навчального процесу, а не на повноцінний облік студентів та аналітику академічних показників.

Існують також спеціалізовані інформаційні системи, що використовуються безпосередньо у закладах освіти для ведення обліку студентів та формування звітності. Такі системи зазвичай інтегрують у собі модулі для роботи з контингентом студентів, розкладом занять, успішністю та адміністративними процесами. Водночас вони часто є складними у впровадженні, потребують значних

фінансових витрат та мають обмежену гнучкість щодо адаптації під конкретні потреби навчального закладу.

Крім того, на практиці нерідко використовуються універсальні інструменти, такі як електронні таблиці або локальні бази даних. Вони дозволяють реалізувати базові функції обліку, проте не забезпечують належного рівня автоматизації, безпеки та масштабованості. Відсутність централізованого доступу та механізмів контролю призводить до дублювання даних і ускладнює процес аналізу інформації.

Для організації навчального процесу в сучасних закладах освіти використовуються різноманітні платформи дистанційного навчання, які забезпечують взаємодію між викладачами та студентами, контроль знань і управління навчальними матеріалами.

До таких систем належать Moodle, Google Classroom та Quizizz, які мають різні підходи до організації навчання, але спільну мету — підвищення ефективності освітнього процесу.

Система Moodle [3] є однією з найпоширеніших платформ дистанційного навчання з відкритим кодом. Вона забезпечує широкі можливості для створення курсів, організації навчального процесу, проведення тестування та оцінювання студентів. Moodle активно використовується у закладах освіти завдяки гнучкості та можливості налаштування під конкретні потреби.

Платформа Google Classroom [4] призначена для організації навчання в онлайн-середовищі та інтегрується з іншими сервісами Google. Вона дозволяє викладачам створювати завдання, обмінюватися матеріалами зі студентами та контролювати виконання робіт. Система відзначається простотою використання та швидкістю впровадження.

Сервіс Quizizz [5] орієнтований на проведення інтерактивного тестування та перевірку знань у формі вікторин. Він підтримує ігрові елементи, що підвищують зацікавленість студентів, та надає можливість аналізу результатів у реальному часі. Quizizz часто використовується як додатковий інструмент для контролю знань.

Наведемо порівняльну характеристику розглянутих систем (таблиця 1.3).

Таблиця 1.3 – Порівняння програм-аналогів

Характеристика	Moodle	Google Classroom	Quizizz	Власна розробка
Облік студентів	Частково	Ні	Ні	Так
Облік успішності	Так	Частково	Частково	Так
Аналітика	Частково	Ні	Статистика результатів	Так
Гнучкість	Висока	Середня	Середня	Висока
Простота використання	Середня	Висока	Низька	Висока
Доступність	Встановлення на сервер, доступ через браузер	Онлайн (через браузер)	Онлайн (через браузер)	Десктопний додаток

Отже, проведений огляд програм-аналогів показав, що існуючі програмні рішення не повною мірою задовольняють вимоги до системи обліку студентів та моніторингу їх академічних досягнень. Зокрема, виявлено обмеження у функціональних можливостях, недостатню гнучкість налаштування під специфіку закладу освіти, а також складність інтеграції з іншими інформаційними системами.

Крім того, значна частина розглянутих систем не забезпечує повного охоплення бізнес-процесів, пов'язаних із веденням обліку студентів та аналізом їх навчальних результатів, що ускладнює процес прийняття управлінських рішень. Недостатній рівень автоматизації окремих операцій призводить до підвищення трудомісткості роботи персоналу та збільшення ймовірності виникнення помилок.

Враховуючи виявлені недоліки, можна зробити висновок про актуальність та доцільність розробки власного програмного продукту, який буде орієнтований на потреби конкретного закладу освіти. Така система повинна забезпечувати комплексну автоматизацію обліку студентів, ефективний моніторинг академічних досягнень, зручний інтерфейс користувача, а також можливість подальшого розширення та інтеграції.

Розробка та впровадження власної інформаційної системи дозволить підвищити якість обробки інформації, забезпечити її достовірність і актуальність,

оптимізувати роботу персоналу та підвищити ефективність управління освітнім процесом загалом.

Таким чином, результати проведеного аналізу обґрунтовують необхідність створення нової інформаційної системи, яка відповідатиме сучасним вимогам автоматизації та сприятиме підвищенню якості освітніх послуг.

Висновки до 1 розділу

У першому розділі проаналізовано предметну область автоматизації обліку студентів та моніторингу академічних досягнень. Визначено основні бізнес-процеси, що супроводжують навчальний процес, та сформульовано вимоги до інформаційної системи.

Огляд існуючих програм-аналогів показав, що хоча на ринку присутні потужні рішення, вони не завжди повністю відповідають потребам окремих навчальних закладів, що підтверджує актуальність розробки спеціалізованої системи.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ СТУДЕНТІВ ТА МОНІТОРИНГУ АКАДЕМІЧНИХ ДОСЯГНЕНЬ

2.1 Розробка архітектури інформаційної системи

У процесі виконання роботи буде здійснено проєктування архітектури інформаційної системи обліку студентів та моніторингу їхніх академічних досягнень. Архітектура системи буде побудована з урахуванням сучасних підходів до розробки програмного забезпечення та забезпечення ефективної взаємодії між її складовими.

Передбачається, що система буде реалізована за клієнт-серверною архітектурою, що дозволить розділити логіку обробки даних і представлення інформації користувачу. Такий підхід забезпечить гнучкість, масштабованість та зручність подальшого розвитку системи (рисунк 2.1) [6].

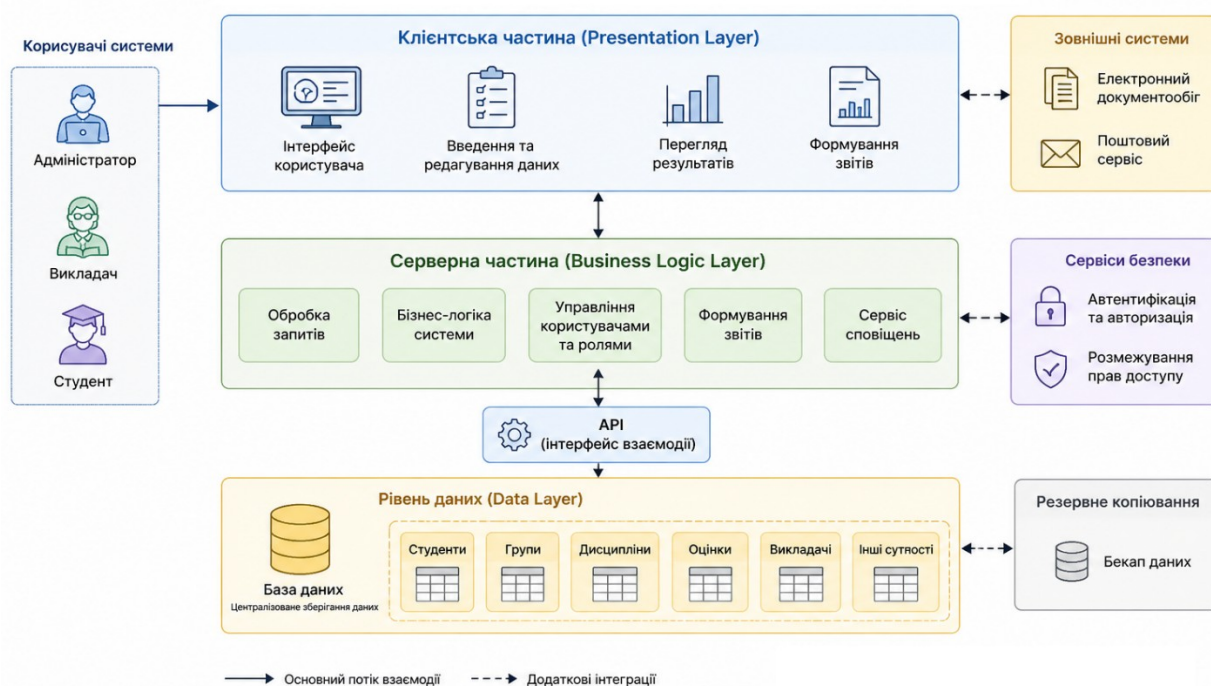


Рисунок 2.1 – Архітектура інформаційної системи

Серверна частина системи буде відповідати за обробку запитів, виконання бізнес-логіки, зберігання та управління даними. Вона забезпечуватиме централізований доступ до інформації про студентів, їхні академічні результати,

дисципліни, викладачів та інші сутності предметної області.

Клієнтська частина буде реалізовувати інтерфейс взаємодії з користувачем. Вона забезпечуватиме введення, перегляд та редагування даних, а також відображення результатів у зручному та наочному вигляді. Особлива увага буде приділена зручності користування, інтуїтивності інтерфейсу та швидкості виконання основних операцій.

У межах архітектури буде передбачено використання прикладного програмного інтерфейсу (API), який забезпечить обмін даними між клієнтською та серверною частинами. Це дозволить відокремити фронтенд від бекенду та спростить інтеграцію з іншими системами в майбутньому.

Архітектура системи буде модульною, що дозволить розділити функціонал на окремі компоненти відповідно до бізнес-завдань. Це спростить супровід системи, її тестування та подальше масштабування. Окрім цього, буде забезпечено можливість розширення функціоналу без суттєвого впливу на вже існуючі модулі.

