

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Західноукраїнський національний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра комп'ютерної інженерії

Островський Андрій Петрович

**База даних обліку успішності студентів /  
Database of students' performance record**

спеціальність: 123 – Комп'ютерна інженерія  
освітньо-професійна програма – Комп'ютерна інженерія

Кваліфікаційна робота

Виконав: студент групи КІ–41  
Островський Андрій Петрович

Науковий Керівник  
к.т.н. Савка Н. Я.

ТЕРНОПІЛЬ 2023

## РЕЗЮМЕ

Кваліфікаційна робота на тему «База даних обліку успішності студентів» зі спеціальності 123 «Комп'ютерна інженерія» освітнього ступеня «бакалавр» містить 80 сторінок пояснюючої записки, 28 рисунків, 16 таблиць, 2 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою кваліфікаційної роботи є розроблення бази даних обліку успішності студентів.

База даних обліку успішності студентів включає методи збору, організації та аналізу даних.

Розроблено програмне забезпечення для бази даних обліку успішності студентів, яке включає такі функціональні можливості: збір даних про академічні досягнення студентів, автоматизоване отримання інформації про їх успішність; аналіз та обробка даних про виконані завдання та оцінки; зберігання цих даних в базі даних; виведення рейтингу студентів залежно від їхніх досягнень;

Ключові слова: БАЗА ДАНИХ, ОБЛІК УСПІШНОСТІ.

## RESUME

The qualification work on the topic "Database for recording students' success" in the field of Computer Engineering, Bachelor's degree program, consists of an 80-page explanatory note, 28 figures, 16 tables, and 2 appendices. The amount of graphic material is 2 sheets of A3 format.

The aim of the qualification work is to develop a database for recording students' success.

The database for recording students' success includes methods of data collection, organization, and analysis.

Software has been developed for the database of students' performance records, which includes the following functionalities: data collection on students' academic achievements, automated retrieval of information on their performance, analysis and processing of data on completed tasks and assessments, storage of this data in the database, derivation of students' rating based on their achievements.

Keywords: DATABASE, RECORDING OF SUCCESS.

## ЗМІСТ

Перелік умовних скорочень .....	9
Вступ.....	10
1 Успішність студентів як результат навчального процесу.....	12
1.1 Аналіз предметної області .....	12
1.2 Аналіз систем обліку успішності .....	16
1.3 Аналіз технологій автоматизації обліку успішності .....	23
1.4 Постановка задач кваліфікаційної роботи.....	26
2 Алгоритми проектування бази даних обліку успішності.....	29
2.1 Програмні інструменти організації обліку успішності.....	29
2.2 Структура бази даних .....	32
2.3 Алгоритм проектування бази даних обліку успішності .....	38
3 Програмна реалізація системи .....	44
3.1 Інфологічна та логічна моделі баз даних.....	44
3.2 Фізична модель баз даних .....	49
3.3 Програмна реалізація систем обліку успішності.....	52
3.4 Тестування та верифікація розробленого програмного забезпечення .....	58
4 Техніко–економічний розділ.....	65
4.1 Визначення витрат на оплату праці та відрахувань у соціальні фонди .....	65
Висновки .....	73
Список використаних джерел .....	74
Додаток А Світлокопії виданих публікацій .....	79
Додаток Б Довідка про використання .....	80

					КР.КІ.9500102.00.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розробив		Островський А.П			БАЗА ДАНИХ ОБЛІКУ УСПІШНОСТІ СТУДЕНТІВ	Літ.	Арк.	Акрушів
Перевір.		Мельник Г.М.					7	
Консульт.		Савка Н.Я.				ЗУНУ,ФКІТ, КІ-41		
Н. Контр.		Мельник Г.М.						
Затвердив		Дубчак Л.О.						

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

SQL	–	Structured Query Language
LMS	–	Learning Management System
ОС	–	Операційна система
РБД	–	Реляційна база даних
РСКБД	–	Реляційна система керування базами даних
ID	–	Унікальний ідентифікатор
ER	–	Entity–Relationship
БД	–	База даних
СУБД	–	Система управління базами даних

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

## ВСТУП

У сучасному освітньому середовищі, де знання та навички є ключовими факторами розвитку і успіху, важливо мати систему обліку та оцінки успішності студентів. Студентська успішність не тільки є мірою досягнень індивідуальних учнів, але й впливає на репутацію та якість навчального закладу.

У цьому контексті, розробка та впровадження бази даних, спеціально призначеної для обліку успішності студентів, стає незамінним інструментом, який допомагає збирати, зберігати та аналізувати інформацію про навчальні досягнення студентів, їх академічні результати, участь у навчальних заходах та інші показники успішності. Вона не лише сприяє забезпеченню точності та цілісності даних, але й надає можливості для ефективного аналізу та прийняття обґрунтованих управлінських рішень. Актуальність реалізації бази даних обліку успішності студентів полягає в необхідності ефективного збору, зберігання та аналізу даних про успішність студентів для забезпечення кращого управління освітнім процесом, виявлення слабких місць у навчанні, підтримки прийняття рішень та забезпечення персоналізованого підходу до студентів. Таким чином, виникає необхідність в розробці системи бази даних для автоматизації процедури обліку успішності студентів у навчальному процесі.

Метою даної дипломної роботи є розробка та реалізація бази даних обліку успішності студентів, яка буде спеціально адаптована до потреб конкретного навчального закладу. Для досягнення мети роботи, необхідно розробити такі задачі:

Об'єктом дослідження є навчальний процес.

Предметом дослідження є алгоритми та програмні засоби проектування бази даних для забезпечення автоматизації обліку успішності.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

Практичне значення полягає у розробці програмної системи автоматизації обліку успішності студентів.

Для розробки та реалізації бази даних обліку успішності студентів, важливим етапом є розробка програми. Розробка програми означає розбиття загальної задачі на менші частини або модулі, що спрощує процес розробки та полегшує впровадження функціональності бази даних. Програма повинна бути інтуїтивно зрозумілою. Вона повинна надавати легкий доступ до основних функцій, таких як внесення нових даних, перегляд та аналітика успішності. Дослідження передбачає аналіз вимог та потреб студентів, викладачів та адміністрації, розробку структури бази даних, розроблення відповідних інтерфейсів для введення та відображення даних, а також забезпечення захисту конфіденційності та безпеки інформації.

Впровадження бази даних обліку успішності студентів має на меті поліпшити процеси оцінювання та моніторингу навчальних досягнень, сприяючи зростанню якості освіти та підвищенню рівня академічної успішності студентів. Крім того, вона може стати ефективним інструментом для планування навчальних програм, адаптації підходів до навчання та розвитку індивідуальних навчальних траєкторій.

За результатами досліджень у роботі опубліковано тези доповіді на VII науково-практична конференція "інтелектуальні комп'ютерні системи та мережі", ЗУНУ кафедра компютерної інженерії. Копії публікації наведено у додатку А.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

# 1 УСПІШНІСТЬ СТУДЕНТІВ ЯК РЕЗУЛЬТАТ НАВЧАЛЬНОГО ПРОЦЕСУ

## 1.1 Аналіз предметної області

Дослідження предметної області є важливим етапом, оскільки воно дозволяє оглянути та проаналізувати існуючі методи, підходи та програмні засоби, що застосовуються для обліку успішності студентів. Це включає вивчення актуальної літератури, наукових публікацій, досліджень та практичного досвіду, пов'язаного з цією темою.

Успішність студентів відноситься до оцінки та вимірювання їхнього академічного прогресу та досягнень у навчанні. Це включає оцінки, рейтинги, відвідуваність, здатність до самостійного навчання, участь у додаткових академічних програмах та інші показники, що відображають успіх студента у навчанні.

Успішність студентів є важливим показником ефективності освітнього процесу та досягнення навчальних цілей. Вона може бути визначена за допомогою різних критеріїв, таких як середній бал, пропорція високих оцінок, відсоток успішно складених іспитів, академічні нагороди тощо. Успішність студентів може впливати на їхню подальшу кар'єру, отримання стипендій, можливості професійного розвитку та подальше навчання.

Успішність студентів має важливе значення як для самого студента, так і для навчального закладу. Для студента успішність відображає його зусилля та здатність відповідати академічним вимогам. Вона може впливати на його майбутні перспективи з працевлаштування та можливості отримання стипендій чи стипендійних програм.

Для навчального закладу успішність студентів є показником якості освіти, ефективності навчальних програм та викладацького персоналу. Вона може використовуватися для оцінки результативності навчального закладу, порівняння з іншими закладами, а також прийняття рішень щодо поліпшення навчального процесу та планування додаткових заходів підтримки для студентів.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12



Під час дослідження предметної області проводиться огляд і аналіз існуючих аналогів та програмних засобів, що реалізують функції систем обліку успішності в академічних групах. Це дозволяє виявити їх переваги, недоліки та особливості функціонування.

Існують різноманітні методики, підходи та програмні засоби, які використовуються в освітніх установах для обліку та аналізу успішності студентів. Деякі з них включають:

1. Традиційний паперовий облік успішності: Цей метод передбачає ведення ручного обліку студентських результатів у паперових журналах або електронних таблицях. Викладачі вручну вводять та оцінюють дані, що вимагає значних зусиль і може бути часом незручним.

Позитивні сторони:

- Простота впровадження та використання.
- Відсутність потреби в спеціалізованому програмному забезпеченні.

Негативні сторони:

- Підвищена часова затримка при обробці та аналізі даних.
- Висока ймовірність помилок при ручному вводі даних.
- Обмежені можливості аналітики та звітності.

2. Електронні таблиці та програми для обліку успішності: Використання програмних засобів, таких як Microsoft Excel або Google Sheets, для створення електронних таблиць дозволяє зручно зберігати та обробляти дані успішності студентів. Проте ці рішення можуть бути обмежені в можливостях аналізу та генерації звітів.

Позитивні сторони:

- Зручне зберігання та обробка даних.
- Можливість використання базових аналітичних функцій.

Негативні сторони:

- Висока витрата часу на введення та оновлення даних.
- Обмежені можливості комплексного аналізу та звітності.
- Ризик втрати або пошкодження даних при некоректній обробці.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

3. Спеціалізовані системи управління навчальним процесом (LMS): Багато навчальних закладів використовують LMS, такі як Moodle або Blackboard, для обліку успішності студентів. Ці системи дозволяють вести журнали оцінок, стежити за відвідуванням, надавати доступ до навчального матеріалу та спілкуватися зі студентами. Вони можуть мати обмежені можливості аналітики та адаптації до конкретних потреб навчального закладу.

Позитивні сторони:

- Централізоване зберігання та доступ до даних успішності студентів.
- Можливість ведення журналів, спілкування зі студентами та надання навчального матеріалу.
- Деякі можливості аналітики та звітності.

Негативні сторони:

- Обмежені можливості настройки та адаптації під конкретні потреби навчального закладу.
- Залежність від постачальників системи LMS.
- Необхідність додаткового навчання та підтримки для використання системи.

4. Спеціалізовані системи обліку успішності: Існують програмні рішення, спеціально розроблені для обліку успішності студентів, які надають більш широкі можливості аналізу та звітності. Ці системи дозволяють вводити та обробляти дані успішності, генерувати звіти, стежити за академічним прогресом студентів, використовувати аналітичні інструменти для оцінки та прогнозування успішності.

Позитивні сторони:

- Розширені можливості аналітики та звітності.
- Гнучкість налаштування під потреби конкретного навчального закладу.
- Автоматизована обробка та оновлення даних.

Негативні сторони:

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

– Висока вартість розробки, впровадження та підтримки програмного забезпечення.

– Необхідність навчання персоналу та користувачів системи.

– Вимоги до технічної інфраструктури та обслуговування.

База даних обліку успішності відіграє важливу роль у зборі, зберіганні та організації даних, пов'язаних з успішністю студентів. Вона служить центральним репозиторієм, де зберігаються інформація про оцінки, заліки, академічні досягнення, прогрес студентів та інші відомості, які відображають їхній академічний прогрес.

Роль бази даних обліку успішності включає наступні аспекти:

– Збір та організація даних: База даних забезпечує збір, систематизацію та організацію різноманітної інформації про успішність студентів. Це включає оцінки за кожен предмет, прогнозовані та актуальні результати, історію заліків та інші важливі дані.

– Моніторинг та аналітика: База даних дозволяє проводити моніторинг та аналіз успішності студентів. За допомогою різних запитів та звітів можна вивчати тренди, порівнювати результати студентів, виявляти сильні та слабкі сторони, а також здійснювати аналіз ефективності навчальних програм та методів навчання.

– Підтримка прийняття рішень: База даних надає основу для прийняття рішень на різних рівнях, починаючи від індивідуальних студентів до викладачів, адміністраторів та керівників навчальних закладів. За допомогою аналітичної інформації, яку надає база даних, можна здійснювати персоналізовані підходи до навчання, виявляти потреби студентів та підтримувати прийняття ефективних рішень для поліпшення їхньої успішності.

– Забезпечення доступу та обміну даними: База даних обліку успішності дозволяє забезпечити доступ до відповідної інформації студентам, викладачам, адміністраторам та іншим зацікавленим сторонам. Це сприяє зручному обміну даними, забезпечує однаковий доступ до актуальної

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

інформації та допомагає в реалізації електронних сервісів, таких як онлайн-журнали, електронні заявки та інші функції.

Загалом, база даних обліку успішності є важливим інструментом для збору, зберігання, аналізу та управління даними, пов'язаними з успішністю студентів. Вона допомагає вдосконалювати процеси навчання, покращувати якість освіти та забезпечувати ефективну підтримку навчальних програм та студентів.

Для дослідження предметної області будуть проаналізовані ці методики, підходи та програмні засоби з метою визначення їх переваг, недоліків та відповідності потребам конкретного навчального закладу. Після цього буде розроблена база даних обліку успішності студентів, що відповідає специфіці та вимогам навчального закладу, з урахуванням знайдених результатів дослідження.

## 1.2 Аналіз систем обліку успішності

Розглянемо існуючі автоматизовані системи, які забезпечують облік успішності. Microsoft Excel не є спеціалізованою системою обліку успішності студентів, але може бути використаний для цих цілей завдяки своїм функціям табличного процесора та можливостям обробки даних. Нижче наведено загальний опис того, як Microsoft Excel може бути використаний для обліку успішності студентів:

- Створення таблиць: Excel дозволяє створювати таблиці для збереження та організації даних студентів. Кожен студент може мати відповідний рядок, а кожен показник успішності (наприклад, оцінки за курси) може бути представлений у відповідних стовпцях.
- Розрахунки та формули: За допомогою формул Excel можна обчислювати різні показники успішності, наприклад середній бал, кількість

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

кредитів, зараховані курси і так далі. Функції, такі як AVERAGE, SUM, COUNT і IF, допомагають виконувати розрахунки на основі введених даних.