Також буде враховано необхідність забезпечення безпеки даних, зокрема реалізацію механізмів автентифікації та авторизації користувачів, розмежування прав доступу та захисту інформації від несанкціонованого доступу.

У результаті буде сформовано архітектурне рішення, яке забезпечить надійну, масштабовану та зручну у використанні інформаційну систему для обліку студентів і моніторингу їхніх академічних досягнень.

2.2 Проектування бази даних

Виконаємо проектування бази даних інформаційної системи обліку студентів та моніторингу академічних досягнень. Основною метою є створення структурованого, надійного та ефективного сховища даних, що забезпечує цілісність, узгодженість та швидкий доступ до інформації [7].

Інформаційна система обліку студентів та моніторингу академічної успішності базується на наборі взаємопов'язаних сутностей, які відображають ключові об'єкти навчального процесу. Вони забезпечують зберігання, обробку та

аналіз даних щодо студентів, навчальних груп, викладачів, дисциплін, оцінок та відвідуваності. Узгоджені зв'язки між сутностями гарантують цілісність інформації та дозволяють формувати аналітичні показники успішності (таблиця 2.1).

Таблиця 2.1 – Опис сутностей інформаційної системи

Сутність	Опис
Студент	Основна сутність, що містить інформацію про студентів
Група	Навчальні групи студентів
Спеціальність	Напрямок підготовки студентів
Кафедра	Структурний підрозділ, що об'єднує викладачів
Викладач	Викладачі навчального закладу
Дисципліна	Навчальні предмети
Дисципліни викладача	Зв'язок між викладачем і дисциплінами
Оцінка	Результати навчання студентів

Зв'язки між сутностями інформаційної системи відображають логічну структуру навчального процесу та визначають правила взаємодії між основними об'єктами предметної області. Вони забезпечують цілісність даних, коректність їх зберігання та можливість узгодженого доступу до інформації [8]. Типи зв'язків (один-до-одного, один-до-багатьох) дозволяють коректно моделювати реальні процеси навчального закладу, зокрема розподіл студентів по групах і спеціальностях, закріплення викладачів за кафедрами та дисциплінами, а також фіксацію результатів навчання студентів (таблиця 2.2).

Таблиця 2.2 – Опис зв'язків між сутностями

Зв'язок	Тип зв'язку	Опис
Група – Студент	Один-до-багатьох	Одна група містить багато студентів
Спеціальність – Група	Один-до-багатьох	Одна спеціальність включає багато груп
Кафедра – Викладач	Один-до-багатьох	Одна кафедра має багато викладачів
Викладач – Дисципліни викладача	Один-до-багатьох	Один викладач має багато записів у проміжній таблиці
Дисципліна – Дисципліни	Один-до-багатьох	Одна дисципліна може бути призначена багатьом викладачам

Зв'язок	Тип зв'язку	Опис
викладача		
Студент – Оцінка	Один-до-багатьох	Один студент має багато оцінок
Дисципліна викладача – Оцінка	Один-до-багатьох	Один студент має оцінки з кількох дисциплін

На основі визначених сутностей формується модель даних у вигляді ER-діаграми, що відображає основні об'єкти та зв'язки між ними. ER-діаграма наведена на рисунку 2.2.



Рисунок 2.2 – ER-діаграма предметної області

Після побудови ER-діаграми виконується логічне проектування бази даних. Кожна сутність доповнюється набором атрибутів, визначаються первинні ключі та встановлюються зовнішні ключі для забезпечення зв'язків між таблицями (таблиця 2.3).

Таблиця 2.3 – Логічна структура бази даних

Сутність	Атрибути	Ключі
Студент	ПІБ, дата народження, стать, email, телефон, група	первинний ключ: студент_id; зовнішній ключ: група_id
Група	Назва групи, курс, спеціальність	первинний ключ: груп_id; зовнішній ключ: спеціальність_id
Спеціальність	Назва спеціальності	первинний ключ: спеціальність_id

Сутність	Атрибути	Ключі
Кафедра	Назва кафедри	первинний ключ: кафедра_id
Викладач	ПІБ, email, кафедра	первинний ключ: викладвч_id; зовнішній ключ: кафедра_id
Дисципліна	Назва дисципліни, кількість кредитів	первинний ключ: дисципліна_id

Кінець таблиці 2.3

Сутність	Атрибути	Ключі
Дисципліни викладача	викладач, дисципліна	первинний ключ: id; зовнішні ключі: викладвч_id, дисципліна_id
Оцінка	студент, дисципліна викладача, оцінка, семестр, дата	первинний ключ: оцінка_id; зовнішні ключі: студент_id, id

Логічна модель бази даних наведена на рисунку 2.3.

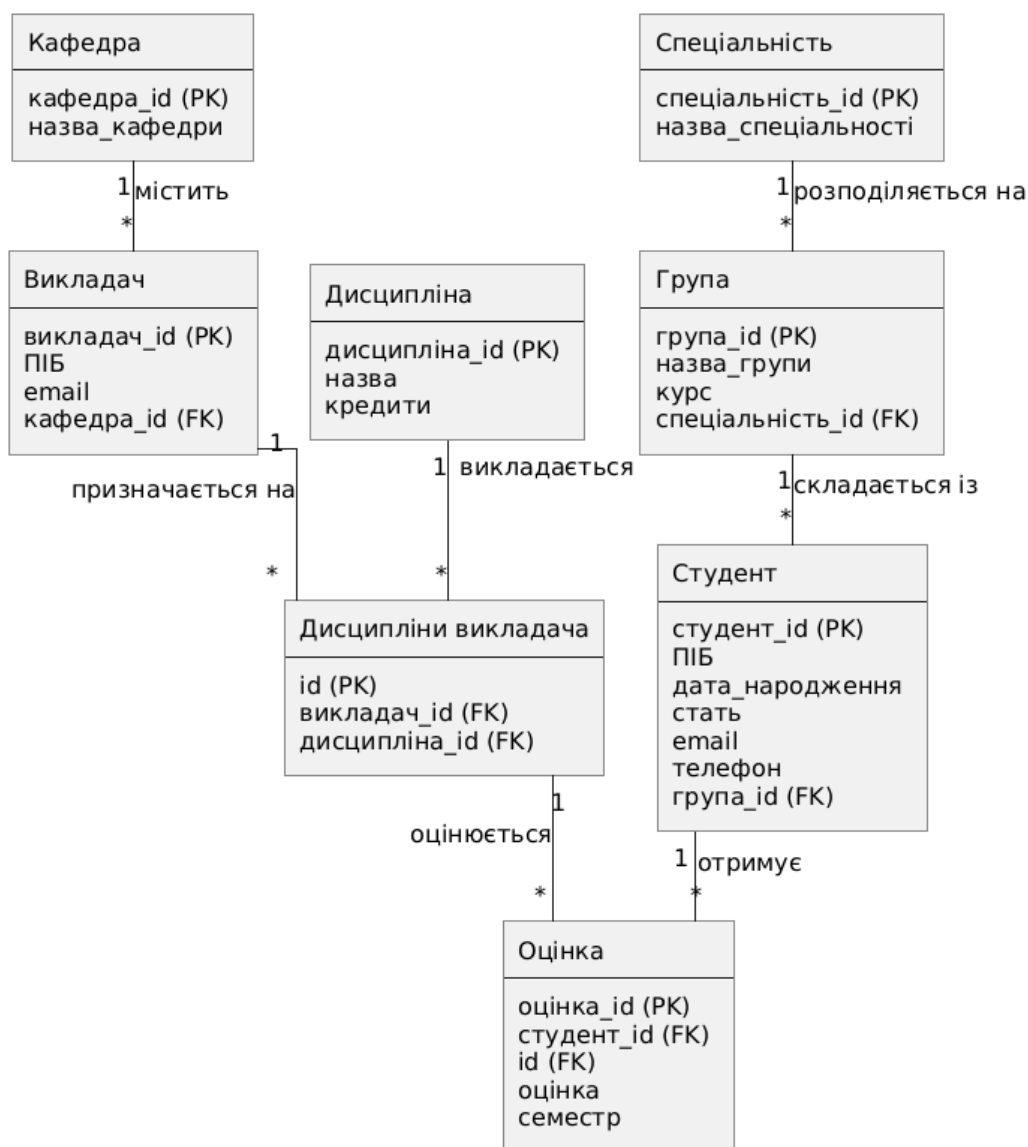


Рисунок 2.3 – Логічна модель даних

У процесі проектування проводиться нормалізація бази даних з метою усунення надлишковості та забезпечення цілісності даних. Структура бази даних відповідає третій нормальній формі, що дозволяє уникнути аномалій при додаванні, оновленні та видаленні даних [9].

Також визначаються обмеження цілісності:

- унікальність первинних ключів,
- обов’язковість заповнення критичних полів,
- коректність значень атрибутів,
- узгодженість зовнішніх ключів.

У результаті проектування отримано структуровану та масштабовану модель даних, яка забезпечить створення бази даних інформаційної системи.

2.3 Вибір засобів розробки інформаційної системи

Для реалізації інформаційної системи обліку студентів та моніторингу навчального процесу потрібно обрати сучасний та поширений технологічний стек, який забезпечить надійність, продуктивність та зручність розробки.

Для зберігання, обробки та управління даними в інформаційних системах використовуються системи управління базами даних, які забезпечують надійність, цілісність та швидкий доступ до інформації.

Найпоширенішими серед них є MS SQL Server, MySQL та PostgreSQL, кожна з яких має свої особливості та сфери застосування.