– Графіки та діаграми: Excel надає можливість побудови графіків та діаграм, що дозволяє візуально відобразити успішність студентів. Наприклад, графік може показувати розподіл оцінок або зміну успішності студента протягом семестру.

– Фільтри та сортування: Excel дозволяє використовувати фільтри та сортування для швидкого пошуку та відбору інформації. Наприклад, можна відфільтрувати дані за певним курсом або відсортувати студентів за алфавітом або успішністю.

– Шаблони та форматування: Excel надає широкий вибір готових шаблонів, які можна використовувати для обліку успішності студентів. Також можна використовувати форматування, щоб надати дані більш зрозумілий та професійний вигляд.

Важливо зазначити, що використання Excel для обліку успішності студентів може бути простим та ефективним рішенням для невеликих обсягів даних. Проте, для більших навчальних закладів, де велика кількість студентів та багато показників успішності, може бути бажаною спеціалізована система управління навчальним процесом, яка надає більше функцій та можливостей для аналізу даних студентів.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

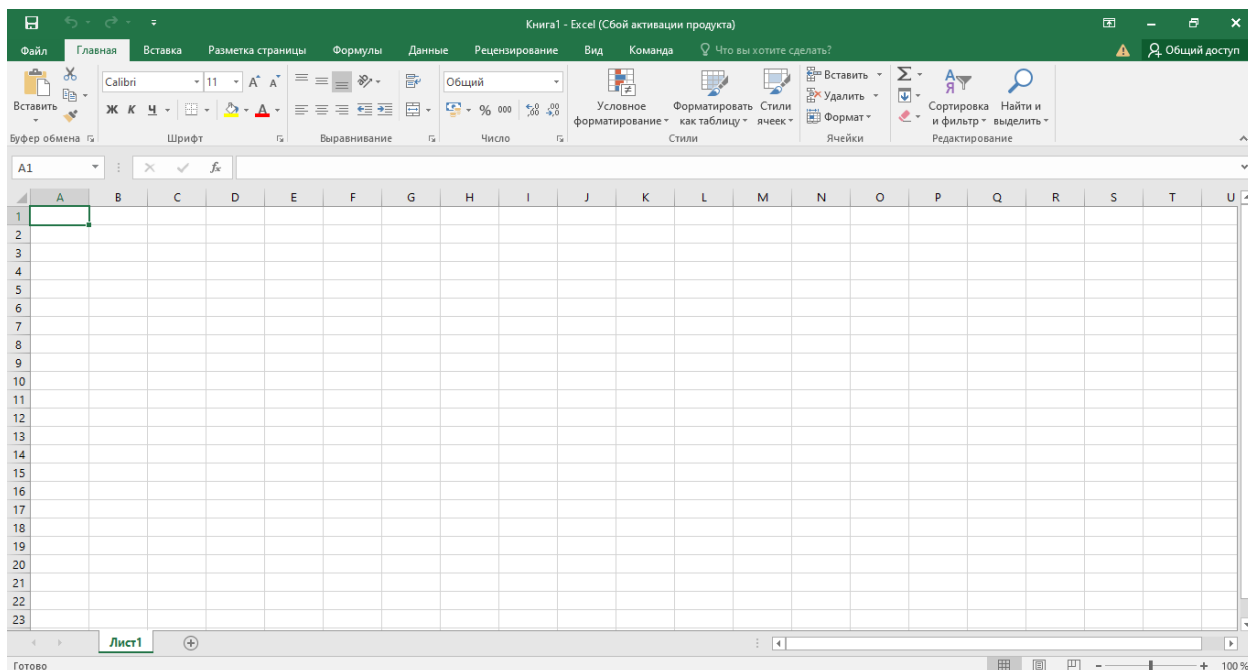


Рисунок 1.1 – Зовнішній вигляд Microsoft Excel

Вимоги до операційної системи (ОС) для Microsoft Excel можуть залежати від конкретної версії програмного забезпечення. Основні вимоги до ОС для останньої версії Microsoft Excel (Microsoft Excel 2021 або Office 365) включають наступне:

1. Windows: Microsoft Excel 2021 підтримує роботу на різних версіях операційних систем Windows, таких як Windows 10, Windows 8.1 та Windows 7 . Більш нові версії ОС можуть також підтримуватись.
2. macOS: Для користувачів macOS, Microsoft Excel 2021 підтримує версії macOS 10.14 (Mojave) або новіші.
3. Мінімальні апаратні вимоги: Мінімальні вимоги до апаратної частини залежать від конкретної версії Excel і можуть бути змінені розробником. Зазвичай, Microsoft рекомендує наявність процесора з тактовою частотою не менше 1 ГГц, ОЗП не менше 2 ГБ (32–бітна версія) або 4 ГБ (64–бітна версія) і мінімальний об'єм вільного простору на жорсткому диску.
4. Інтернет–підключення: Для отримання оновлень та використання деяких функцій, таких як спільне використання файлів в хмарі або використання веб–сервісів, може бути потрібне стабільне Інтернет–підключення.

Програмне забезпечення ISpro Student Lifecycle Management. ISpro SLM є комплексним програмним рішенням, розробленим для ефективного керування життєвим циклом студента та обліку їх успішності. Нижче наведено огляд та аналіз ключових функцій та можливостей ISpro SLM як системи обліку успішності:

– Внесення оцінок та рейтингів: ISpro SLM дозволяє вчителям та адміністраторам вносити оцінки та рейтинги студентів за різними предметами, курсами або модулями. Система підтримує різні шкали оцінювання та дозволяє зберігати дані про оцінки для подальшого аналізу.

– Розрахунок середнього балу та показників успішності: ISpro SLM автоматично розраховує середній бал та інші показники успішності студентів на основі внесених оцінок. Це включає середній бал за семестр, курс або весь період навчання. Результати можуть бути представлені у зручній формі, такій як графіки чи таблиці.

– Аналітика успішності: ISpro SLM надає розширені функції аналізу успішності студентів. Система дозволяє створювати звіти та графіки, які відображають академічний прогрес студентів, виявляють тенденції та допомагають виявити потенційні проблеми. Аналітичні інструменти допомагають вчителям та адміністраторам отримувати значиму інформацію щодо успішності та допомагають у прийнятті рішень.

– Система відстеження прогресу: ISpro SLM зберігає історію успішності студента, дозволяючи відстежувати його прогрес на протязі усього періоду навчання. Використовуючи цю інформацію, вчителі та адміністратори можуть проводити аналіз індивідуального розвитку студентів та забезпечувати необхідну підтримку.

– Сповіщення та комунікація: ISpro SLM має вбудовану систему сповіщень та комунікації, яка дозволяє надсилати повідомлення про академічну успішність студентів, включаючи оголошення про результати, відмінні досягнення та повідомлення щодо важливих подій. Це сприяє покращенню комунікації між студентами, вчителями та адміністрацією.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

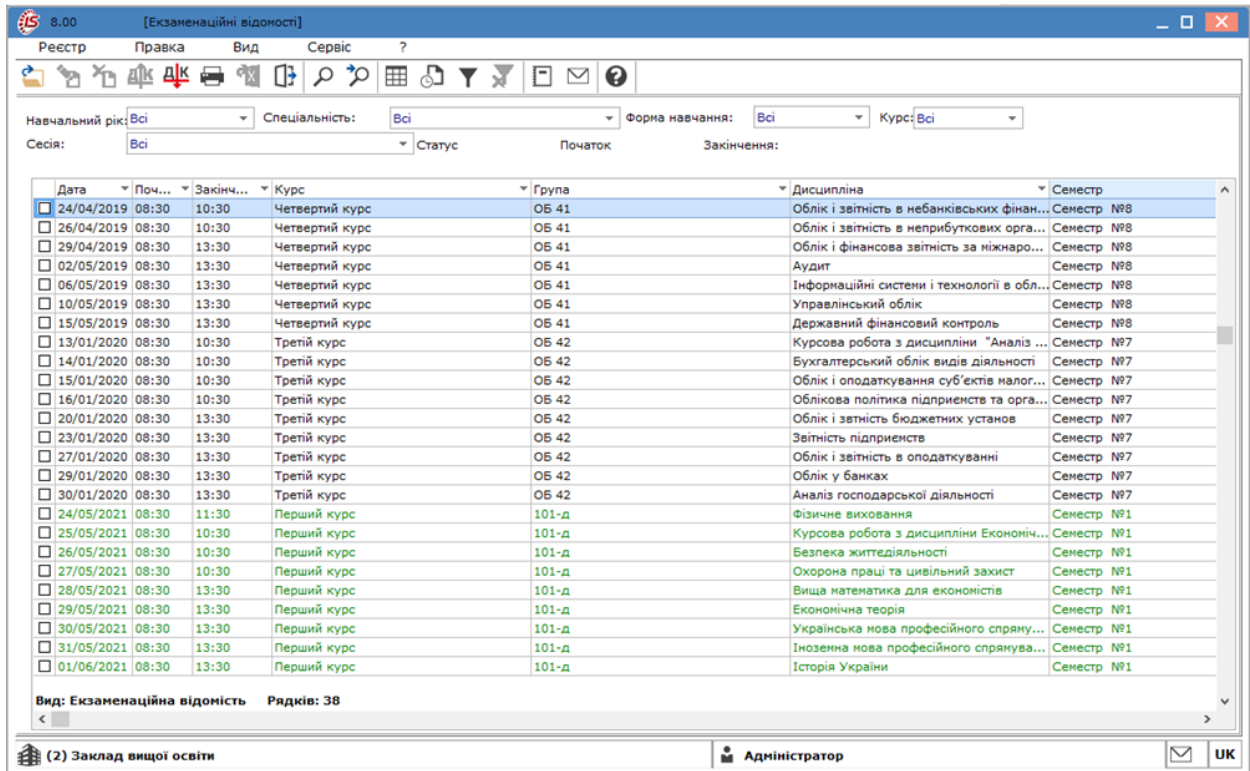


Рисунок 1.2 – Зовнішній вигляд ISpro Student Lifecycle Management

Оскільки конкретні вимоги до операційної системи (ОС) в програмі ISpro Student Lifecycle Management (ISpro SLM) можуть змінюватись в залежності від версії програмного забезпечення, основні вимоги до ОС можуть включати наступне:

1. Підтримувані операційні системи: ISpro SLM може підтримувати різні операційні системи, такі як Windows, macOS або Linux. Конкретні версії ОС, які підтримуються, будуть вказані в технічній документації або на офіційному веб-сайті розробника.
2. Версія ОС: ISpro SLM може вимагати певну версію операційної системи. Наприклад, вимога до Windows може бути Windows 10 або вище, а для macOS – macOS 10.14 або вище. Важливо перевірити вимоги до конкретної версії ОС, щоб забезпечити сумісність.
3. Апаратні вимоги: Поміж вимог до ОС також можуть бути апаратні вимоги, які стосуються процесора, оперативної пам'яті (ОЗП) та доступного простору на жорсткому диску. Наприклад, може бути вимога до процесора Intel



Core i5 або вище, ОЗП не менше 4 ГБ і наявність достатнього простору на жорсткому диску для встановлення програми.

Переваги та недоліки ISpro SLM відображені в таблиці 1.1

Таблиця 1.1 – Переваги та недоліки ISpro SLM

Переваги	Недоліки
Автоматизація процесів: Система автоматизує процеси внесення оцінок, розрахунку середнього балу, формування звітів та інших академічних документів, що полегшує роботу вчителів та адміністраторів.	Складність встановлення: Установка ISpro SLM може бути складною для користувачів без досвіду в налаштуванні та розгортанні програмного забезпечення.
Аналітика та звітність: ISpro SLM надає розширені можливості аналітики та звітності, що дозволяє адміністраторам та вчителям аналізувати дані про успішність студентів, ідентифікувати тенденції та приймати обґрунтовані рішення.	Обмеження функціональності: В деяких випадках ISpro SLM може мати обмежену функціональність порівняно з іншими системами обліку успішності.

<p>Система сповіщень та комунікації: ISpro SLM має вбудовану систему сповіщень та комунікації, яка дозволяє надсилати повідомлення про академічну успішність студентів, важливі оголошення та іншу інформацію, покращуючи комунікацію між студентами, вчителями та адміністрацією.</p>	<p>Відсутність інтеграції з іншими системами: ISpro SLM може мати обмежені можливості інтеграції з іншими системами, що може ускладнити обмін даними між ним та іншими системами, які використовуються в університеті або навчальному закладі.</p>
<p>Безпека даних: ISpro SLM забезпечує захист особистої та конфіденційної інформації студентів. Дані зберігаються в безпечному середовищі з контролем доступу та резервним копіюванням.</p>	<p>Витрати на ліцензії та підтримку: ISpro SLM комерційне програмне забезпечення, що вимагає придбання ліцензій та підписки на підтримку</p>
<p>Централізований облік: ISpro SLM надає централізовану систему обліку успішності студентів, що дозволяє зберігати та керувати всіма академічними даними на одній платформі.</p>	<p>Складність користування: Для нових користувачів ISpro SLM може бути складною в освоєнні через велику кількість функцій та складність інтерфейсу.</p>

В ході проведеного аналізу систем обліку успішності було вивчено різноманітні аспекти та характеристики існуючих рішень. Дослідження

включало оцінку функціональності систем, їх інтеграційні можливості, ефективність та продуктивність, ступінь налаштування та безпеку даних.

Оцінка функціональності систем дозволила з'ясувати, які конкретні функції та можливості вони надають для обліку та відстеження успішності. При цьому враховувалися потреби організації у зборі та аналізі даних, створенні звітів та показників успішності.

Ступінь налаштування систем також визначався в процесі аналізу. Було оцінено, наскільки гнучкі системи та чи задовольняють вони специфічні потреби організації.

Крім того, безпека даних була одним з основних аспектів, що проаналізовувалися. Було враховано рівень захисту даних у системах обліку успішності, включаючи контроль доступу та захист інформації від несанкціонованого доступу.

В результаті аналізу було отримано повну картину щодо сильних та слабких сторін існуючих систем обліку успішності.

### 1.3 Аналіз технологій автоматизації обліку успішності

Реляційна база даних є структурованою збіркою даних, організованих у вигляді таблиць або "реляцій". Вона заснована на математичній теорії реляційних моделей, яка була запропонована Едгаром Коддом у 1970 році. РБД є однією з найпоширеніших форм зберігання і управління даними, використовуваних у сучасній інформаційній технології.

Основними компонентами реляційної бази даних є таблиці, які представляють сутності або об'єкти, що взаємодіють в системі. Кожна таблиця складається з рядків (кортежів) і стовпців (атрибутів). Рядки таблиці відповідають окремим записам або екземплярам сутностей, тоді як стовпці визначають характеристики або властивості цих сутностей.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

Ключовим поняттям у реляційних базах даних є первинний ключ. Це унікальний ідентифікатор кожного запису в таблиці, який дозволяє однозначно ідентифікувати окрему сутність. Зв'язки між таблицями встановлюються за допомогою зовнішніх ключів, які посилаються на первинний ключ іншої таблиці. Це дозволяє здійснювати зв'язки та взаємодію між різними сутностями в базі даних.

Для керування даними в реляційній базі використовується мова структурованих запитів – SQL. SQL надає набір команд і операторів для створення, зміни, видалення і вибору даних з таблиць. Це дозволяє розробникам створювати складні запити до бази даних і отримувати потрібну інформацію.

Однією з переваг реляційних баз даних є їх гнучкість і стандартизованість. Вони дозволяють ефективно зберігати і керувати великими обсягами даних і забезпечують надійність та цілісність даних. Крім того, реляційні бази даних підтримують можливість використання транзакцій, які гарантують атомарність, консистентність, ізолюваність та стійкість даних (ACID властивості).

Однак, реляційні бази даних також мають певні обмеження. Вони можуть бути неефективними для деяких типів операцій, особливо тих, які вимагають складних зв'язків між даними. Крім того, при великому обсязі даних і складних зв'язків між таблицями можуть виникати проблеми з продуктивністю.

У підсумку, реляційна база даних є потужним інструментом для зберігання, організації і управління даними. Вона забезпечує структурований підхід до зберігання даних, дозволяє встановлювати зв'язки між ними і надає можливість ефективно виконувати запити до бази даних.

Ієрархічна база даних – це модель організації та зберігання даних, в якій дані представлені у вигляді ієрархічної структури або дерева. ІБД базується на ієрархічній моделі даних і була однією з перших моделей баз даних, що з'явилися в інформаційній технології.

У ієрархічній базі даних дані організовані у вигляді колекції записів, які зв'язані між собою за допомогою ієрархічних відношень батько–дитина. Кожен

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

запис містить дані та посилання на його батьківський запис. Таким чином, структура даних утворює ієрархію, де кожен запис може мати одного батька і багато дітей, але не може мати більше одного батька.

Ключовим поняттям у ієрархічній базі даних є кореневий запис, який є верхнім рівнем ієрархії. Він може мати кілька дочірніх записів, які в свою чергу можуть мати свої власні дочірні записи, і так далі. Кожен запис може мати свої атрибути, які зберігають дані.

Доступ до даних в ієрархічній базі даних здійснюється за допомогою механізму навігації по ієрархії. Запити формуються шляхом переходу від батьківських записів до дочірніх за допомогою спеціальних операторів, таких як "GET", "NEXT" або "GO TO". Операції вставки, видалення та оновлення даних також виконуються за допомогою цих операторів.

Однією з переваг ієрархічних баз даних є ефективність при роботі з даними, оскільки доступ до записів відбувається швидко та безпосередньо. Вони також добре підходять для представлення даних зі структурою батько–дитина, наприклад, для організаційних структур або файлових систем.

Однак, ієрархічні бази даних мають деякі обмеження. Вони важкі для модифікації та розширення, оскільки зміни в структурі даних можуть потребувати змін в багатьох записах. Крім того, ієрархічні бази даних не підтримують зв'язки між записами, що належать до різних гілок ієрархії.

У підсумку, ієрархічна база даних є моделлю зберігання даних, яка використовує ієрархічну структуру для організації та доступу до даних. Вона ефективна для представлення даних зі структурою батько–дитина, але може бути складною для модифікації.

Мережева база даних – це модель організації та зберігання даних, яка базується на мережевій моделі даних. Вона була розроблена в 1960–х роках як покращення ієрархічної моделі баз даних.

У мережевій базі даних дані організовані у вигляді колекції записів, які містять поля з даними та посилання на інші записи. Записи зв'язані між собою за допомогою зв'язків багато–до–багатьох, де кожен запис може мати багато

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

дочірніх записів і багато батьківських записів. Така структура створює мережу зв'язків між записами, де кожен запис може бути доступним з різних напрямків.

У мережевій базі даних використовується концепція "власник–підлеглий", де власний запис має дочірні записи, а підлеглий запис має посилання на батьківський запис. Доступ до даних здійснюється шляхом переходу від власника до підлеглого запису за допомогою спеціальних операторів навігації, таких як "FIND", "GO TO" або "GET".

Однією з переваг мережеских баз даних є їх гнучкість і здатність до виразного представлення зв'язків між даними. Вони можуть виражати складні зв'язки, такі як багато–до–багатьох, багато–до–одного та один–до–багатьох, що дозволяє моделювати різноманітні взаємозв'язки між сутностями.

Однак, мережеві бази даних також мають певні обмеження. Одне з основних обмежень – це складність побудови та керування структурою даних. При зміні структури даних необхідно актуалізувати всі посилання, що може бути складним та ресурсозатратним завданням. Крім того, зі зростанням обсягу даних та складності зв'язків, продуктивність може страждати.

У підсумку, мережева база даних є моделлю зберігання даних, яка використовує мережеву структуру для організації та доступу до даних. Вона забезпечує гнучкість та можливість виразного моделювання зв'язків між даними, але може бути складною у побудові та керуванні структурою даних.

#### 1.4 Постановка задач кваліфікаційної роботи

Сьогодні все більше уваги приділяють системам обліку успішності студентів, які відповідають вимогам доступності та відкритості освітньої системи. При цьому найважливішу роль відіграє методика інтеграції таких систем у повсякденну навчальну роботу навчальних закладів. Впровадження у навчальних закладах автоматизованих систем є вимогою часу та суттєво спрощує роботу у певних напрямках.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

Проаналізувавши опис проблеми було вияснено що у сучасних вищих навчальних закладах важливо мати систему обліку та аналізу успішності студентів, яка би сприяла збору та зберіганню даних про академічні досягнення, а також надала можливість ефективного аналізу цих даних. Наразі багато навчальних закладів використовують різні системи, але часто вони не є повноцінними базами даних, не інтегруються між собою та не забезпечують достатньої функціональності для аналізу успішності студентів. Однак, варто зазначити, що впровадження нових систем вимагає фінансових і технічних зусиль.

Результати проведених досліджень свідчать про те, що більшість вищих навчальних закладів успішно автоматизували свої системи обліку успішності студентів. Однак, ситуація в загальноосвітніх школах вимагає ще багато роботи, оскільки багато з них знаходяться на етапі впровадження та вдосконалення застарілих інформаційно–програмних систем.

Основною метою цієї кваліфікаційної роботи є розробка бази даних обліку успішності студентів, яка б забезпечувала централізоване збирання, зберігання та аналіз даних про навчальні досягнення студентів. Крім того, робота має на меті розробку інтерфейсу користувача, який буде зручним та легким у використанні, забезпечуватиме широкий спектр функціональних можливостей для аналізу даних та надаватиме потужні інструменти для здійснення різноманітних статистичних операцій та звітності.

Для досягнення вищезазначених цілей, в рамках даної кваліфікаційної роботи будуть виконані наступні завдання:

- Аналіз існуючих систем обліку успішності студентів в навчальних закладах та виявлення їх недоліків та обмежень.
- Визначення необхідного функціоналу та вимог до бази даних обліку успішності студентів.
- Проектування структури бази даних, включаючи визначення таблиць, полів та зв'язків між ними.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

- Розробка інтерфейсу користувача для зручного та ефективного доступу до бази даних та її функціоналу.
- Тестування та валідація розробленої бази даних шляхом проведення різних сценаріїв тестування та перевірки коректності результатів.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28



## 2 АЛГОРИТМИ ПРОЕКТУВАННЯ БАЗИ ДАНИХ ОБЛІКУ УСПІШНОСТІ

### 2.1 Програмні інструменти організації обліку успішності

Існує багато програмних інструментів для організації баз даних обліку успішності. Ось декілька популярних інструментів, які можуть бути використані для цієї мети:

1. MySQL: MySQL є однією з найпоширеніших реляційних систем керування базами даних (РСКБД). Він забезпечує широкі можливості для створення та управління базами даних, включаючи можливості для зберігання, організації та обробки даних обліку успішності.

2. PostgreSQL: PostgreSQL є іншим потужним РСКБД, який надає розширені функції для організації баз даних обліку успішності. Він має вбудовану підтримку для географічних об'єктів, розширюваність та розширені можливості управління даними.

3. Microsoft SQL Server: Microsoft SQL Server є реляційною системою керування базами даних, розробленою компанією Microsoft. Він має широкий набір інструментів для організації та аналізу баз даних обліку успішності, включаючи різноманітні можливості звітності та аналітики.

4. Oracle Дата зверненняbase: Oracle Дата зверненняbase є однією з найпотужніших та розповсюджених РСКБД у світі. Вона надає розширені функції для управління та аналізу даних обліку успішності, включаючи можливості для складних запитів та аналітики даних.

5. MongoDB: MongoDB є документ-орієнтованою системою керування базами даних, яка добре підходить для зберігання та організації даних обліку успішності у веб-додатках та інших проектах з великим обсягом неструктурованих даних.

Було обрано MySQL для організації бази даних обліку успішності з таких причин:

– Надійність та стабільність: MySQL є дуже надійною та стабільною реляційною системою керування базами даних. Вона має довгу історію

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

використання в багатьох веб-додатках та проектах, що свідчить про її надійність.

– Велике співтовариство користувачів: MySQL має широке співтовариство користувачів, що означає, що можна знайти багато ресурсів, документації та підтримки. Це спрощує вивчення та роботу з цією системою керування базами даних.

– Легкість використання: MySQL має простий у використанні синтаксис SQL, що полегшує розробку та виконання запитів. Вона також має інтуїтивно зрозумілий інтерфейс для адміністрування та керування базою даних.

– Широкий функціонал: MySQL надає розширені можливості для роботи з даними, включаючи можливості для індексування, транзакційності, забезпечення безпеки даних, а також підтримку реплікації та масштабованості.

– Відкрите програмне забезпечення: MySQL є відкритим програмним забезпеченням, що означає, що ви можете використовувати його безкоштовно та змінювати вихідний код відповідно до своїх потреб. Відкритість дозволяє також використовувати різноманітні сторонні інструменти та розширення.

Ці фактори сприяють вибору MySQL для організації бази даних обліку успішності, а також забезпечують комфортну та надійну роботу з даними.

Для управління базами даних існує багато інтерфейсів, які надають зручність та функціональність при роботі з даними. Ось деякі з них:

1. phpMyAdmin: Це популярний веб-інтерфейс для керування базами даних MySQL. Він дозволяє створювати, редагувати та видаляти таблиці, виконувати SQL-запити, керувати користувачами та привілегіями, імпортувати та експортувати дані.

2. MySQL Workbench: Це інтегроване середовище розробки (IDE) для баз даних MySQL. Воно надає графічний інтерфейс для моделювання баз даних, розробки SQL-запитів, адміністрування баз даних та багато іншого.

3. Navicat: Це комерційний інтерфейс для керування різними базами даних, такими як MySQL, PostgreSQL, Oracle, SQLite та інші. Navicat надає

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

широкий спектр функцій, включаючи розширений редактор SQL, візуальне моделювання баз даних, імпорт/експорт даних та інструменти для адміністрування.

4. DBeaver: Це універсальний безкоштовний інтерфейс для керування різними базами даних, включаючи MySQL, PostgreSQL, Oracle, SQL Server та багато інших. DBeaver надає широкі можливості для виконання SQL-запитів, керування структурою баз даних та адміністрування.

5. pgAdmin: Це інтерфейс для керування базами даних PostgreSQL. Він має інтуїтивний інтерфейс, який дозволяє адмініструвати бази даних, створювати запити, керувати таблицями та іншими об'єктами.

Було вибрано phpMyAdmin, тому що він є одним з найпопулярніших інструментів для управління базами даних, і він має кілька переваг порівняно з іншими аналогами:

1. Зручний веб-інтерфейс: це найбільше мене зацікавило, phpMyAdmin забезпечує легкий доступ до баз даних через веб-браузер без необхідності встановлення додаткового програмного забезпечення. Це дозволяє зручно працювати з базами даних з будь-якого комп'ютера з підключенням до Інтернету.

2. Безкоштовність та відкритий код: phpMyAdmin є безкоштовним програмним забезпеченням з відкритим кодом, що дозволяє користувачам використовувати його безкоштовно та адаптувати до своїх потреб.

3. Підтримка різних систем управління базами даних: phpMyAdmin підтримує не тільки MySQL, але і деякі інші системи управління базами даних, такі як MariaDB та Drizzle. Це дає можливість працювати з різними системами управління базами даних, використовуючи один інтерфейс.

4. Розширені можливості: phpMyAdmin надає багато функцій для управління базами даних, таких як створення таблиць, редагування даних, виконання SQL-запитів, керування користувачами та привілегіями, імпорт та експорт даних тощо. Інтерфейс phpMyAdmin також має розширені можливості для візуалізації та аналізу даних.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

Активна спільнота та оновлення: phpMyAdmin має велику спільноту користувачів та розробників, що сприяє швидкому виявленню та виправленню помилок, а також вдосконаленню функціональності. Регулярні оновлення забезпечують покращення безпеки та нові можливості.

## 2.2 Структура бази даних

Проектування об'єктів бази даних є важливим етапом в розробці системи обліку успішності студентів. На цьому етапі визначається структура, таблиці, індекси, представлення та функції, необхідні для збереження та опрацювання даних.

Основні елементи проектування об'єктів бази даних включають:

1. **Таблиці:** Визначаються таблиці, які використовуються для збереження різних типів даних, таких як інформація про студентів, оцінки тощо. Кожна таблиця має свою структуру, включаючи поля (колонки) та типи даних, що відображають різні атрибути студентів та їх успішності.