MS SQL Server [10] — це реляційна система управління базами даних, розроблена компанією Microsoft. Вона забезпечує високу продуктивність, надійність та розширені можливості для роботи з великими обсягами даних. MS SQL Server широко використовується в корпоративних інформаційних системах завдяки інтеграції з іншими продуктами Microsoft та підтримці складних SQL-запитів.

MySQL [11] — це популярна відкрита система управління базами даних, яка

часто використовується у веб-розробці. Вона характеризується простотою використання, високою швидкістю роботи та підтримкою основних операцій з реляційними даними. MySQL є одним із найпоширеніших рішень для невеликих і середніх проєктів.

PostgreSQL [12] — це потужна об'єктно-реляційна система управління базами даних з відкритим кодом. Вона підтримує розширені можливості SQL, складні типи даних та високу надійність збереження інформації. PostgreSQL часто використовується у складних аналітичних та корпоративних системах, де важлива масштабованість і точність обробки даних.

Порівняння СУБД для розробки інформаційної системи подано в таблиці 2.4.

Таблиця 2.4 – Порівняння СУБД для розробки інформаційної системи

Критерій	MS SQL Server [10]	MySQL [11]	PostgreSQL [12]
Ліцензія	Платна (є Express)	Безкоштовна / комерційна	Безкоштовна (open-source)
Інтеграція з C# / .NET	Дуже висока	Середня	Висока
Інтеграція з Visual Studio	Нативна (вбудовані інструменти)	Обмежена	Через сторонні драйвери
Продуктивність	Висока (OLTP, enterprise)	Висока для web	Висока для складних запитів
Простота використання	Висока	Висока	Середня
Масштабованість	Висока	Середня	Висока
Адміністрування	Зручне (SSMS)	Просте	Середнє

Хоча PostgreSQL та MySQL є популярними безкоштовними рішеннями, саме MS SQL Server є найбільш доцільним вибором для даної інформаційної системи. Це зумовлено його тісною інтеграцією з технологіями Microsoft та високою продуктивністю при роботі з навчальними базами даних.

Для розробки сучасних інформаційних систем використовуються різні інтегровані середовища розробки (IDE) та мови програмування, які забезпечують створення, налагодження та підтримку програмного забезпечення. Вибір інструментів розробки залежить від вимог проєкту, продуктивності та зручності

роботи з кодом.

Visual Studio + C# (.NET) [13] — це потужне інтегроване середовище розробки від Microsoft, яке використовується для створення настільних, веб- та корпоративних застосунків. Платформа .NET та мова C# забезпечують високу продуктивність, підтримку об'єктно-орієнтованого програмування та тісну інтеграцію з базами даних і Windows-середовищем.

IntelliJ IDEA + Java [14] — це професійне середовище розробки для мови Java, яке забезпечує зручну роботу з великими проєктами, автоматизацію кодування та потужні інструменти для налагодження. Java широко використовується для створення кросплатформних корпоративних систем.

PyCharm + Python [15] — це спеціалізоване IDE для мови Python, яке підтримує швидку розробку застосунків, аналіз даних та автоматизацію процесів. Python відзначається простим синтаксисом і широко використовується в наукових, веб- та аналітичних системах.

Порівняння середовищ та технологій розробки інформаційних систем подано у таблиці 2.5.

Таблиця 2.5 – Порівняння середовищ та технологій розробки інформаційних систем

Критерій	Visual Studio + C# (.NET) [13]	IntelliJ IDEA + Java [14]	PyCharm + Python [15]
Тип платформи	Windows / .NET (кросплатформенний .NET Core)	Кросплатформенна JVM	Кросплатформенна
Простота навчання	Висока	Середня	Дуже висока
Продуктивність	Висока	Висока	Середня
Типізація	Строга (C#)	Строга (Java)	Динамічна
Підтримка БД	Нативна інтеграція з MSSQL	Добра	Через бібліотеки
Інструменти IDE	Дуже розвинені (Visual Studio)	Розвинені	Добрі
Підтримка GUI	WinForms / WPF / MAUI	JavaFX	Обмежена
Підходить для систем типу «деканат»	Так (ідеально)	Так	Частково

У порівнянні з іншими технологіями, Visual Studio + C# є найбільш оптимальним вибором для розробки навчальної інформаційної системи, оскільки поєднує високу продуктивність, простоту реалізації інтерфейсу та потужну інтеграцію з базою даних MSSQL Server.

Як система керування базами даних буде використано Microsoft SQL Server. Дана СУБД забезпечує високу швидкодію обробки запитів, підтримку складних зв'язків між таблицями, механізми забезпечення цілісності даних, а також засоби резервного копіювання та відновлення. MSSQL Server є оптимальним вибором для реляційних баз даних у корпоративних та навчальних інформаційних системах [16].

Для розробки програмної частини системи буде використано середовище розробки Visual Studio та мова програмування C#. Visual Studio надає зручні інструменти для створення інтерфейсу користувача, налагодження коду та інтеграції з базами даних. Мова C# дозволяє реалізувати об'єктно-орієнтований підхід, що спрощує структуру програми та підвищує її масштабованість і підтримуваність [17].

Взаємодія між застосунком і базою даних буде здійснюватися через технології .NET (зокрема ADO.NET), що забезпечить ефективний обмін даними та спрощує реалізацію CRUD-операцій (створення, читання, оновлення, видалення) [18].

Обраний стек MSSQL Server + Visual Studio (C#) є оптимальним для реалізації інформаційної системи даного типу, оскільки поєднує високу продуктивність, надійність та зручність розробки.

2.4 Визначення форм вхідних та вихідних даних

Інформаційна система обліку студентів та моніторингу навчального процесу використовує структуровані вхідні та вихідні дані, що забезпечують коректне збереження інформації в базі даних та формування аналітичних результатів.

Вхідні дані інформаційної системи призначені для первинного заповнення та подальшого оновлення бази даних. Вони вводяться користувачами системи (адміністраторами або викладачами) через відповідні форми інтерфейсу та забезпечують актуальність інформації про навчальний процес (таблиця 2.4).

Таблиця 2.4 – Вхідні дані системи

Форма	Опис	Основні поля
Студент	Додавання та редагування інформації про студентів	ПІБ, дата народження, стать, email, телефон, група
Група	Створення навчальних груп	назва групи, курс, спеціальність
Спеціальність	Додавання напрямів підготовки	назва спеціальності

Кінець таблиці 2.4

Форма	Опис	Основні поля
Кафедра	Облік кафедр навчального закладу	назва кафедри
Викладач	Реєстрація викладачів	ПІБ, email, кафедра
Дисципліна	Створення навчальних дисциплін	назва, кількість кредитів
Дисципліни викладача	Призначення дисциплін викладачам	викладач, дисципліна
Оцінка	Внесення результатів навчання	студент, дисципліна, оцінка, семестр, дата

Вихідні дані системи формуються на основі обробки інформації, що зберігається в базі даних. Вони призначені для відображення результатів навчального процесу, аналізу успішності студентів та формування звітності для користувачів системи.

Таблиця 2.5 – Вихідні дані системи

Звіт / форма	Опис
Список студентів	Перелік студентів із зазначенням групи та спеціальності
Відомість оцінок	Детальна інформація про оцінки студентів
Середній бал студентів	Розрахунок середнього балу за семестр або дисципліни
Рейтинг студентів	Сортування студентів за успішністю
Список груп	Інформація про групи та їх склад
Дисципліни викладача	Перелік дисциплін, які викладає конкретний викладач
Успішність групи	Аналітика середнього рівня успішності групи

Звіт / форма	Опис
Статистика оцінок	Розподіл оцінок (відмінно, добре, задовільно тощо)

Запити інформаційної системи забезпечують доступ до необхідних даних та виконують функцію аналітичної обробки інформації. Вони дозволяють формувати вибірки, розрахунки та рейтинги на основі даних, що зберігаються у базі.

Таблиця 2.6 – Основні запити системи

Запит	Опис
Отримати список студентів	Вибірка всіх студентів із групами та спеціальностями
Оцінки студента	Отримання всіх оцінок конкретного студента
Середній бал студента	Розрахунок середнього значення оцінок
Успішність групи	Аналіз середнього балу по групі

Кінець таблиці 2.6

Запит	Опис
Дисципліни викладача	Отримання списку дисциплін конкретного викладача
Студенти певної групи	Фільтрація студентів за групою
Оцінки по дисципліні	Всі оцінки студентів по конкретній дисципліні
Рейтинг студентів	Сортування студентів за середнім балом

Вхідні дані забезпечують формування та наповнення бази даних інформаційної системи, вихідні дані дозволяють отримувати аналітичну інформацію, а система запитів забезпечує гнучкий доступ до даних та формування звітності для аналізу навчального процесу.

2.5 Навігація та взаємодія з користувачем

Інформаційна система обліку студентів та моніторингу навчального процесу має зручний та інтуїтивно зрозумілий інтерфейс, що забезпечує ефективну взаємодію користувача з даними. Навігація в системі організована за модульним принципом, що дозволяє розподілити функціонал на окремі логічні блоки (модулі), кожен з яких відповідає за певну предметну область.

До основних модулів системи належать: модуль управління студентами,

модуль управління навчальними групами, модуль спеціальностей, модуль кафедр, модуль викладачів, модуль дисциплін, модуль обліку оцінок, а також модуль формування звітності та аналітики. Така структура забезпечує чітке розмежування функціональних можливостей, спрощує навігацію та підвищує зручність використання системи.