2. **Індекси:** Встановлюються індекси для покращення швидкості пошуку та доступу до даних в базі даних. Індекси можуть бути створені на певні поля таблиць, що дозволяє прискорити пошук і фільтрацію даних.

3. **Представлення:** Визначаються представлення, які надають зручний та зрозумілий доступ до даних в базі даних. Це можуть бути запити (запити SQL), які використовуються для отримання конкретної інформації з бази даних, або представлення результатів запитів у зручному форматі.

Сама база даних складається з двох таблиць users де зберігаються дані для входу і task де зберігаються дані про студента (прізвище, ім'я) і його оцінку, а також час і дату коли була виставлена оцінка.

Таблиця 2.1 – Структура бази даних

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

База даних datapr
Дані для входу
Список студентів
Група
Спеціальність
Кафедра
Факультет

Таблиця 2.2 – Структура таблиці де зберігаються дані для входу

Дані для входу
ID
Логін
Пароль
Сіль
Роль

Таблиця 2.3 – Структура таблиці список студентів

Список студентів
ID
ФІО студента
Оцінка
Час виставлення
ID групи

Таблиця 2.4 – Структура таблиці група

Група
ID
Назва групи
ID спеціальності

Таблиця 2.5 – Структура таблиці спеціальність

Спеціальність
ID
Назва спеціальності
ID кафедри

Таблиця 2.6 – Структура таблиці кафедри

Кафедра
ID
Назва кафедри
ID факультета

Таблиця 2.7 – Структура таблиці факультет

Факультет
ID
Назва факультету

В першу чергу, відбувається підключення до бази даних це є необхідним кроком для подальшої взаємодії з нею.

На рисунку 2.1 зображено код для підключення до бази даних.

```

db = QSqlDatabase::addDatabase("QMYSQL", "MyConnection");
db.setUser("root");
db.setDatabaseName("datart");
db.setHostName("localhost");
db.setPassword("usbw");
db.setPort(3306);

```

Рисунок 2.1 – Підключення до бази даних

На рисунку 2.2 представлений код та SQL-запит для реєстрації користувача, де дані (логін, пароль, а також роль користувача – студент або викладач) додаються до бази даних. Особливість цього процесу полягає у хешуванні пароля користувача з використанням алгоритму SHA256 з сіллю (salt). Сіль є додатковим випадковим значенням, яке додається до пароля перед хешуванням. Це забезпечує додатковий рівень безпеки, оскільки випадкова сіль унеможливує пряме відтворення паролів навіть при використанні одного й того ж пароля для різних користувачів. Сіль зберігається в базі даних разом з хешованим паролем для подальшої авторизації та порівняння з введеним користувачем під час процесу авторизації.

```

QSqlQuery query(db);
query.prepare("INSERT INTO users (login, password, salt, role)"
             "VALUES (:login, :password, :salt, :role)");
query.bindValue(":login", login);
query.bindValue(":password", hashedPassword);
query.bindValue(":salt", salt);

```

Рисунок 2.2 – Реєстрація користувача

Процес авторизації включає наступні кроки:

1. Користувач вводить свій логін та пароль на формі авторизації.
2. Система отримує введені дані і виконує запит до бази даних для отримання солі та хешованого пароля, пов'язаного з введеним логіном (рис. 2.7).
3. Отримавши сіль з бази даних, система об'єднує її з введеним паролем користувача (рис. 2.4).
4. Пароль разом із сіллю піддається хешуванню за допомогою криптографічного алгоритму SHA256 (рис. 2.4).

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

5. Отриманий хешований пароль порівнюється зі збереженим хешованим паролем, що знаходиться в базі даних для введеного логіну (рис. 2.9).

6. Якщо хешовані паролі співпадають, система вважає користувача авторизованим і надає йому доступ до відповідних ресурсів або функціональності.

7. У разі невідповідності хешованих паролів, система відхиляє авторизацію і повідомляє користувачеві про невірні облікові дані.

Хешування пароля забезпечує безпеку, оскільки хешований пароль неможливо розшифрувати знову в початковий пароль без знання використаного алгоритму та солі. Використання солі (рандомної послідовності даних) додає додатковий рівень безпеки, оскільки вона перешкоджає використанню заздалегідь створених радужних таблиць (rainbow tables) для швидкого підбору паролів.

На рисунку 2.3 зображений SQL-запит, який використовується для вибору пароля та солі користувача з бази даних за його логіном. Запит дозволяє отримати з бази даних збережений хешований пароль та відповідну сіль для подальшого використання під час процесу авторизації користувача. Цей запит допомагає забезпечити безпеку паролів та відповідність введеного пароля збереженому в базі даних.

```
QSqlQuery query = QSqlQuery(db);
query.prepare("SELECT password, salt FROM users WHERE login = :login");
query.bindValue(":login", login);
```

Рисунок 2.3 – Вибір хешованого пароля і солі за логіном

На рисунку 2.4 зображено хешування введеного користувачем пароля, для подальшої перевірки



```

QString savedSalt = query.value("salt").toString();

// Шифрування введеного пароля зі збереженою сіллю
QString hashedUserInputPassword = hashPassword(password, savedSalt);

```

Рисунок 2.4 – Хешування пароля

На рисунку 2.5 зображено отриманий хешований пароль який порівнюється зі збереженим хешованим паролем, що знаходиться в базі даних

```

QString savedHashedPassword = query.value("password").toString();
QString hashedUserInputPassword = hashPassword(password, savedSalt);
if (hashedUserInputPassword == savedHashedPassword) {

```

Рисунок 2.5 – Порівняння паролів

На рисунку 2.6 зображено код і sql запит для внесення даних в БД даних про студента і його оцінку

```

QString nameStudent = ui->lineEnglish->text();
QString rating = ui->lineUkr->text();
QString query = QSqlQuery(QSqlDatabase::database("MyConnection"));
query.prepare("INSERT INTO task (nameStudent, rating)"
              "VALUES (:nameStudent, :rating)");
//query.bindValue(":path", pathSave);
query.bindValue(":nameStudent", nameStudent);
query.bindValue(":rating", rating);

```

Рисунок 2.6 – Внесення даних в базу даних

Найосновніші вимоги для надійної роботи ПЗ є те, що усі файли, які використовуються даним програмним продуктом, повинні знаходитися в одному каталозі або папці і категорично забороняється їх редагування, тобто будь-які зміни у їх назві і розширенні. Це може призвести до втрати працезданості програми або її некоректної роботи. У системних файлах програми зберігається інформація про стилі шрифтів

Для коректної роботи системи потрібний сервер, який забезпечує хостинг та обробку веб-додатків. Сервер може бути представлений різними програмними рішеннями, такими як USBWebserver, ХАМРР, WAMP, LAMP та інші. Ці сервери включають в себе комплект необхідного програмного

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

забезпечення, таке як веб-сервер Apache, база даних MySQL. Програма розроблена буде сумісна лише з операційною системою Windows. Це означає, що програма буде призначена для використання на комп'ютерах або серверах, які працюють під управлінням операційних систем Windows, таких як Windows 10, Windows 8, Windows 7 або Windows Server.

Програма немає жодних обмежень на апаратне забезпечення. Завдяки своїй легкості та простоті, вона може працювати на різних типах комп'ютерів і серверів. Вона не вимагає великої потужності обчислювальної системи або великого обсягу оперативної пам'яті. Це дозволяє користувачам використовувати програму навіть на більш застарілих або менш потужних комп'ютерах. Таким чином, програма є доступною і працездатною на різних пристроях, що робить її більш універсальною для використання в різних середовищах.

### 2.3 Алгоритм проектування бази даних обліку успішності

У сфері оцінювання академічної продуктивності студентів, існує значна різниця між навчальними закладами. Багато факторів впливають на цей рівень: місцезнаходження університету, рівень фінансування, доступність залучення зовнішніх інвестицій та кількість студентів, серед інших. Крім того, стан інформаційних ресурсів та рівень їхньої інтеграції напряму залежать від цих чинників. На жаль, більшість вищих навчальних закладів не мають достатніх ресурсів для ефективної автоматизації систем обліку студентської успішності на відповідному рівні. Проте, впровадження передових технологій шляхом імплементації автоматизованих інформаційних систем для обліку успішності студентів є одним з прогресивних рішень у цій сфері. Це передбачає активну участь спеціалістів у сфері освіти у процесі освоєння комп'ютерних навичок та роботи зі спеціалізованим програмним забезпеченням.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

Впровадження систем для обліку успішності студентів має значний вплив на розвиток освітньої сфери. Одним з ключових результатів такого впровадження є забезпечення доступу до інформації на вищому рівні. Це досягається завдяки створенню різноманітних інформаційних продуктів, що максимально полегшують роботу з обліком успішності студентів. Декілька прикладів цих продуктів включають:

– Розробка інтегрованої бази даних для зберігання та управління інформацією про успішність студентів, включаючи дані про оцінки, академічні досягнення, курси та інші релевантні дані.

– Створення системи аналітики даних, яка дозволяє проводити комплексний аналіз успішності студентів на основі зібраних в базі даних даних. Це допомагає виявляти тенденції, робити прогнози та розробляти стратегії для покращення навчального процесу.

– Розробка інтерфейсу, що забезпечує доступ до бази даних успішності студентів для викладачів та адміністраторів, щоб вони могли відстежувати та оцінювати прогрес студентів, складати звіти та здійснювати необхідні дії.

– Розвиток системи автоматичного оновлення бази даних успішності студентів на основі отриманих даних з інших джерел, таких як оцінки тестів, проекти та інші академічні роботи.

– Впровадження системи забезпечення безпеки та конфіденційності бази даних успішності студентів, включаючи резервне копіювання, ролеву авторизацію та шифрування даних.

Впровадження сучасних систем для обліку успішності студентів покращує якість освіти, сприяє зручному моніторингу академічних досягнень і підвищує доступність необхідної інформації для всіх учасників навчального процесу. Це стимулює розвиток освітньої системи та сприяє підвищенню якості навчання та досягненню кращих результатів у сфері освіти.

Сьогоднішні навчальні заклади все більше використовують спеціалізоване програмне забезпечення для точного обліку та моніторингу

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

успішності студентів. Це програмне забезпечення забезпечує автоматизовану систему для збору, зберігання та аналізу різноманітних даних, пов'язаних з академічними досягненнями студентів.

Замість стандартного сайту, створюється комплексне програмне забезпечення, яке забезпечує зручний та безперервний доступ до інформації про успішність студентів, незалежно від місця та часу. За допомогою цього програмного забезпечення, студенти можуть швидко переглядати свої академічні дані.

Програмне забезпечення для обліку успішності студентів також дозволяє викладачам та адміністраторам здійснювати ефективне управління та моніторинг академічного прогресу студентів. Вони можуть швидко оцінювати роботу студентів, створювати звіти та надавати зворотний зв'язок. За допомогою цього програмного забезпечення, всі відповідні дані зберігаються в централізованій базі даних, що спрощує процеси збору, оновлення та аналізу інформації.

Таке програмне забезпечення не тільки полегшує процеси обліку успішності студентів, але і сприяє покращенню комунікації між студентами, викладачами та адміністрацією. Вони можуть легко обмінюватися необхідною інформацією та співпрацювати над вирішенням академічних питань. Використання програмного забезпечення для обліку успішності студентів сприяє ефективному управлінню навчальним процесом та сприяє досягненню високих результатів.

Це програмне забезпечення розроблене з метою полегшити облік та аналіз успішності студентів за допомогою централізованої системи керування даними. Воно дозволяє зберігати, оновлювати та відстежувати різноманітні показники успішності, допомагаючи студентам, педагогам та адміністрації швидко отримувати актуальну інформацію.

Крім того, це програмне забезпечення розроблене з урахуванням зручності користувача. Інтуїтивний і дружній інтерфейс дозволяє легко навігуватися по системі, швидко знаходити необхідну інформацію та проводити аналіз результатів. Воно створено з метою забезпечення ефективності та

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

зручності використання, сприяючи покращенню якості освіти та навчального процесу.

Основою будь-якої системи обліку успішності студентів є централізована база даних. При розробці такої бази даних, необхідно провести детальний аналіз потоків інформації, які входять та виходять з системи, а також вивчити форми подання та частоту обробки відповідних документів.

Під час розробки бази даних для системи обліку успішності студентів, проводиться глибокий аналіз сфери застосування. Це включає вивчення основних процесів, що відбуваються в освітньому закладі, визначення інформаційних потреб викладачів та адміністрації, а також аналіз форм вхідних та вихідних інформації.

У базі даних для системи обліку успішності студентів необхідно врахувати різноманітні види вхідної інформації, зокрема:

- дані про студентів;
- дані про групу;
- дані про спеціальність;
- дані про факультет;
- дані про оцінки студентів;
- інформацію про студентів.

БД повинна відповідати таким вимогам:

1. БД «Обліку успішності» призначена для введення, зберігання і обробки інформації про студента, оцінку, спеціальність.

2. БД «Обліку успішності» повинна забезпечити виконання таких дій:

- реєстрація користувачів;
- виставлення оцінки.

Один із основних кроків у розробленні БД обліку успішності студентів – це розробка контекстної діаграми, що надає візуальне уявлення про процеси, які відбуваються в системі обліку успішності студентів. Ця діаграма відображає інформаційні компоненти, які необхідні для збору, обробки та аналізу даних щодо успішності студентів.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

При розробці контекстної діаграми використовується принцип ієрархічного структурування, що сприяє організації складових елементів системи. Елементами діаграми є функції системи, такі як вхідні та вихідні дані, механізм функціонування системи та керуюча інформація. Ця модель є основою для розробки програмно-інформаційної системи.

Важливо також враховувати взаємодію інформаційної системи обліку успішності студентів з зовнішнім середовищем. Це взаємодія може бути відображена на відповідному графіку, де видно, як інформаційна система взаємодіє з іншими системами або зовнішніми агентами з метою отримання додаткової інформації або передачі результатів аналізу успішності студентів.

На рисунку 2.7 зображено контекстну діаграму.

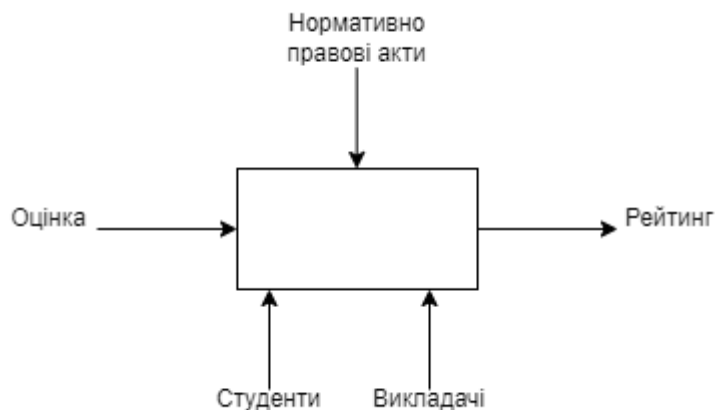


Рисунок 2.7 – Контекстна діаграма

Після створення контекстної діаграми наступним кроком є розробка моделей бази даних, включаючи логічну модель, ER діаграму, фізичну модель. Ця модель відображає структуру даних і допомагає забезпечити ефективне зберігання та обробку інформації. Вона включає у себе таблиці, які представляють різні сутності та їх взаємозв'язки, що забезпечують зв'язаність та цілісність даних.

На рисунку 2.8 зображено блок схему організації баз даних

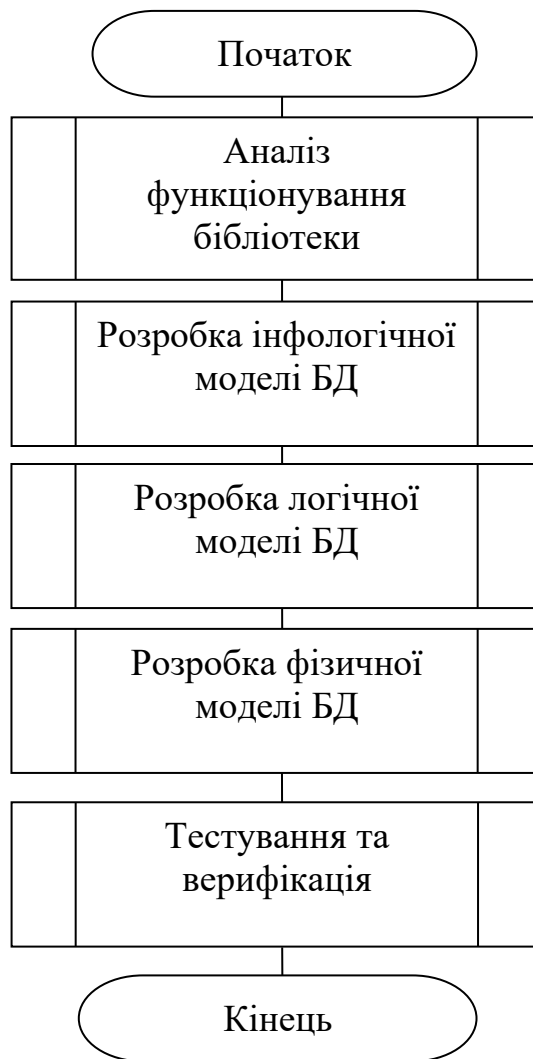


Рисунок 2.8 – Блок схема

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

### 3.1 Інфологічна та логічна моделі баз даних

Логічна модель бази даних визначає структуру та зв'язки між даними, незалежно від конкретної системи управління базами даних чи фізичної реалізації. Вона є абстрактним рівнем, де встановлюються сутності, атрибути та їх взаємозв'язки.

Таблиця 3.1 – Характеристика атрибутів сутностей

Сутність	Первинний ключ	Атрибути	Тип даних
Список користувачів	ID	Логін	Текстовий
		Пароль	Текстовий
		Сіль	Текстовий
		Роль	Текстовий
Список студентів	Номер студента	ФІО студента	Текстовий
		Оцінка	Числовий
		Час виставлення оцінки	Дата і час
		Номер групи	Числовий
Група	Номер групи	Група	Текстовий
		Номер спеціальності	Числовий



Продовження таблиці 3.1

Спеціальність	Номер спеціальності	Назва спеціальності Номер кафедри	Текстовий Числовий
Кафедра	Номер кафедри	Назва кафедри Номер факультета	Текстовий Числовий
Факультет	Номер факультета	Назва факультета	Текстовий

ER–діаграма є графічним засобом моделювання бази даних, який використовується для візуалізації структури даних та взаємозв'язків між ними. Вона використовується для розробки моделі бази даних, що відображає концептуальне розуміння системи.

У ER–діаграмі бази даних можна зобразити різні компоненти, включаючи сутності, атрибути та взаємозв'язки між ними.

Сутності представляють реальні або концептуальні об'єкти, про які збираються дані. Наприклад, у системі університету можуть бути сутності "студент", "викладач" та "предмет". Кожна сутність має унікальний ідентифікатор, який використовується для ідентифікації цієї сутності.

Атрибути визначають характеристики або властивості сутностей. Наприклад, для сутності "студент" можуть бути атрибути, такі як "ім'я", "прізвище" та "номер студентського квитка". Атрибути допомагають деталізувати інформацію про сутності.

Взаємозв'язки показують зв'язки між різними сутностями. Наприклад, університетська система може мати взаємозв'язок "студент вчиться на предметі", який з'єднує сутності "студент" та "предмет". Взаємозв'язки можуть мати кардинальність, яка визначає кількість зв'язків між сутностями.

ER – діаграма дозволяє вам візуалізувати структуру бази даних та зрозуміти взаємозв'язки між сутностями. Вона є важливим інструментом для розробки і розуміння моделі бази даних та допомагає забезпечити правильне проектування та розвиток бази даних.

На рисунку 3.1 зображено зв'язок багато до багатьох

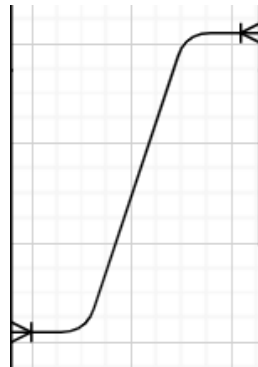


Рисунок 3.1 – Зв'язок багато до багатьох

На рисунку 3.2 зображено зв'язок один до багатьох

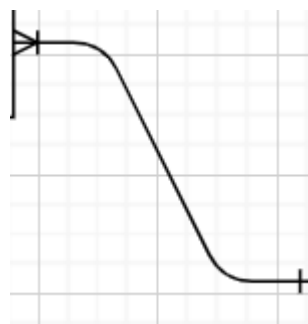


Рисунок 3.2 – Зв'язок багато до одного

На рисунку 3.3 відображено зв'язок один до одного

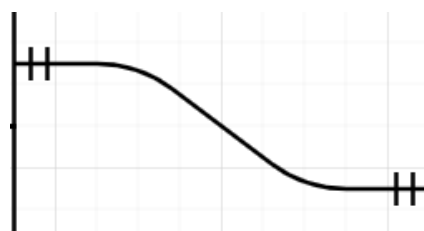


Рисунок 3.3 – Зв'язок один до одного

На рисунку 3.4 зображено ER діаграму обліку успішності.

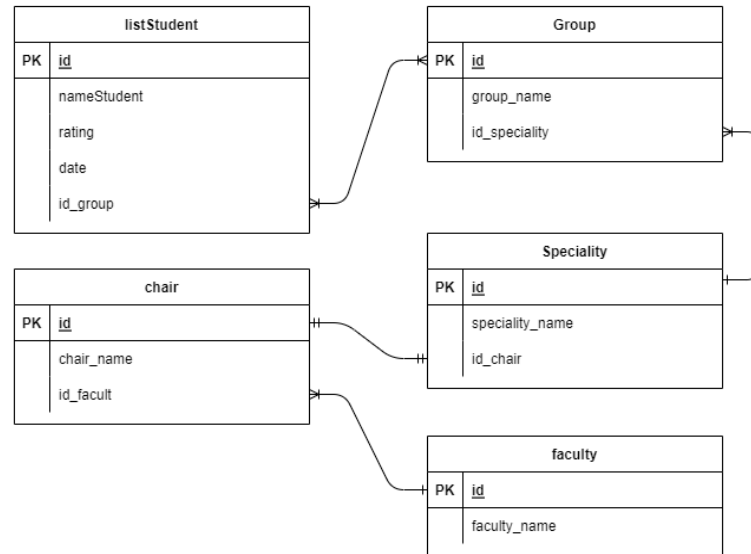


Рисунок 3.4 – ER діаграма

На рисунку 3.5 зображено логічна модель даних обліку успішності

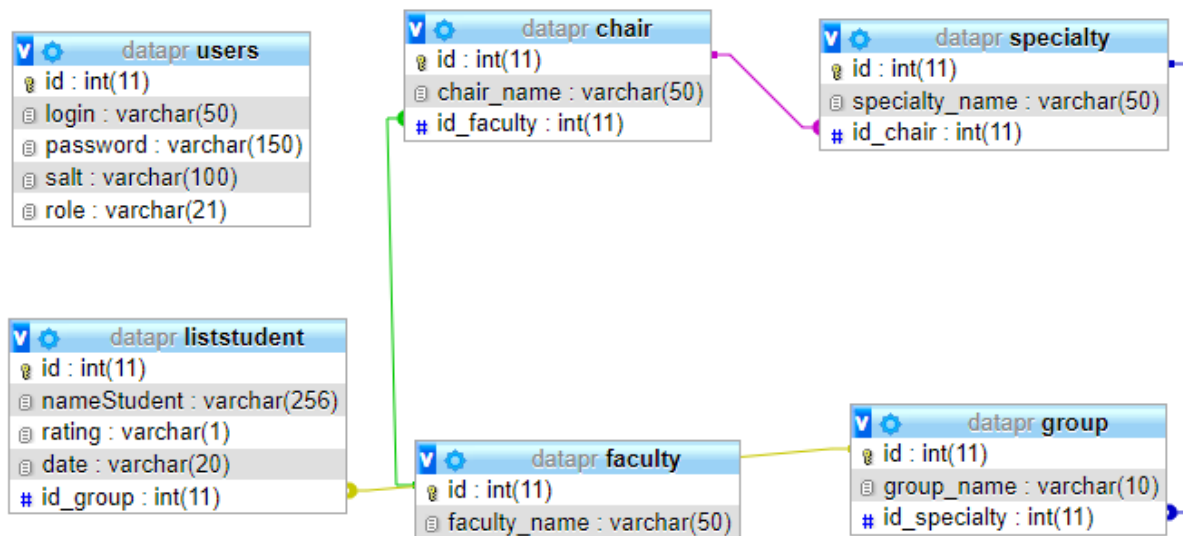


Рисунок 3.5 – Логічна модель даних

Інформаційна модель (інфологічна модель) є ключовим поняттям в області інформаційних наук і комп'ютерних систем. Вона використовується для опису та представлення структури, функцій і взаємодії компонентів системи або процесу збору, обробки, зберігання та передачі інформації.

Інформаційна модель може бути представлена в графічному або текстовому вигляді і складається з набору елементів, які відображають ключові поняття, взаємозв'язки та властивості системи або процесу. Така модель допомагає визначити, які типи даних, функції, обробки та взаємодії виконуються в системі, а також які є вимоги до цих компонентів.

Інформаційна модель може бути побудована на різних рівнях абстракції, починаючи від високорівневих концепційної моделі, яка описує загальні характеристики системи, до деталізованої моделі, що включає деталі реалізації компонентів та взаємодії між ними.

В інформаційній моделі можуть бути використані такі елементи:

– Сутності (entities): відображають основні об'єкти або поняття, що представляють дані в системі. Вони мають атрибути, які описують характеристики цих об'єктів.

– Атрибути (attributes): відображають властивості сутностей і описують їх характеристики. Наприклад, для сутності "Студент" атрибутами можуть бути "ім'я", "оцінка".

– Відношення (relationships): відображають зв'язки між сутностями. Наприклад, відношення "Студент належить до Групи" вказує на те, що кожен студент може належати до однієї групи.

– Атрибути відношень (relationship attributes): відображають характеристики або властивості відношень між сутностями. Наприклад, атрибутом відношення "Студент належить до Групи" може бути "рейтинг студента".

– Схема бази даних (database schema): визначає структуру та організацію бази даних, включаючи сутності, атрибути, відношення та обмеження цих компонентів.

– Діаграми (diagrams): графічне представлення інформаційної моделі, такі як діаграми потоку даних, діаграми сутностей–відношень та діаграми класів.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

Описи (descriptions): текстовий опис структури та функцій системи або процесу, який доповнює графічне представлення.

### 3.2 Фізична модель баз даних

Фізична модель бази даних є конкретною реалізацією бази даних і включає компоненти, що визначають спосіб зберігання та доступу до даних.

Одним з основних компонентів фізичної моделі бази даних є таблиці. Таблиці визначають структуру даних і складаються з рядків і стовпців. Рядки представляють окремі записи або кортежі, а стовпці визначають атрибути або поля даних.

Кожен стовпець таблиці має властивості, такі як ім'я і тип даних, що визначає формат даних, що зберігаються в цьому стовпці. Типи даних можуть бути числовими, рядковими, датами, булевими тощо. Стовпці також можуть мати обмеження, наприклад, вимагати заповнення або забороняти дублювання значень.

Інший важливий компонент – ключі. Головний ключ (Primary Key) є унікальним ідентифікатором для кожного запису в таблиці і гарантує їх унікальність. Зовнішні ключі (Foreign Key) використовуються для встановлення зв'язків між таблицями, вказуючи на ключове поле іншої таблиці.

Ще один компонент – індекси. Індекси створюються для поліпшення швидкодії пошуку та сортування даних. Вони створюються для одного або кількох стовпців таблиці і дозволяють ефективніше здійснювати пошук даних за певними критеріями.

Додатково, фізична модель бази даних включає інші компоненти, такі як індексні структури (В-дерева, хеш-таблиці тощо), які використовуються для ефективного збереження та доступу до даних, а також параметри зберігання, які визначають спосіб фізичного збереження даних на диску (файли, розташування тощо).

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

Фізична модель бази даних враховує специфічні особливості конкретної СУБД і визначає, як дані фактично зберігаються і доступні для операцій, таких як пошук, вставка, оновлення та видалення.