Таблиця 2.7 – Опис модулів інформаційної системи

Модуль	Призначення	Основні функції
Модуль студентів	Облік та управління інформацією про студентів	Додавання, редагування, видалення студентів; перегляд інформації; прив'язка до груп
Модуль груп	Формування та ведення навчальних груп	Створення груп; редагування; призначення спеціальності; перегляд складу груп

Кінець таблиці 2.7

Модуль	Призначення	Основні функції
Модуль спеціальностей	Управління напрямками підготовки	Додавання та редагування спеціальностей; перегляд переліку
Модуль кафедр	Облік структурних підрозділів	Додавання кафедр; редагування; перегляд викладацького складу
Модуль викладачів	Управління даними про викладачів	Реєстрація викладачів; редагування; прив'язка до кафедри
Модуль дисциплін	Облік навчальних дисциплін	Додавання дисциплін; редагування; перегляд переліку
Модуль дисциплін викладача	Призначення дисциплін викладачам	Зв'язування викладачів із дисциплінами; редагування призначень
Модуль оцінок	Облік результатів навчання студентів	Внесення оцінок; редагування; перегляд оцінок за дисциплінами і семестрами
Модуль звітності	Формування аналітичної інформації	Розрахунок середнього балу; рейтинг студентів; статистика успішності

Модульна архітектура інформаційної системи забезпечує чіткий поділ функціоналу, спрощує супровід програмного продукту та дозволяє масштабувати

систему в майбутньому. Взаємодія між модулями побудована на основі логічних зв'язків між сутностями предметної області, що гарантує цілісність даних та коректність аналітичних результатів.

Висновки до 2 розділу

У другому розділі було виконано проектування інформаційної системи обліку студентів та моніторингу академічних досягнень. Визначено архітектуру системи, яка базується на багаторівневому підході, що забезпечує розподіл функцій між рівнями та підвищує гнучкість і зручність супроводу.

Розроблено структуру бази даних, визначено основні сутності та зв'язки між ними, що гарантує цілісність і коректність даних. Обґрунтовано вибір засобів розробки, які забезпечують реалізацію необхідного функціоналу системи.

Також визначено форми вхідних і вихідних даних та описано принципи навігації і взаємодії з користувачем, що забезпечують зручність роботи з системою.

У результаті сформовано основу для подальшої реалізації інформаційної системи.

3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Створення бази даних у MS SQL Server

Створення бази даних є першим етапом реалізації інформаційної системи обліку студентів та моніторингу навчального процесу. Для реалізації використовується система керування базами даних Microsoft SQL Server, яка забезпечує надійне зберігання даних, підтримку зв'язків між таблицями та високу продуктивність обробки запитів [19].

На початковому етапі створюється нова база даних, яка є контейнером для всіх таблиць системи. Далі визначаються структури таблиць, первинні та зовнішні ключі, а також встановлюються зв'язки між сутностями відповідно до логічної моделі даних (таблиця 3.1).

Таблиця 3.1 – Відповідність сутностей і таблиць БД StudentSystem

Сутність	Таблиця	Поле	Тип ключа	Опис
Спеціальність	Specialty	specialty_id	PK	Ідентифікатор спеціальності
	Specialty	name		Назва спеціальності
Група	Group	group_id	PK	Ідентифікатор групи
	Group	group_name		Назва групи
	Group	course		Курс навчання
	Group	specialty_id	FK	Посилання на Specialty
Кафедра	Department	department_id	PK	Ідентифікатор кафедри
	Department	name		Назва кафедри
Викладач	Teacher	teacher_id	PK	Ідентифікатор викладача
	Teacher	full_name		ПІБ викладача
	Teacher	email		Електронна пошта
	Teacher	department_id	FK	Посилання на Department
Оцінка	Grade	grade_id	PK	Ідентифікатор оцінки
	Grade	student_id	FK	Посилання на Student
	Grade	teacher_discipline_id	FK	Посилання на TeacherDiscipline

Сутність	Таблиця	Поле	Тип ключа	Опис
	Grade	grade_value		Значення оцінки
	Grade	semester		Семестр

Кінець таблиці 3.1

Сутність	Таблиця	Поле	Тип ключа	Опис
Студент	Student	student_id	PK	Ідентифікатор студента
	Student	full_name		ПІБ студента
	Student	birth_date		Дата народження
	Student	gender		Стать
	Student	email		Email
	Student	phone		Телефон
	Student	group_id	FK	Посилання на Group
Дисципліна	Discipline	discipline_id	PK	Ідентифікатор дисципліни
	Discipline	name		Назва дисципліни
	Discipline	credits		Кількість кредитів
Викладач– Дисципліна	TeacherDiscipline	id	PK	Ідентифікатор зв'язку
	TeacherDiscipline	teacher_id	FK	Посилання на Teacher
	TeacherDiscipline	discipline_id	FK	Посилання на Discipline

Схема бази даних представлена на рисунку 3.1.

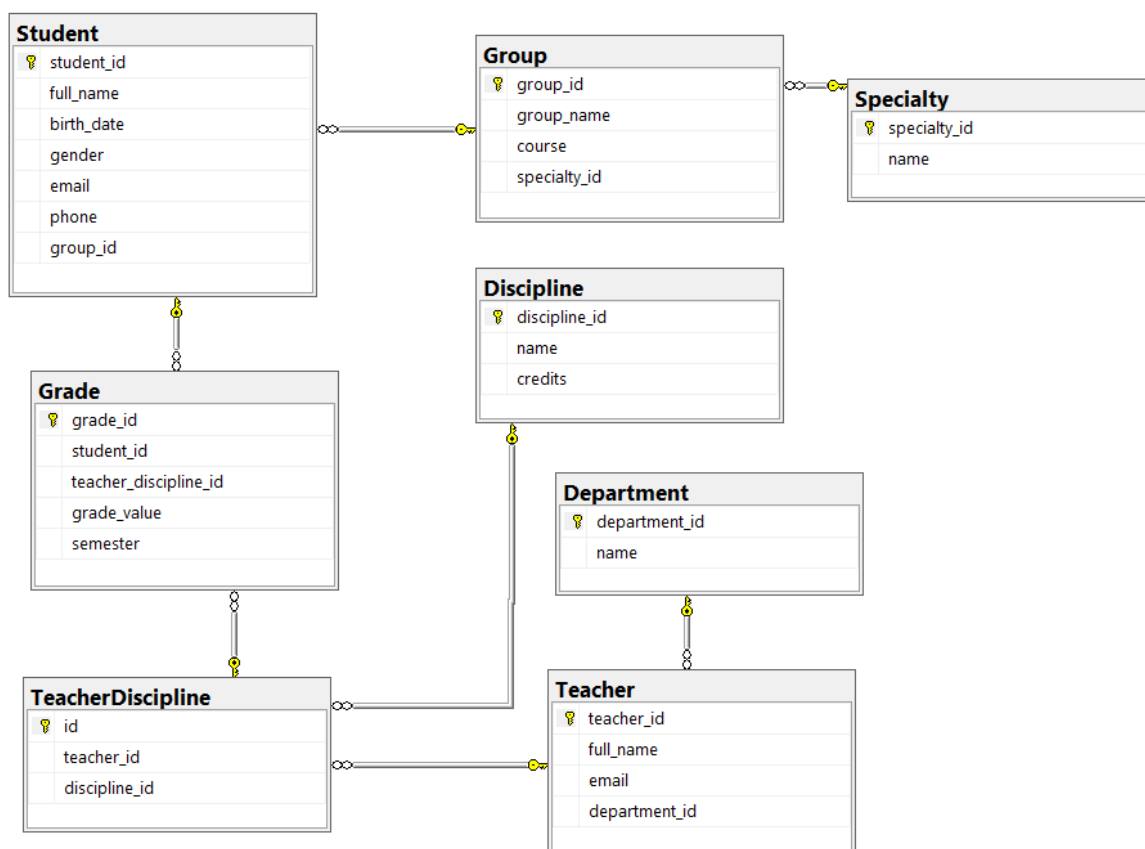


Рисунок 3.1 – Діаграма бази даних

При створенні бази даних використовуються наступні принципи:

- нормалізація даних для усунення надлишковості;
- використання первинних ключів (PK) для унікальної ідентифікації записів та зовнішніх ключів (FK) для забезпечення зв'язків між таблицями;
- цілісність даних через обмеження (NOT NULL, UNIQUE, FOREIGN KEY);
- модульна структура, що відповідає інфологічній моделі даних.

Повний скрипт створення таблиць бази даних інформаційної системи обліку студентів та моніторингу навчального процесу наведено у Додатку А. Він містить SQL-команди для створення бази даних, всіх таблиць бази даних, зв'язків між таблицями, визначених на етапі логічного проектування та запити для створення представлень.

Створена структура бази даних у MS SQL Server повністю відповідає логічній моделі інформаційної системи. Вона забезпечує коректне зберігання даних, підтримку зв'язків між сутностями та можливість подальшої реалізації

функціоналу системи у середовищі Visual Studio з використанням мови програмування C#.

3.2 Реалізація інтерфейсу інформаційної системи

Інтерфейс інформаційної системи «Облік студентів та навчального процесу» реалізовано на основі взаємодії користувача з базою даних StudentSystem, яка побудована на платформі Microsoft SQL Server.