На рисунку 3.6 зображено базу даних з її таблицями

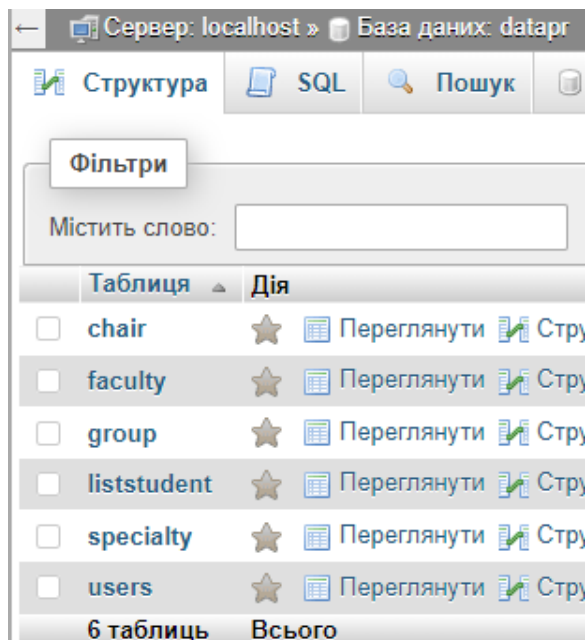


Рисунок 3.6 – База даних і її таблиці

На рисунку 3.7 зображену структуру таблиці users

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/> 1	<b>id</b>	int(11)			Ні	Немає		AUTO_INCREMENT
<input type="checkbox"/> 2	<b>login</b>	varchar(50)	utf8_general_ci		Ні	Немає		
<input type="checkbox"/> 3	<b>password</b>	varchar(150)	utf8_general_ci		Ні	Немає		
<input type="checkbox"/> 4	<b>salt</b>	varchar(100)	utf8_general_ci		Ні	Немає		
<input type="checkbox"/> 5	<b>role</b>	varchar(21)	utf8_general_ci		Ні	Немає		

Рисунок 3.7 – Структура таблиці users

На рисунку 3.8 зображено структуру таблиці listStudent

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1	id			Ні	Немає		AUTO_INCREMENT
<input type="checkbox"/>	2	nameStudent	varchar(256)	utf8_general_ci	Ні	Немає		
<input type="checkbox"/>	3	rating	varchar(1)	utf8_general_ci	Ні	Немає		
<input type="checkbox"/>	4	date	varchar(20)	utf8_general_ci	Ні	Немає		

Рисунок 3.8 – Структура таблиці listStudent

На рисунку 3.9 зображено структуру таблиці group, де знаходяться дані про групи

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1	id			Ні	Немає		AUTO_INCREMENT
<input type="checkbox"/>	2	group_name	varchar(10)	utf8_general_ci	Ні	Немає		
<input type="checkbox"/>	3	id_specialty	int(11)		Так	NULL		

Рисунок 3.9 – Структура таблиці group

На рисунку 3.10 зображено структуру таблиці specialty, де знаходяться дані про спеціальність студента.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1	id			Ні	Немає		AUTO_INCREMENT
<input type="checkbox"/>	2	specialty_name	varchar(50)	utf8_general_ci	Ні	Немає		
<input type="checkbox"/>	3	id_chair	int(11)		Так	NULL		

Рисунок 3.10 – Структура таблиці specialty

На рисунку 3.11 зображено структуру таблиці chair, де знаходяться дані про кафедру.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1	id			Ні	Немає		AUTO_INCREMENT
<input type="checkbox"/>	2	chair_name	varchar(50)	utf8_general_ci	Ні	Немає		
<input type="checkbox"/>	3	id_faculty	int(11)		Так	NULL		

Рисунок 3.11 – Структура таблиці chair

На рисунку 3.12 зображено структуру таблиці faculty, де знаходяться дані про факультет.


#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1 id 	int(11)			Ні	Немає		AUTO_INCREMENT
<input type="checkbox"/>	2 faculty_name	varchar(50)	utf8_general_ci		Ні	Немає		

Рисунок 3.12 – Структура таблиці faculty

### 3.3 Програмна реалізація систем обліку успішності

Програма була розроблена за допомогою фреймворка Qt Creator, на мові програмування C++.

Qt (вимовляється "к'ю-ті") є кросплатформовим фреймворком розробки програмного забезпечення, який дозволяє розробникам створювати програми для різних операційних систем, таких як Windows, Linux, macOS, Android та інші. Qt надає набір інструментів, бібліотек і компонентів для розробки графічних інтерфейсів користувача (рис 3.13), мережевих додатків, баз даних, роботи зі зображеннями, аудіо, відео, паралельного програмування та багатьох інших функцій. Він надає широкий набір інструментів і бібліотек для створення високоефективних програм з використанням мов програмування C++ і QML.



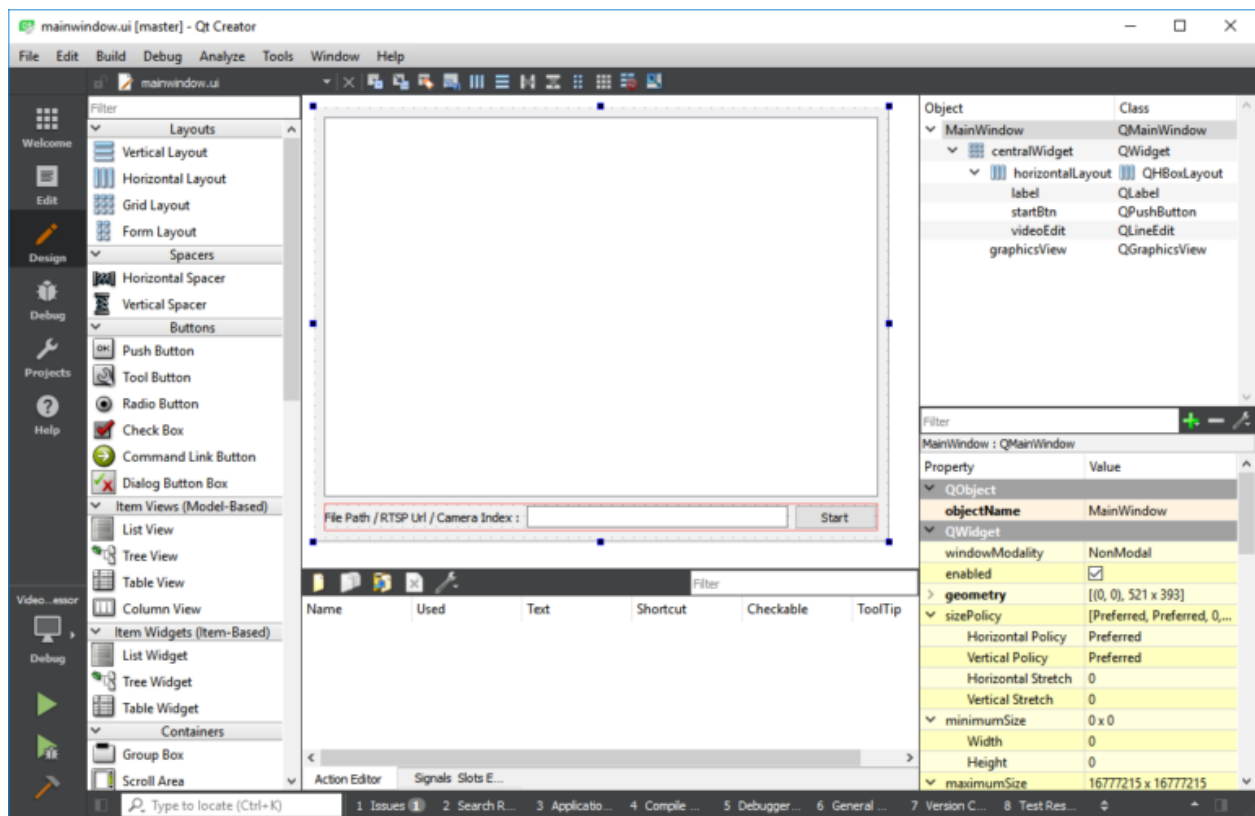


Рисунок 3.13 – QtDesigner, вбудований редактор форм

Qt дозволяє розробникам швидко створювати графічні інтерфейси користувача за допомогою готових компонентів і віджетів (рис. 3.14). Він надає можливості для роботи зі звуком, зображеннями, відео, анімацією і багатьма графічними можливостями.

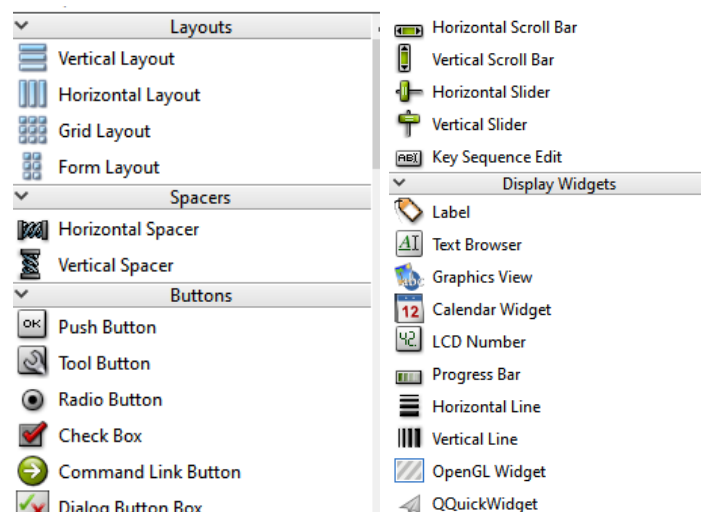


Рисунок 3.14 – Готові віджети і компоненти

Одним з ключових переваг Qt є його кросплатформовість. За допомогою Qt, розробники можуть написати код один раз і компілювати його для різних платформ, що значно зменшує зусилля та витрати на розробку для різних операційних систем.

Qt також надає потужні інструменти для мережевого програмування, роботи з базами даних, обробки XML, паралельного програмування, тестування і багатьох інших завдань, що сприяють створенню повноцінних і сучасних додатків.

Загалом, призначення Qt полягає в тому, щоб допомогти розробникам ефективно створювати кросплатформові додатки з графічним інтерфейсом, що мають високу продуктивність, багатий функціонал і професійний вигляд.

Основні функції Qt Creator. Qt є потужним фреймворком розробки програмного забезпечення, який надає широкі можливості для створення кросплатформових додатків. Основні функції і особливості Qt включають:

- Графічний інтерфейс користувача: Qt надає багато інструментів для розробки професійних і естетично привабливих графічних інтерфейсів. Він має велику бібліотеку готових елементів керування, таких як кнопки, поля введення, таблиці, діалогові вікна, меню та інші. Qt також підтримує розміщення елементів у вікні за допомогою гнучкого системи розташування.

- Мультимедіа: Qt дозволяє роботу з мультимедійними даними, включаючи відтворення аудіо та відео, запис звуку, обробку зображень та відео. Це відкриває широкі можливості для розробки додатків, пов'язаних з мультимедіа, таких як медіаплеєри, фоторедактори, відеоредактори та інші.

- Мережеве програмування: Qt надає потужні інструменти для розробки мережевих додатків. Він підтримує різні протоколи мережевого зв'язку, такі як TCP, UDP, HTTP і FTP, і дозволяє здійснювати мережеві операції, такі як створення серверів, відправлення та отримання даних через мережу.

- Робота з базами даних: Qt надає інструменти для роботи з різними СУБД, такими як MySQL, SQLite, PostgreSQL та інші. Він дозволяє

здійснювати з'єднання з базами даних, виконувати запити SQL, отримувати та зберігати дані. Qt також надає ORM (Object–Relational Mapping) для спрощення роботи з базами даних.

– Мультиплатформовість: Одна з ключових особливостей Qt – це його здатність працювати на різних платформах, включаючи Windows, macOS, Linux, Android та інші. Розроблені додатки можуть бути компільовані для різних цільових платформ без змін у вихідному коді. Це дозволяє створювати кросплатформові додатки з одним кодом, що спрощує розробку та підтримку.

– Паралельне програмування: Qt підтримує паралельне програмування за допомогою потоків та паралельних обчислень. Він надає засоби для створення та керування потоками, синхронізації даних між потоками та інші інструменти, що допомагають виконувати обчислення паралельно та ефективно використовувати ресурси системи.

– Інтернаціоналізація та локалізація: Qt забезпечує підтримку інтернаціоналізації та локалізації, що дозволяє створювати додатки, які можуть працювати з різними мовами та регіональними налаштуваннями. Він надає інструменти для перекладу тексту, форматування чисел, дат та інших локалізованих даних.

– Інструменти для тестування: Qt надає набір інструментів для автоматизованого тестування, що допомагає виявляти та виправляти помилки в програмному забезпеченні. Ці інструменти дозволяють створювати тестові сценарії, запускати їх автоматично та аналізувати результати.

Програма складається з чотирьох основних класів: MainWindow, MenuWindow, TaskWindow та CreateTaskWindow. Кожен з цих класів виконує визначену функціональність. Інші класи є допоміжними і служать для додаткових можливостей даного програмного продукту

Клас MainWindow є головним вікном програми і виконує функції реєстрації та авторизації користувача. Він відповідає за взаємодію з користувачем, введення та перевірку даних, а також управління основними функціями системи.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

Основні функції та властивості класу MainWindow можуть включати:

1. Функція реєстрації:
  - Введення та перевірка даних користувача, таких як ім'я, електронна пошта та пароль;
  - Збереження даних користувача в базі даних.
2. Функція авторизації:
  - Введення та перевірка облікових даних користувача;
  - Перевірка наявності користувача в базі даних;
  - Відкриття основного інтерфейсу після успішної авторизації.
3. Взаємодія з іншими класами та модулями:
  - Звернення до бази даних для збереження та отримання інформації про користувачів;
    - Виклик методів та функцій інших класів для виконання певних операцій.
4. Обробка помилок та повідомлення користувачу:
  - Відображення повідомлень про невірні дані, помилки аутентифікації або інші виняткові ситуації;
  - Забезпечення зручного інтерфейсу для користувача під час взаємодії з програмою.

Клас MenuWindow є меню або головним вікном програми, яке надає можливість користувачеві переходити до різних функціональних частин системи обліку успішності студентів. Оніць клас відповідає за відображення головного меню та навігацію до двох основних функціональних вікон: CreateTaskWindow (вікно для реєстрації оцінки) та TaskList (вікно для відображення списку оцінок).

Основні функції та властивості класу MenuWindow можуть включати:

1. Відображення головного меню:
  - Відображення пунктів меню для навігації до різних функціональних вікон;
  - Надання користувачеві можливості обрати потрібний пункт меню.
2. Навігація до CreateTaskWindow:

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

- Забезпечення можливості переходу до вікна реєстрації оцінки (CreateTaskWindow);

- Запуск вікна, яке дозволяє користувачеві ввести дані про оцінку студента, а також автоматичне введення дати і часу.