Основною метою інтерфейсу є забезпечення зручного доступу до даних про студентів, викладачів, дисципліни, групи та результати навчання, а також можливість виконання аналітичних запитів.

Інтерфейс системи побудований за трирівневою архітектурою:

- рівень представлення (UI) – форми для роботи користувача (перегляд студентів, оцінок, груп);
- рівень логіки – обробка запитів та формування SQL-звернень;
- рівень даних – база даних StudentSystem.

Зв'язок між інтерфейсом та базою даних здійснюється через SQL-запити до таблиць і представлень (VIEW).

Користувач взаємодіє з системою через графічні форми, які надсилають SQL-запити до бази даних. Результати запитів відображаються у вигляді таблиць.

Форми інтерфейсу представлені на рисунках 3.2-3.15.

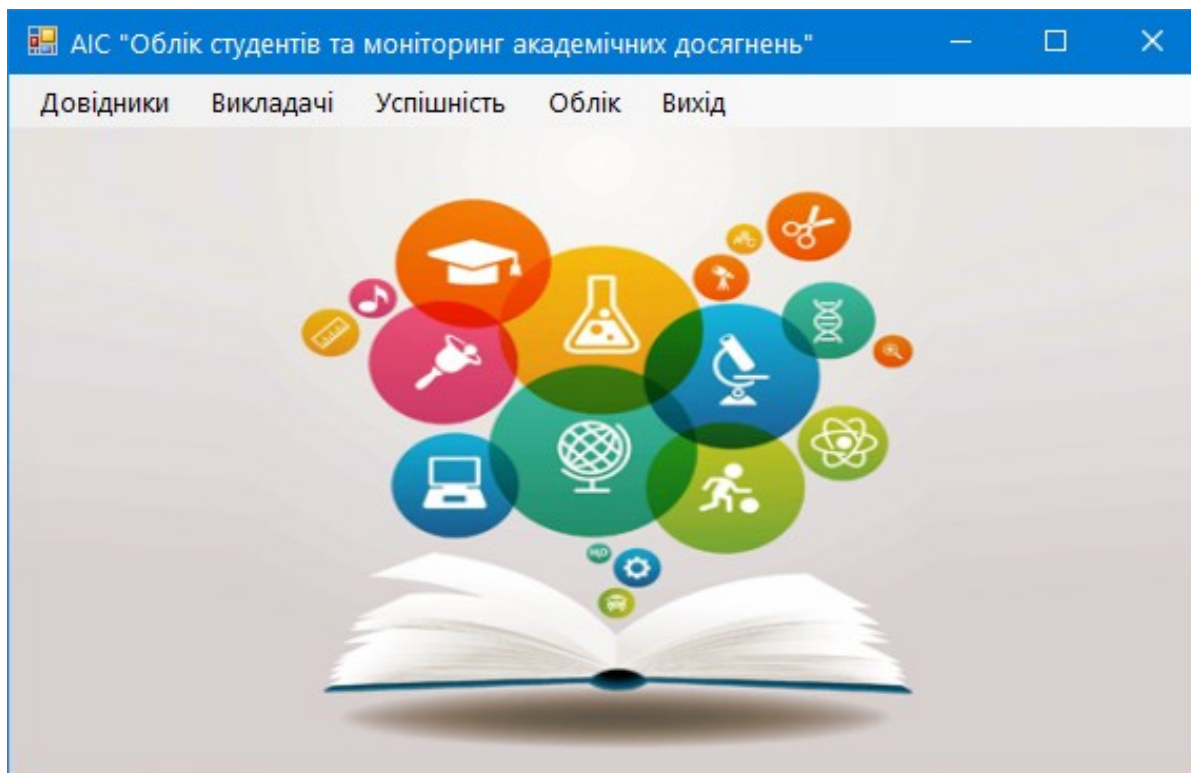


Рисунок 3.2 – Головна форма системи

ПІБ	Дата народження	Стать	E-mail	Телефон	Група
Шевченко О.Ю.	05.01.2026	Жіноча	student1@mail.com	+380670000001	Група-1
Коваленко М.А.	07.09.2025	Чоловіча	student2@mail.com	+380670000002	Група-2
Петренко І.В.	10.05.2025	Жіноча	student3@mail.com	+380670000003	Група-3
Бондар А.С.	10.01.2025	Чоловіча	student4@mail.com	+380670000004	Група-4
Ткачук О.В.	12.09.2024	Жіноча	student5@mail.com	+380670000005	Група-5
Мельник Д.С.	15.05.2024	Чоловіча	student6@mail.com	+380670000006	Група-6
Сидоренко І.О.	16.01.2024	Жіноча	student7@mail.com	+380670000007	Група-7

Рисунок 3.3 – Форма «Довідники», сторінка «Студенти»

Групи

Код групи	Група	Курс	Спеціальність
1	Група-1	2	Програмна інженерія
2	Група-2	3	Кибербезпека
3	Група-3	4	Прикладна математика
4	Група-4	1	Економіка та бізнес
5	Група-5	2	Програмна інженерія
6	Група-6	3	Кибербезпека

Кафедри

department_id	Кафедра
1	Кафедра інформатики
2	Кафедра математики
3	Кафедра економіки
4	Кафедра фізики

Додати Видалити Зберегти Оновити Вихід

Рисунок 3.4 – Форма «Довідники», сторінка «Групи/Кафедри»

Спеціальності

Код	Спеціальність
1	Програмна інженерія
2	Кибербезпека
3	Прикладна математика
4	Економіка та бізнес

Дисципліни

discipline_id	Дисципліна	Година
1	Математичний аналіз	5
2	Лінійна алгебра	4
3	Бази даних	5
4	Програмування	6
5	Операційні системи	5
6	Комп'ютерні мережі	4
7	Основи економіки	3
8	Статистика	4

Додати Видалити Зберегти Оновити Вихід

Рисунок 3.5 – Форма «Довідники», сторінка «Спеціальності/Дисципліни»

Викладачі

ПІБ	E-mail	Кафедра
Іваненко Олександр Петр...	ivanenko@university.edu	Кафедра інформатики
Шевченко Марія Андріївна	shevchenko@university.edu	Кафедра математики
Коваль Дмитро Сергійович	koval@university.edu	Кафедра економіки
Бондаренко Олена Василі...	bondarenko@university.edu	Кафедра фізики
Ткаченко Андрій Миколай...	tkachenko@university.edu	Кафедра інформатики
Мельник Ірина Олегівна	melnik@university.edu	Кафедра математики

Додати Видалити Зберегти Оновити

Кафедра: Кафедра інформатики Скасувати

Вихід

Рисунок 3.6 – Форма «Викладачі і дисципліни», сторінка «Викладачі»

Викладачі і дисципліни

Викладачі | **Дисципліни викладача**

1 для 10

	id	Викладач	Дисципліна
▶	1	Іваненко Олександр Петрович	Математичний аналіз
	2	Шевченко Марія Андріївна	Лінійна алгебра
	3	Коваль Дмитро Сергійович	Бази даних
	4	Бондаренко Олена Василівна	Програмування
	5	Ткаченко Андрій Миколайович	Операційні системи
	6	Мельник Ірина Олегівна	Комп'ютерні мережі

Додати Видалити Зберегти Оновити

Вихід

Рисунок 3.7 – Форма «Викладачі і дисципліни», сторінка «Дисципліни викладача»

Успішність

1 для 1200

	Студент	Дисципліна	Оцінка	Семестр
▶	Шевченко О.Ю.	1	84	1
	Шевченко О.Ю.	2	89	1
	Шевченко О.Ю.	3	72	1
	Шевченко О.Ю.	4	95	1
	Шевченко О.Ю.	5	90	1
	Шевченко О.Ю.	6	64	1
	Шевченко О.Ю.	7	65	1
	Шевченко О.Ю.	8	96	1
	Шевченко О.Ю.	9	63	1
	Шевченко О.Ю.	10	93	1
	Шевченко О.Ю.	1	78	2
	Шевченко О.Ю.	2	83	2
	Шевченко О.Ю.	3	80	2

Додати Видалити Зберегти Оновити Студент Шевченко О.Ю. Скасувати **Вихід**

Рисунок 3.8 – Форма «Успішність»

Облік успішності і досягнень студентів

Дані студентів | Спеціальності студентів | Успішність по семестрах | Дисципліни викладача | Студенти по групах | Середня успішність | Рейтинг студентів

Студент	Група	Курс	Спеціальність
Шевченко О.Ю.	Група-1	2	Програмна інженерія
Коваленко М.А.	Група-2	3	Кибербезпека
Петренко І.В.	Група-3	4	Прикладна математика
Бондар А.С.	Група-4	1	Економіка та бізнес
Ткачук О.В.	Група-5	2	Програмна інженерія
Мельник Д.С.	Група-6	3	Кибербезпека
Сидоренко І.О.	Група-7	4	Прикладна математика
Гришук В.Ю.	Група-8	1	Економіка та бізнес
Лисенко Н.П.	Група-1	2	Програмна інженерія
Кравченко Р.В.	Група-2	3	Кибербезпека
Романенко К.А.	Група-3	4	Прикладна математика

Студент: Шевченко О.Ю.