### 3. Навігація до TaskList:

- Забезпечення можливості переходу до вікна відображення списку оцінок (TaskList);

- Запуск вікна, яке показує користувачеві інформацію про оцінки студентів.

Клас CreateTaskWindow відповідає за вікно, де користувач може здійснювати реєстрацію оцінки студента та занесення цієї інформації до бази даних системи обліку успішності студентів.

Клас TaskWindow відповідає за вікно, де користувач може переглядати оцінки студентів, взяті з бази даних системи обліку успішності.

На рисунку зображено спрощену діаграму класів для програми

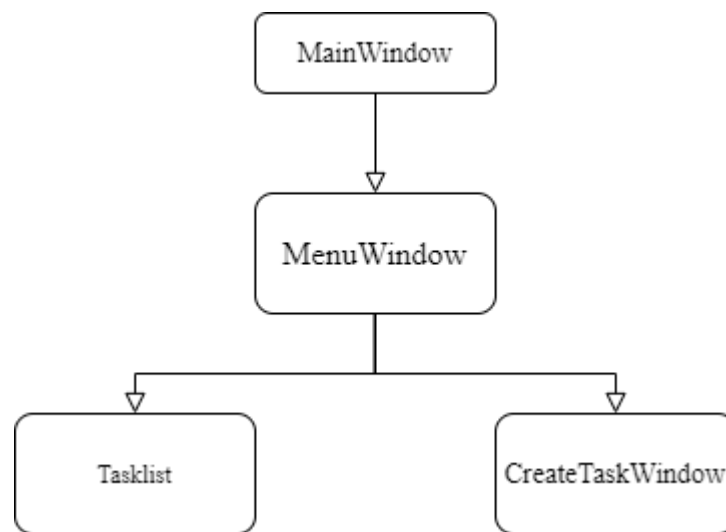


Рисунок 3.15 – Спрощена діаграма класів для програми

На рисунку 3.16 зображено діаграму варіантів використання

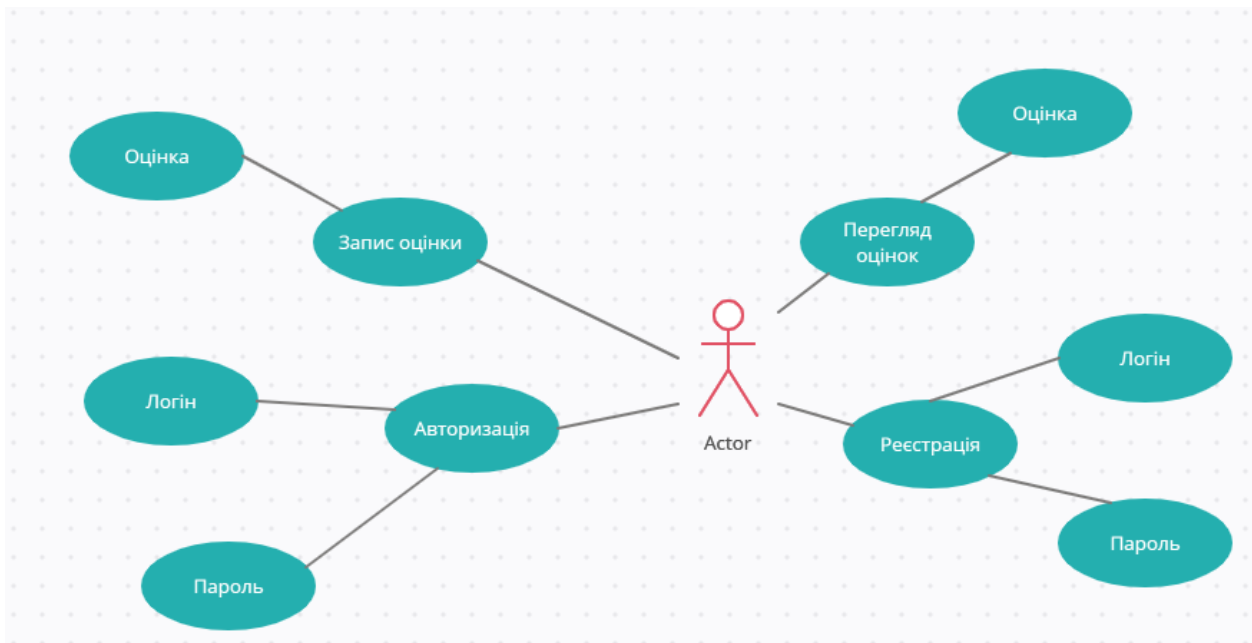


Рисунок 3.16 – Діаграма варіантів використання

### 3.4 Тестування та верифікація розробленого програмного забезпечення

Є кілька підходів до тестування програм, основні з них: модульне тестування, автоматизоване тестування інтерфейсу користувача, ручне тестування.

Модульні тести – це вид тестування програмного коду, де окремі модулі (наприклад, функції, методи, класи) перевіряються на правильність своєї роботи. Це дозволяє виявляти та виправляти помилки в коді на ранніх етапах розробки, покращує стабільність та якість програми, а також спрощує рефакторинг і розширення функціональності.

Основні характеристики модульних тестів включають:

- Незалежність: Кожен модульний тест повинен бути незалежним від інших тестів, щоб результати виконання одного тесту не впливали на інші. Кожен тест повинен перевіряти лише один аспект або функціональність модуля;

– Повнота: Модульні тести повинні покривати всі можливі шляхи виконання коду, включаючи всі гілки коду, умовні оператори, вхідні та вихідні параметри, а також виключні ситуації. Це допомагає забезпечити, що всі частини модуля працюють належним чином;

– Автоматизація: Модульні тести повинні бути легко виконуваними автоматично без необхідності втручання користувача. Це дозволяє швидко та ефективно перевіряти правильність роботи модулів під час розробки та змін коду;

– Перевірка результатів: В модульних тестах використовуються перевірки, які порівнюють очікувані результати з фактичними результатами виконання модулів. Це допомагає виявити неправильну роботу коду та помилки;

– Ізоляція: Модульні тести повинні бути ізольованими від залежностей та впливів зовнішніх факторів. Зазвичай, для досягнення цього використовуються спеціальні заглушки (mocks) або фейкові об'єкти, які імітують поведінку залежних компонентів.

Незважаючи на численні переваги модульного тестування, воно також має свої недоліки. Деякі з них включають:

– Витрати на розробку: Написання модульних тестів може зайняти певний час і зусилля. Необхідно витратити час на написання коду тестування, налагодження і підтримку тестів;

– Витрати на підтримку: З розвитком програмного забезпечення та зміною вимог можуть змінюватися модулі, які потребують тестування. Це може призвести до необхідності вносити зміни в існуючі модульні тести, що вимагає додаткового часу та ресурсів;

– Обмежена покриття: Незважаючи на те, що модульні тести дозволяють перевірити окремі функції та методи, вони не забезпечують повне покриття всього програмного коду. Іноді важко або неможливо перевірити взаємодію між модулями, інтеграцію зовнішніх компонентів або нелінійні аспекти програми;

– Вразливість до змін: Якщо модульне тестування залежить від конкретних реалізацій модулів, то будь-які зміни у внутрішніх деталях модуля можуть призвести до зламу тестів. Це може стати проблемою, якщо необхідно вносити зміни у функціональність модулів;

– Вартість підтримки: З плином часу, з розширенням програмного забезпечення та збільшенням кількості модулів, вартість підтримки модульних тестів може зрости. Ресурси, необхідні для виконання всіх тестів, можуть стати значними.

Автоматизоване тестування інтерфейсу користувача (UI Tests) є одним з видів тестування програмного забезпечення, яке спрямоване на перевірку функціональності, надійності та коректності роботи інтерфейсу користувача програми. Воно використовується для перевірки взаємодії між користувачем та програмою, а також для виявлення можливих проблем та дефектів, які можуть виникнути при взаємодії з програмним інтерфейсом.

Основні особливості автоматизованого тестування інтерфейсу користувача включають наступні аспекти:

– Запис та відтворення дій: Один із способів автоматизованого тестування інтерфейсу полягає в записі дій користувача і створенні скрипту, який може відтворювати ці дії. Такий підхід дозволяє повторювати одні й ті ж тестові сценарії без необхідності вручного втручання;

– Маніпулювання елементами інтерфейсу: Автоматизовані тести можуть маніпулювати елементами інтерфейсу, такими як кнопки, поля введення, вкладки тощо. Це дозволяє перевіряти реакцію програми на різні взаємодії користувача та перевіряти правильність роботи інтерфейсу;

– Перевірка результатів: У автоматизованих тестах використовуються перевірки для порівняння очікуваних результатів з фактичними результатами взаємодії з інтерфейсом. Наприклад, можна перевірити, чи змінюється вміст поля введення після введення даних або чи з'являється очікуване повідомлення після натискання кнопки;

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60



– Перехоплення подій: Автоматизовані тести можуть перехоплювати події, що виникають при взаємодії з інтерфейсом, такі як натискання кнопок, переміщення курсору миші, введення клавіш тощо. Це дозволяє тестам симулювати взаємодію користувача з програмним інтерфейсом;

– Кросплатформеність: Автоматизовані тести можуть бути написані таким чином, щоб вони працювали на різних платформах, таких як Windows, macOS або Linux. Це дозволяє забезпечити однакову якість тестування на різних середовищах;

– Інтеграція з іншими інструментами: Автоматизовані тести можуть бути інтегровані з іншими інструментами, такими як системи управління версіями, засоби збирання та постачання програмного забезпечення (CI/CD), середовища для тестування інтеграції та інше. Це сприяє автоматизації процесу тестування та підвищує ефективність розробки.

Незважаючи на переваги, автоматизоване тестування інтерфейсу користувача також має деякі недоліки:

– Витрати на розробку: Розробка автоматизованих тестів може вимагати значних зусиль та ресурсів. Необхідно планувати, розробляти та підтримувати тестові скрипти, що може забирати час та зусилля розробників;

– Вразливість до змін інтерфейсу: Якщо інтерфейс програми часто змінюється, автоматизовані тести можуть вимагати постійного оновлення та підтримки. Зміни в розташуванні елементів, вигляді, назвах тощо можуть призвести до невідпрацьованих тестів і вимагати їх переробки;

– Вартість підтримки: Підтримка автоматизованих тестів вимагає постійного моніторингу та внесення змін у випадку несправностей або змін у програмі. Це може вимагати додаткових ресурсів та затрат;

– Обмежена можливих сценаріїв: Автоматизовані тести можуть бути обмеженими в покритті всіх можливих сценаріїв взаємодії з програмним інтерфейсом. Не завжди можливо передбачити всі можливі варіанти взаємодії користувача з програмою, тому деякі потенційні проблеми можуть залишатися невиявленими;

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

– Чутливість до змін у кодї: Якщо внесені зміни в код програми порушують логіку взаємодії з інтерфейсом, автоматизовані тести можуть стати некоректними та потребувати оновлення або переробки.

Враховуючи ці недоліки, важливо ретельно планувати тестування інтерфейсу користувача, визначати, які аспекти потребують автоматизації, і забезпечувати належну підтримку та обслуговування автоматизованих тестів, щоб максимально скористатися перевагами цього виду тестування.

Ручне тестування – це процес перевірки та валідації програмного продукту або його частини вручну без використання автоматичних інструментів чи скриптів. У ручному тестуванні тестувальник (людина) виконує ролі користувача та перевіряє функціональність, надійність, взаємодію та інші аспекти програмного продукту.

Під час ручного тестування тестувальник виконує різні дії та взаємодіє з програмою, спостерігаючи за результатами та перевіряючи, чи вони відповідають очікуванням. Це може включати введення даних, виконання операцій, перевірку відповідей, перегляд даних та багато іншого.

Було вибрано ручне тестування. В процесі ручного тестування, результати цих тестів будуть відображені в таблиці (таблиця 3.1). Крім того, для наглядності та детального представлення результатів, можуть використовуватися рисунки.

Перевірка на хешування. Для перевірки на хешування ми зареєструємо нового користувача (рис. 3.17). Дані будуть наступні: логін – Ostrovskyi, пароль – Ostrovskyi.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

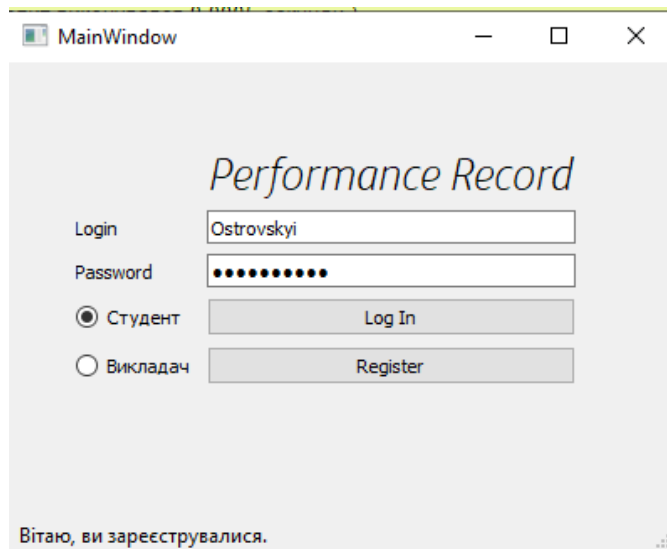


Рисунок 3.17 – Реєстрація користувача для тестування

Перейдемо в базу даних щоб побачити результат, як бачим в БД логін – Ostrovskiy, а пароль в хешованому вигляді це можна побачити на рисунку 3.18

	id	login	password	salt	role
<input type="checkbox"/> Редагувати <input type="checkbox"/> Копіювати <input type="checkbox"/> Видалити	50	123	4822a7d7f79c045bb92883a107b2bc50d5e3d88fc07491d533...	1294d97f99528f8cd139d8f874369527	student
<input type="checkbox"/> Редагувати <input type="checkbox"/> Копіювати <input type="checkbox"/> Видалити	51	1234	d7a6d2c4ace3fccb6a6cf5463f8b6ef1433eeb3936b83b1dc3...	3f3970c13f0994e0ebd07a9d886eed13	teacher
<input type="checkbox"/> Редагувати <input type="checkbox"/> Копіювати <input type="checkbox"/> Видалити	53	Ostrovskiy	a87b59769a74248c8712ae3e54a2ab537baf8c4f08adb94598...	fe543b655f74b8de1b90d2aa93d77de3	student

Рисунок 3.18 – Хешовані дані

В ході тестування було виявлено невеликий недолік, пов'язаний з функцією вибору ролі під час авторизації. В процесі тестування було помічено, що після введення облікових даних користувача для авторизації, можна натиснути на роль, яка використовується під час реєстрації. Однак, виявлено, що ця дія не призводить до жодних очевидних змін у функціональності програми або подальшому ходу тестування.