Скасувати

Вихід

Рисунок 3.9 – Форма «Облік», сторінка «Дані студентів»

Облік успішності і досягнень студентів

Дані студентів | Спеціальності студентів | Успішність по семестрах | Дисципліни викладача | Студенти по групах | Середня успішність | Рейтинг студентів

ПІБ студента	Група	Спеціальність
Бондар А.С.	Група-4	Економіка та бізнес
Бондар А.С.	Група-4	Економіка та бізнес
Гришук В.Ю.	Група-4	Економіка та бізнес
Орлов В.В.	Група-4	Економіка та бізнес
Руденко П.С.	Група-4	Економіка та бізнес
Руденко П.С.	Група-4	Економіка та бізнес
Савченко Ю.М.	Група-4	Економіка та бізнес
Савченко Ю.М.	Група-4	Економіка та бізнес
Бондар А.С.	Група-8	Економіка та бізнес
Гришук В.Ю.	Група-8	Економіка та бізнес
Гришук В.Ю.	Група-8	Економіка та бізнес
Орлов В.В.	Група-8	Економіка та бізнес

Спеціальність:

Скасувати

Вихід

Рисунок 3.10 – Форма «Облік», сторінка «Спеціальності студентів»

Облік успішності і досягнень студентів

Дані студентів | Спеціальності студентів | Успішність по семестрах | Дисципліни викладача | Студенти по групах | Середня успішність | Рейтинг студентів

Дисципліна	Студент	Оцінка	Семестр
Математичний аналіз	Шевченко О.Ю.	84	1
Лінійна алгебра	Шевченко О.Ю.	89	1
Бази даних	Шевченко О.Ю.	72	1
Програмування	Шевченко О.Ю.	95	1
Операційні системи	Шевченко О.Ю.	90	1
Комп'ютерні мережі	Шевченко О.Ю.	64	1
Основи економіки	Шевченко О.Ю.	65	1
Статистика	Шевченко О.Ю.	96	1
Фізика	Шевченко О.Ю.	63	1
Штучний інтелект	Шевченко О.Ю.	93	1
Математичний аналіз	Шевченко О.Ю.	78	2
Лінійна алгебра	Шевченко О.Ю.	83	2

Семестр: ☐ 1 ☒ 2

Скасувати

Вихід

Рисунок 3.11 – Форма «Облік», сторінка «Успішність по семестрах»

Облік успішності і досягнень студентів

Дані студентів | Спеціальності студентів | Успішність по семестрах | **Дисципліни викладача** | Студенти по групах | Середня успішність | Рейтинг студентів

Викладач	Дисципліна
Іваненко Олександр Петрович	Математичний аналіз
Шевченко Марія Андріївна	Лнійна алгебра
Коваль Дмитро Сергійович	Бази даних
Бондаренко Олена Василівна	Програмування
Ткаченко Андрій Миколайович	Операційні системи
Мельник Ірина Олегівна	Комп'ютерні мережі
Сидоренко Віталій Юрійович	Основи економіки
Петренко Наталія Ігорівна	Статистика
Гриценко Роман Вікторович	Фізика
Лисенко Катерина Андріївна	Штучний інтелект
*	

Викладач
Іваненко Олександр Петрович

Скасувати

Вихід

Рисунок 3.12 – Форма «Облік», сторінка «Дисципліни викладача»

Облік успішності і досягнень студентів

Дані студентів | Спеціальності студентів | Успішність по семестрах | Дисципліни викладача | **Студенти по групах** | Середня успішність | Рейтинг студентів

Група	Студент
Група-1	Шевченко О.Ю.
Група-2	Коваленко М.А.
Група-3	Петренко І.В.
Група-4	Бондар А.С.
Група-5	Ткачук О.В.
Група-6	Мельник Д.С.
Група-7	Сидоренко І.О.
Група-8	Гришук В.Ю.
Група-1	Лисенко Н.П.
Група-2	Кравченко Р.В.
Група-3	Романенко К.А.
Група-4	Савченко Ю.М.

Група
Група-2

Скасувати

Вихід

Рисунок 3.13 – Форма «Облік», сторінка «Студенти по групах»

Облік успішності і досягнень студентів

Дані студентів | Спеціальності студентів | Успішність по семестрах | Дисципліни викладача | Студенти по групах | **Середня успішність** | Рейтинг студентів

Середній бал студента по дисциплінах

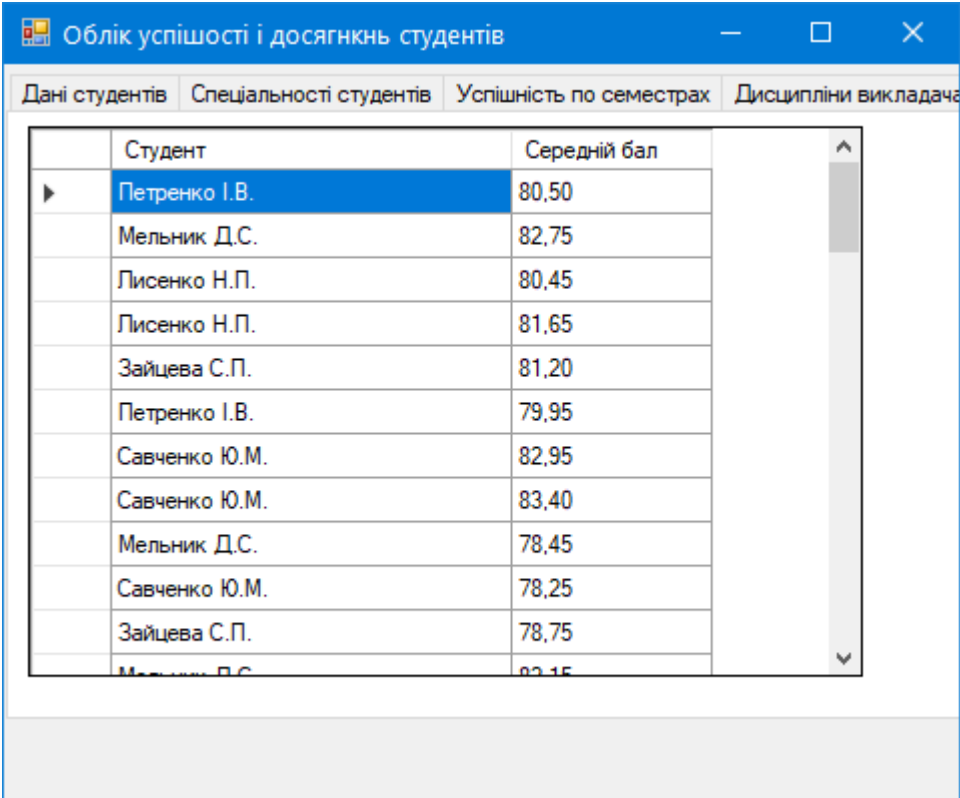
Студент	Середній бал
Петренко І.В.	80,50
Мельник Д.С.	82,75
Лисенко Н.П.	80,45
Лисенко Н.П.	81,65
Зайцева С.П.	81,20
Петренко І.В.	79,95
Савченко Ю.М.	82,95
Савченко Ю.М.	83,40
Мельник Д.С.	78,45
Савченко Ю.М.	78,25
Зайцева С.П.	78,75

Середній бал по групах

Група	Середній бал
Група-3	78,71
Група-6	81,16
Група-7	79,59
Група-1	79,98
Група-4	80,84
Група-5	81,13
Група-2	78,83
Група-8	80,24
*	

Вихід

Рисунок 3.14 – Форма «Облік», сторінка «Середня успішність»



Студент	Середній бал
Петренко І.В.	80,50
Мельник Д.С.	82,75
Лисенко Н.П.	80,45
Лисенко Н.П.	81,65
Зайцева С.П.	81,20
Петренко І.В.	79,95
Савченко Ю.М.	82,95
Савченко Ю.М.	83,40
Мельник Д.С.	78,45
Савченко Ю.М.	78,25
Зайцева С.П.	78,75
Мельник Д.С.	82,15

Рисунок 3.15 – Форма «Облік», сторінка «Рейтинг студентів»

У розробленій інформаційній системі реалізовано механізм фільтрації даних, який дозволяє користувачу швидко знаходити необхідну інформацію та працювати лише з релевантними записами. Фільтрація застосовується до табличних даних, що відображаються у вигляді таблиць, і виконується без повторного звернення до бази даних.

Основою реалізації є використання компонента `BindingSource`, який забезпечує зв'язок між джерелом даних та елементами інтерфейсу. Завдяки цьому фільтрація відбувається на рівні клієнтського додатку, що підвищує швидкодію системи та зменшує навантаження на базу даних.

Фільтрація здійснюється за допомогою властивості `Filter`, яка дозволяє застосовувати умови відбору до даних, аналогічно до оператора `WHERE` у SQL-запитах. Наприклад, при виборі значення у випадяючому списку (`ComboBox`) формується умова фільтрації, яка обмежує відображення записів відповідно до вибраного параметра.

Реалізований інтерфейс забезпечує зручну роботу з навчальними даними та дозволяє виконувати як операційні, так і аналітичні задачі. Використання

представлень (VIEW) спрощує доступ до складних запитів і підвищує ефективність взаємодії користувача з системою.

3.3 Реалізація інтерфейсу адміністрування та розмежування прав доступу

В інформаційній системі «Облік студентів та навчального процесу» реалізовано механізм розмежування доступу користувачів на рівні інтерфейсу. Доступ до функціоналу системи здійснюється через форму авторизації (логін/пароль), яка визначає роль користувача після входу в систему.

При запуску програми користувачу відображається вікно авторизації, що містить:

- поле для введення логіну;
- поле для введення пароля;
- кнопку «Увійти».

Після введення даних система перевіряє їх коректність та визначає роль користувача.

Ролі користувачів та їх права доступу представлені в таблиці 3.2.

Таблиця 3.2 – Ролі користувачів та їх можливості

Роль	Можливості в системі	Доступ до інтерфейсу
Адміністратор	Повне керування студентами (додавання/редагування/видалення) Керування викладачами Керування дисциплінами Формування груп Перегляд та редагування всіх оцінок Керування користувачами системи	Повний доступ до всіх модулів системи
Викладач	Перегляд студентів своїх дисциплін Перегляд закріплених дисциплін Внесення та редагування оцінок Перегляд успішності студентів	Обмежений доступ (модулі оцінок і дисциплін)

Після успішної авторизації система перевіряє роль користувача та формує відповідний набір доступних елементів інтерфейсу.

Реалізація здійснюється через умовне відображення модулів, наприклад:

- для адміністратора відображаються всі вкладки;
- для викладача приховуються адміністративні функції.

Розглянемо логіку роботи інтерфейсу.

Користувач запускає систему та вводить свій логін і пароль у форму входу. Після цього система перевіряє правильність введених даних. Якщо логін і пароль введені коректно, система визначає роль користувача.

Залежно від ролі відкривається відповідний інтерфейс системи. Для адміністратора завантажується повний інтерфейс із доступом до всіх функцій (AdminPanel). Для викладача відкривається обмежений інтерфейс, що дозволяє працювати з дисциплінами та оцінками (TeacherPanel).

Якщо дані введені неправильно, система повідомляє про помилку та пропонує повторити спробу входу.

Форма авторизації представлена на рисунку 3.16.

The image contains two side-by-side screenshots of a Windows-style application window titled 'Авторизація' (Authorization). Each window has a blue title bar with standard minimize, maximize, and close buttons. The main area is light gray. On the left, the 'Логін' (Login) field is empty, and the 'Пароль' (Password) field is also empty. On the right, the 'Логін' field contains the text 'admin', and the 'Пароль' field contains a series of asterisks. Both windows have two buttons at the bottom: 'Вхід' (Login) on the left and 'Вихід' (Logout) on the right.

Рисунок 3.16 – Форма авторизації

Реалізований механізм авторизації забезпечує просте та ефективне розмежування доступу користувачів через інтерфейс системи. Такий підхід дозволяє приховувати недоступні функції та підвищує рівень безпеки та зручності використання системи.

3.4 Тестування розробленої інформаційної системи

Тестування інформаційної системи «Облік студентів та навчального процесу» було проведено з метою перевірки правильності функціонування

системи, коректності обробки даних та відповідності реалізованого функціоналу поставленим вимогам. Для підтвердження працездатності системи було виконано низку тестових сценаріїв, які показали відповідність фактичних результатів очікуванім [20].

Приклади тестових сценаріїв представлено в таблиці 3.3.

Таблиця 3.3 – Тестові сценарії перевірки роботи системи

№	Користувач	Дія	Очікуваний результат	Фактичний результат
1	Адміністратор	Вхід у систему з правильними даними	Успішна авторизація, відкривається AdminPanel	Виконано
2	Адміністратор	Додавання нового студента	Студент з'являється у списку	Виконано
3	Адміністратор	Редагування даних студента	Дані студента оновлюються	Виконано
4	Адміністратор	Видалення студента	Студент видаляється зі списку	Виконано
5	Адміністратор	Додавання нової дисципліни	Дисципліна додається в систему	Виконано
6	Викладач	Вхід у систему з правильними даними	Успішна авторизація, відкривається TeacherPanel	Виконано
7	Викладач	Перегляд списку студентів	Відображається список студентів	Виконано
8	Викладач	Внесення оцінки студенту	Оцінка зберігається в системі	Виконано
9	Викладач	Редагування оцінки	Оцінка оновлюється	Виконано
10	Викладач	Спроба видалити студента	Операція заборонена	Виконано

У процесі тестування було перевірено роботу основних функцій системи, зокрема операції додавання, редагування та видалення даних про студентів, викладачів і дисципліни. Особлива увага приділялась перевірці коректності відображення інформації та збереження змін у базі даних. Також було перевірено механізм внесення та перегляду оцінок, а також формування аналітичних показників, таких як середній бал та рейтинг студентів. Усі функції працюють коректно, без помилок.

Окремо було протестовано процес авторизації користувачів. Перевірка показала, що система правильно обробляє введені логін і пароль, надає доступ у разі коректних даних та відмовляє у вході при їх неправильному введенні. Після успішної авторизації система визначає роль користувача та відкриває відповідний інтерфейс.

Також було перевірено механізм розмежування доступу. У ході тестування встановлено, що кожна роль має доступ лише до передбачених функцій: адміністратор має повний доступ до системи, викладач може працювати з оцінками та дисциплінами, а студент має доступ лише до перегляду власних даних. Спроби виконання заборонених дій коректно блокуються системою.

Перевірка роботи бази даних показала, що всі зв'язки між таблицями функціонують правильно, SQL-запити виконуються без помилок, а представлення забезпечують коректне отримання необхідної інформації.

У результаті тестування встановлено, що інформаційна система працює стабільно, всі реалізовані функції виконуються коректно, а доступ користувачів до даних обмежений відповідно до їх ролей.

Висновки до 3 розділу

У третьому розділі було реалізовано інформаційну систему обліку студентів та моніторингу академічних досягнень. Створено базу даних у MS SQL Server, яка забезпечує збереження, обробку та цілісність даних.

Реалізовано інтерфейс системи, що дозволяє користувачам зручно працювати з інформацією про студентів, дисципліни та оцінки. Окремо розроблено механізм авторизації та розмежування прав доступу, який забезпечує різні рівні доступу для адміністратора та викладача.

Проведене тестування підтвердило коректність роботи системи, правильність обробки даних і відповідність реалізованого функціоналу поставленим вимогам.

У результаті створено працездатну інформаційну систему, готову до

використання та подальшого розвитку.

ВИСНОВКИ

У процесі виконання роботи було досліджено предметну область обліку студентів та моніторингу їх академічних досягнень. Проведено аналіз основних бізнес-процесів, пов'язаних із веденням обліку студентів, формуванням груп, закріпленням дисциплін та контролем результатів навчання. На основі цього сформовано вимоги до інформаційної системи, яка повинна забезпечувати ефективну обробку, збереження та аналіз даних. Також було здійснено огляд існуючих програм-аналогів, що дозволило визначити переваги та недоліки сучасних рішень.

У другому розділі виконано проектування інформаційної системи. Розроблено її архітектуру з урахуванням розподілу функцій між основними компонентами. Спроектовано структуру бази даних, визначено сутності, їх атрибути та зв'язки, що забезпечують цілісність і узгодженість даних. Обрано засоби розробки, які дозволяють реалізувати систему відповідно до поставлених вимог. Крім того, визначено форми вхідних і вихідних даних та описано принципи навігації і взаємодії користувача з системою, що забезпечують зручність її використання.

У третьому розділі реалізовано інформаційну систему. Створено базу даних у середовищі MS SQL Server, реалізовано основні таблиці, зв'язки та запити для роботи з даними. Розроблено інтерфейс системи, який забезпечує доступ до функціоналу відповідно до ролі користувача. Реалізовано механізм авторизації та розмежування прав доступу, що дозволяє забезпечити безпечну роботу з даними. Проведене тестування підтвердило коректність роботи системи, стабільність її функціонування та відповідність поставленим вимогам.

У результаті виконаної роботи розроблено інформаційну систему, яка автоматизує процес обліку студентів і моніторингу їх академічних досягнень. Система забезпечує зручний доступ до даних, їх обробку та аналіз, підвищує ефективність роботи користувачів і може бути використана як основа для подальшого розвитку, зокрема розширення функціоналу та інтеграції з іншими

інформаційними системами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Авраменко В.С. Проектування інформаційних систем: навчальний посібник / В.С. Авраменко, А.С. Авраменко. – Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2017. – 434 с.
2. Івашкевич, М. І., Шишкін, О. В. Проектування інформаційних систем : навч. посіб. – Харків : ХНУРЕ, 2019. – 280 с.
3. Moodle [Електронний ресурс]. – Режим доступу: <https://moodle.org/> (дата звернення: 27.04.2026).
4. Google Classroom [Електронний ресурс]. – Режим доступу: <https://classroom.google.com/> (дата звернення: 27.04.2026).
5. Quizizz – платформа для створення інтерактивних вікторин і тестів. – Режим доступу: <https://quizizz.com/> (дата звернення: 27.04.2026).
6. Морзе Н. В. Основи інформаційно-комунікаційних технологій. – Київ: Видавнича група BHV, 2019. – 352 с.
7. Мулеса О.Ю. Інформаційні системи та реляційні бази даних. Навч.посібник. – Ужгород : Ужгородський національний університет», 2018. – 118 с.
8. Бази даних/організація баз даних: методичні рекомендації для виконання практичних занять / уклад.: Ю. Є. Добришин – Київ: Університет економіки та права «КРОК», 2017. – 130 с.
9. Кашниця, В. М. Теорія баз даних: навч. посіб. / В. М. Кашниця, В. В. Тихомиров. – Львів: ЛНУ ім. І. Франка, 2017. – 500 с.
10. Microsoft SQL Server – система управління базами даних. [Електронний ресурс]. – Режим доступу: <https://www.microsoft.com/sql-server/> (дата звернення: 28.04.2026).
11. MySQL – система управління базами даних. [Електронний ресурс]. – Режим доступу: <https://www.mysql.com/> (дата звернення: 28.04.2026).
12. PostgreSQL – система управління базами даних. [Електронний ресурс]. – Режим доступу: <https://www.postgresql.org/> (дата звернення: 28.04.2026).
13. Visual Studio – інтегроване середовище розробки [Електронний ресурс]. – Режим доступу: <https://visualstudio.microsoft.com/> (дата звернення: 29.04.2026).

14. IntelliJ IDEA [Електронний ресурс]. – Режим доступу: <https://www.jetbrains.com/idea/> (дата звернення: 29.04.2026).
15. PyCharm – інтегроване середовище розробки для Python [Електронний ресурс]. – Режим доступу: <https://www.jetbrains.com/pycharm/> (дата звернення: 29.04.2026).
16. Черниш, І. П. Системи управління базами даних : навчальний посібник. – Львів: Львівська політехніка, 2020. – 312 с.
17. Петриченко О. С. Розробка застосунків на платформі .NET / О. С. Петриченко. – Київ : Видавництво Ліра-К, 2020. – 256 с.
18. База знань Prometheus. Курс «Основи програмування мовою C# та .NET» [Електронний ресурс]. – Режим доступу: <https://courses.prometheus.org.ua/> (дата звернення: 28.04.2026)
19. Трофименко О. Г. Організація баз даних : навч. посібник / О. Г. Трофименко, Ю. В. Прокоп, Н. І. Логінова, І. М. Копитчук. 2-ге вид. виправ. і доповн. – Одеса : Фенікс, 2019. – 246 с.
20. Метод «чорної скриньки» в тестуванні програмного забезпечення [Електронний ресурс]. – Режим доступу: <https://www.softwaretestinghelp.com/black-box-testing/> (дата звернення: 30.04.2026).

ДОДАТОК А

СКРИПТИ СТВОРЕННЯ БАЗИ ДАНИХ ТА ЇЇ ТАБЛИЦЬ

```
-- =====
-- Створення бази даних
-- =====
CREATE DATABASE StudentSystem;
GO

USE StudentSystem;
GO

-- =====
-- Таблиця: Спеціальність
-- =====
CREATE TABLE Specialty (
    specialty_id INT IDENTITY(1,1) PRIMARY KEY,
    name NVARCHAR(100) NOT NULL
);

-- =====
-- Таблиця: Група
-- =====
CREATE TABLE [Group] (
    group_id INT IDENTITY(1,1) PRIMARY KEY,
    group_name NVARCHAR(50) NOT NULL,
    course INT NOT NULL,
    specialty_id INT NOT NULL,
    FOREIGN KEY (specialty_id) REFERENCES Specialty(specialty_id)
);

-- =====
-- Таблиця: Кафедра
-- =====
CREATE TABLE Department (
    department_id INT IDENTITY(1,1) PRIMARY KEY,
    name NVARCHAR(100) NOT NULL
);

-- =====
-- Таблиця: Викладач
-- =====
CREATE TABLE Teacher (
    teacher_id INT IDENTITY(1,1) PRIMARY KEY,
```

```

    full_name NVARCHAR(150) NOT NULL,
    email NVARCHAR(100),
    department_id INT NOT NULL,
    FOREIGN KEY (department_id) REFERENCES Department(department_id)
);

```

```

-- =====
-- Таблиця: Студент
-- =====

```

```

CREATE TABLE Student (
    student_id INT IDENTITY(1,1) PRIMARY KEY,
    full_name NVARCHAR(150) NOT NULL,
    birth_date DATE,
    gender NVARCHAR(10),
    email NVARCHAR(100),
    phone NVARCHAR(30),
    group_id INT NOT NULL,
    FOREIGN KEY (group_id) REFERENCES [Group](group_id)
);

```

```

-- =====
-- Таблиця: Дисципліна
-- =====

```

```

CREATE TABLE Discipline (
    discipline_id INT IDENTITY(1,1) PRIMARY KEY,
    name NVARCHAR(100) NOT NULL,
    credits INT NOT NULL
);

```

```

-- =====
-- Таблиця: Дисципліни викладача
-- =====

```

```

CREATE TABLE TeacherDiscipline (
    id INT IDENTITY(1,1) PRIMARY KEY,
    teacher_id INT NOT NULL,
    discipline_id INT NOT NULL,
    FOREIGN KEY (teacher_id) REFERENCES Teacher(teacher_id),
    FOREIGN KEY (discipline_id) REFERENCES Discipline(discipline_id)
);

```

```

-- =====
-- Таблиця: Оцінка
-- =====

```

```

CREATE TABLE Grade (
    grade_id INT IDENTITY(1,1) PRIMARY KEY,

```

```

        student_id INT NOT NULL,
        teacher_discipline_id INT NOT NULL,
        grade_value INT NOT NULL,
        semester INT NOT NULL,
        FOREIGN KEY (student_id) REFERENCES Student(student_id),
        FOREIGN KEY (teacher_discipline_id) REFERENCES
TeacherDiscipline(id)
);

```

```

-- =====
-- ПРЕДСТАВЛЕННЯ (VIEW)
-- =====

```

```

USE StudentSystem;
GO

```

```

-- =====
-- 1. Список студентів (група + спеціальність)
-- =====

```

```

CREATE VIEW vw_StudentsList AS
SELECT
    s.full_name AS Student,
    g.group_name AS [Group],
    g.course,
    sp.name AS Specialty
FROM Student s
JOIN [Group] g ON s.group_id = g.group_id
JOIN Specialty sp ON g.specialty_id = sp.specialty_id;
GO

```

```

-- =====
-- 2. Усі оцінки студентів
-- =====

```

```

CREATE VIEW vw_StudentAllGrades AS
SELECT
    s.full_name AS Student,
    d.name AS Discipline,
    g.grade_value,
    g.semester
FROM Grade g
JOIN Student s ON g.student_id = s.student_id
JOIN TeacherDiscipline td ON g.teacher_discipline_id = td.id
JOIN Discipline d ON td.discipline_id = d.discipline_id;
GO

```

```
-- =====
-- 3. Середній бал студента
-- =====
```

```
CREATE VIEW vw_StudentAverageGrade AS
SELECT
    s.full_name AS Student,
    AVG(g.grade_value * 1.0) AS AverageGrade
FROM Student s
JOIN Grade g ON s.student_id = g.student_id
GROUP BY s.student_id, s.full_name;
GO
```

```
-- =====
-- 4. Успішність групи
-- =====
```

```
CREATE VIEW vw_GroupPerformance AS
SELECT
    g.group_name,
    AVG(gr.grade_value * 1.0) AS AverageGrade
FROM [Group] g
JOIN Student s ON g.group_id = s.group_id
JOIN Grade gr ON s.student_id = gr.student_id
GROUP BY g.group_id, g.group_name;
GO
```

```
-- =====
-- 5. Дисципліни викладача
-- =====
```

```
CREATE VIEW vw_TeacherDisciplines AS
SELECT
    t.full_name AS Teacher,
    d.name AS Discipline
FROM TeacherDiscipline td
JOIN Teacher t ON td.teacher_id = t.teacher_id
JOIN Discipline d ON td.discipline_id = d.discipline_id;
GO
```

```
-- =====
-- 6. Студенти певної групи
-- =====
```

```
CREATE VIEW vw_StudentsByGroup AS
SELECT
    g.group_name,
    s.full_name AS Student
```

```

FROM Student s
JOIN [Group] g ON s.group_id = g.group_id;
GO

-- =====
-- 7. Оцінки по дисципліні
-- =====
CREATE VIEW vw_GradesByDiscipline AS
SELECT
    d.name AS Discipline,
    s.full_name AS Student,
    g.grade_value,
    g.semester
FROM Grade g
JOIN Student s ON g.student_id = s.student_id
JOIN TeacherDiscipline td ON g.teacher_discipline_id = td.id
JOIN Discipline d ON td.discipline_id = d.discipline_id;
GO

-- =====
-- 8. Рейтинг студентів
-- =====
CREATE VIEW vw_StudentRating AS
SELECT
    s.full_name AS Student,
    AVG(g.grade_value * 1.0) AS AverageGrade
FROM Student s
JOIN Grade g ON s.student_id = g.student_id
GROUP BY s.student_id, s.full_name;
GO

```

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

Декларація про академічну доброчесність та використання ШІ

Я, Карпович Роман Петрович, здобувач НУБіП України, усвідомлюючи відповідальність за порушення академічної доброчесності, цією заявою підтверджую, що:

1. Моя бакалаврська кваліфікаційна робота на тему «Розробка інформаційної системи обліку студентів та моніторингу академічних досягнень» є результатом моєї особистої праці.
2. Усі запозичення з текстів інших авторів мають відповідні посилання згідно з чинними стандартами.
3. Щодо використання інструментів ШІ зазначаю, що (обрати необхідне):
 - інструменти генеративного ШІ не використовувалися.
 - інструменти ШІ (ChatGPT) були використані для:
 - перекладу іншомовних джерел;
 - стилістичного редагування та перевірки граматики;
 - допомоги у написанні/відлагодженні програмного коду;
 - інше: консультації щодо структури пояснювальної записки та підготовки до захисту.

Я розумію, що надання недостовірної інформації може призвести до скасування результатів захисту та відрахування з числа здобувачів НУБіП України.

«___» _____ 2026 р. _____

Роман КАРПОВИЧ