Після натискання на роль, жодні зміни не відбуваються щодо доступу до різних функціональних можливостей чи зміни інтерфейсу програми. Роль, яка була вибрана під час реєстрації, продовжує залишатися активною, і жодні інші ролі не можуть бути вибрані.

Хоча цей недолік може здатися мінорним, варто відмітити, що він не впливає на коректність роботи програми чи її функціональність. Втім, з точки зору користувача, може бути кращим підходом приховати або вимкнути

можливість натискання на роль під час авторизації, якщо ця дія не має реального впливу на подальшу роботу з програмою.

Таблиця 3.2 – Результати тестування

Дія	Результат	Оцінка
Запуск програми	З'являється вікно з реєстрацією/авторизацією	Вірно
Реєстрація	В рядку станів з'являється сповіщення про успішну реєстрацію	Вірно
Перевірка на хешування	В базі даних пароль відображається хешованим (рис. 3.2)	Вірно
Авторизація з вірними даними	З'являється вікно з головним меню	Вірно
Авторизація з не вірними даними	З'являється діалогове віконце, яке сповіщує про введені не вірні дані	Вірно
Перехід до вікна внесення даних в БД з роллю teacher	З'являється вікно для внесення даних	Вірно
Перехід до вікна внесення даних в БД з роллю student	З'являється діалогове вікно про те що на цю дію в нас немає права	Вірно
Перехід до вікна перегляду даних про студента	Відображається таблиця з прізвищем, імям, оцінкою, і датою виставлення оцінки	Вірно

В ході тестування не виявлено значних проблем.

## 4 ТЕХНІКО–ЕКОНОМІЧНИЙ РОЗДІЛ

### 4.1 Визначення витрат на оплату праці та відрахувань у соціальні фонди

Розроблене програмне забезпечення призначене автоматизації бази даних для обліку успішності студентів. Алгоритм проектування реляційної бази даних, на якому базується розроблене програмне забезпечення не вимагає значних обчислювальних процедур та характеризується високою швидкістю реалізації запитів, що уможлиблює одержати потрібну інформацію із мінімальними витратами часу.

Техніко – економічне обґрунтування розробки комп'ютерних систем включає етапи:

- опис технологічного процесу розробки із зазначенням трудомісткості кожної операції;
- визначення суми витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування у державні соціальні фонди;
- визначення суми матеріальних витрат;
- обчислення витрат на електроенергію;
- розрахунок транспортних витрат;
- нарахування суми амортизаційних відрахувань;
- визначення суми накладних витрат;
- складання кошторису та визначення собівартості розробки;
- розрахунок ціни;
- обґрунтування вибору комплексу технічних і програмних засобів;
- визначення економічної ефективності та терміну окупності розробки.

У розробці проектного рішення задіяні наступні спеціалісти – розробники, а саме всі хто вказані в таблиці 4.1. Форму поділу робіт по всіх основних етапах і видах робіт, які повинні бути виконані показано в таблиці 4.1.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

Таблиця 4.1 – Середній час виконання проекту та стадії технологічного процесу

№ п/п	Назва операції (стадії)	Виконавець, посада	Середній час виконання операції, год.
1.	Підготовка	Project Manager	7
2	Розробка проекту системи	Project Manager	16
		Дата звернення Administrator	15
		Developer	50
3	Проектування технічної частини системи	Developer	21
4	Розробка програмного продукту	Developer	6
5	Тестування системи	QA	25
Разом			130

Визначення витрат на оплату праці та відрахувань на соціальні заходи. Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{оп} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij} \quad (4.1)$$

Середньо годинна ставка працівника може бути розрахована за формулою:

$$C_{ij} = \frac{C_{ij}^0 (1+h)}{PЧ_i} \quad (4.2)$$

де  $C_{ij}^0$  – основна місячна заробітна плата розробника  $i$ -ої спеціальності  $j$ -го тарифного розряду, грн.;

$h$  – коефіцієнт, що визначає розмір додаткової заробітної плати;

$РЧ_i$  – місячний фонд робочого часу працівника  $i$ -ої спеціальності  $j$ -го розряду, год.

Таблиця 4.2 – Вихідні дані для розрахунку витрат на оплату праці

№ п/п	Посада виконавців	Місячний оклад грн.	Коефіцієнт Додаткової з/п	Вартість погодинної оплати	Підсумок
1	Project Manager	15000	0	90	15000
2	Developer	25000	0	150	25000
3	Дата зверненняabase Administrator	20000	0	120	20000
4	QA	15000	0	90	15000

Звідси, загальні витрати на оплату праці (ВОП) дорівнюють:

$$\text{ВОП} = 2070 + 2250 + 7440 + 5040 = 16\,800 \text{ грн.}$$

Крім того, слід визначити відрахування на соціальні заходи. Величину відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5 % від суми заробітної плати:

$$\text{ВФ} = 20.5 * 16\,800 = 3444 \text{ (грн.)}$$

Матеріальні витрати — це вартість витрачених матеріалів, малоцінних та швидкозношуваних предметів на виробництво продукції, робіт або послуг, а

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

також матеріалів, витрачених на адміністративні, збутові та інші потреби підприємства. Загальна сума витрат на матеріальні ресурси (ВМ) визначається за формулою:

$$B_M = \sum_{i=1}^n K_i \cdot C_i, \quad (4.3)$$

де  $K_i$  – витрата  $i$ -го типу матеріалу, натуральні одиниці вимірювання;

$C_i$  – ціна за одиницю  $i$ -го типу матеріалу, грн.;

$i$  – тип матеріального ресурсу;

$n$  – кількість типів матеріальних ресурсів.

Таблиця 4.3 – Розрахунок витрат на матеріали та комплектуючі

№ п/п	Найменування купованих виробів	Одиниця виміру	Ціна, грн	Кількість купованих виробів	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
1	Програмне забезпечення для тестування	шт	987,56	1	987,56	98,75	1086,31
2	Програмне забезпечення для розробки	шт	2200	1	2200	220	2420
3	Папір (формат А4)	шт	50	2	100	10	110
Разом							3616

#### Розрахунок витрат на електроенергію

Для розробки використовується електрообладнання, тому необхідно розрахувати витрати на електроенергію. Загальна сума витрат на електроенергію розраховується за формулою:

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68



$$B_E = \sum_{i=1}^n P_i \cdot k_i \cdot T_i \cdot C, \quad (4.4)$$

де  $P_i$  – паспортна потужність  $i$ -го електрообладнання, кВт;

$k_i$  – коефіцієнт використання потужності  $i$ -го електрообладнання (приймається 0.7...0.9);

$T_i$  – час роботи  $i$ -го устаткування за весь період розробки, год;

$C$  – ціна електроенергії, грн / кВт\*год;

$i$  – тип електрообладнання;

$n$  – кількість електрообладнання.

Для розробки проекту даної системи використовується один ноутбук потужністю  $P = 0,5$  кВт, який за весь період розробки працює 200 годин та друкуючий пристрій потужністю  $P = 0,37$  кВт, який працює 2 години. Проміжні розрахунки на витрату електроенергії подані в таблиці 4.4

Таблиця 4.4 – Проміжні розрахунки на витрату електроенергії

Найменування устаткування	Паспортна потужність, кВт	Коефіцієнт використання потужності	Час роботи обладнання для розробки, год	Ціна електроенергії,	Сума, грн.
Ноутбук	0,5	0,98	61,25	1,44	88,2
Принтер	0,37	0,98	0,45	1,44	0,65
Разом					88,85

Обчислення накладних витрат Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці можуть становити до 150 % від суми основної та додаткової заробітної плати працівників. Накладні витрати для даного проекту подані далі.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

$$B_H = 0,7 \cdot B_{OP} \quad (4.5)$$

$$B_H = 0,7 * 16\,800 = 11\,760 \text{ грн.}$$

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I = 16\,800 * 0.1 = 1680,0 \text{ грн}$$

Складання кошторису витрат та визначення собівартості

Таблиця 4.5 – Кошторис витрат на розробку

Зміст витрат	Сума, грн.
Витрати на оплату праці (осн. і дод. ЗП)	16800
Відрахування на соціальні заходи	3444
Матеріальні витрати	3616
Витрати на електроенергію	88,85
Амортизаційні відрахування	відсутні
Транспортні витрати	відсутні
Накладні витрати	11 760
Інші витрати	1680,0
Разом	37 388,85

Визначення прогнозованої (договірної) ціни КС Величина можливої (договірної) ціни КС повинна визначатися з урахуванням ефективності, якості і термінів її виконання на рівні, що відповідає економічним інтересам замовника (споживача) і виконавця (підрядника). Договірна ціна (Дц) для прикладних КС розраховується за формулою:

$$Дц = 37\,388,85 * 1,25 = 46\,736,0625$$

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

де ВКС – витрати на розробку КС (з таблиці 2.7), грн.;

$r$  – середній рівень рентабельності КС, % (приймається в розмірі 20–30%).

Розраховану можливу ціну КС варто порівняти із цінами на аналогічні розробки, що існують на ринку ІТ продуктів.

$$E_p = \frac{\Pi}{B_{КС}} \quad (4.6)$$

де  $\Pi$  – прибуток, грн.;

$B_{КС}$  – кошторисна вартість, грн.

Прибуток розраховували за формулою:

$$\Pi = B_{КС} \cdot 15\% \quad (4.7)$$

$$\Pi = 5\,608,32 \text{ грн}$$

$$B_{КС} = 37\,388,85$$

$$E_p = \frac{5608,32}{37388} = 0,15$$

Поряд із економічною ефективністю розраховують термін окупності за формулою:

$$T_p = \frac{1}{E_p} \quad (4.8)$$

$$T_p = \frac{1}{0,15} = 6,6$$

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

Таблиця 4.6 – Показники економічної ефективності

Показник	Значення
Загальні Витрати	37 388,85
Прогнозована ціна	46 736,0625
економічна ефективність	0,15
Прибуток	5 608,3275
Термін окупності	6,6

Отже, у даному розділі проведено розрахунок витрат на розробку бази даних для обліку успішності студентів.

В результаті проведених розрахунків, що обґрунтовують економічну ефективність розробки бази даних, можна зробити висновок, що розроблене програмне забезпечення є економічно вигіднішим. Це обумовлено тим, що розробку бази даних здійснював студент.

## ВИСНОВКИ

У кваліфікаційній роботі проведена практична задача з проектування реляційної бази даних для обліку успішності в навчальних закладах з використанням мови SQL. Під час дослідження отримані наступні практичні результати:

1. Проведено аналіз предметної області
2. Проведена оцінка ефективності впровадження інформаційних технологій у навчальних закладах.
3. Охарактеризовані існуючі програмні засоби для обліку успішності.
4. Спроектвана логічна модель бази даних з відображенням зв'язків між таблицями.
5. Розроблена фізична модель бази даних для навчального закладу з використанням програмного середовища РНРMyAdmin.
6. Обґрунтовані техніко–економічні показники ефективності розробки бази даних для шкільної бібліотеки.

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. О. М. Березький, Г. М. Мельник, Л. О. Дубчак, Ю. М. Батько. Методичні вказівки до випускних кваліфікаційних робіт освітнього рівня “Бакалавр” спеціальності “Комп’ютерна інженерія”. Тернопіль: ЗУНУ, 2021. – 52 с.
2. А. П. Островський, Р. А. Бриняк. Аналіз програмних засобів розробки нотаток. 2023 с. 2.
3. Документація Qt: веб-сайт. URL: <https://doc.qt.io> (дата звернення 03.04.2023).
4. Документація phpMyAdmin: веб-сайт. URL: <https://docs.phpmyadmin.net/en/latest> (дата звернення 06.04.2023).
5. USBWebserver: веб-сайт. URL: <https://usbwebserver.yura.mk.ua> (дата звернення 25.01.2023).
6. Реляційна база даних: веб-сайт. URL: [https://uk.wikipedia.org/wiki/Реляційна\\_база\\_даних](https://uk.wikipedia.org/wiki/Реляційна_база_даних) (дата звернення 01.04.2023)
7. Система дистанційного навчання ЗУНУ: веб-сайт. URL: <https://moodle.wunu.edu.ua> (дата звернення 10.12.2022).
8. Flowchart Maker & Online Diagram Software: веб-сайт. URL: <https://app.diagrams.net> (дата звернення 24.05.2023).
9. Управління життєвим циклом студентів: веб-сайт. URL: <https://ispro.ua/page/upravlinnya-zhittyevim-ciklom-studentiv> (дата звернення 16.05.2023).
10. Допомога та навчання з Excel: веб-сайт. URL: <https://support.microsoft.com/uk-UA/excel> (дата звернення 16.05.2023).
11. Електроний журнал ВНМУ ім. М. І. Пирогова: веб-сайт. URL: <https://ez.vnmu.edu.ua> (дата звернення 17.05.2023).
12. Журнал успішності студентів Київського національного економічного університету ім. В. Гетьмана: веб-сайт. URL: [https://kneu.edu.ua/ua/Information\\_for/students/jurnal](https://kneu.edu.ua/ua/Information_for/students/jurnal) (дата звернення 17.05.2023).

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		74

13. Документація MySQL: веб-сайт. URL: <https://dev.mysql.com/doc> (дата звернення 23.03.2023).

14. Цифровізація освіти: мон розпочало пілотування державних шкільних е-журналів та е-щоденників: веб-сайт. URL: <https://mon.gov.ua/ua/news/cifrovizaciya-osviti-mon-rozpochalo-pilotuvannya-derzhavnih-shkilnih-e-zhurnaliv-ta-e-shodenniki> (дата звернення 16.05.2023).

15. Чумак К. О., Петренко В. О. Теоретико-прикладні засади використання інформаційно-комунікативних технологій у діяльності закладу вищої освіти. 2021. с.235.

16. Встановлення phpMyAdmin: веб-сайт. URL: <https://magefan.com/ua/blog/how-to-install-phpmyadmin> (дата звернення 03.03.2023).

17. Карплюк С. О., Вакалюк Т. А. Огляд функціональних можливостей програмного забезпечення для управління освітнім процесом закладу вищої освіти. 2018. с.276.

18. Електронний журнал Національного педагогічного університету (НПУ) імені М. П. Драгоманова: веб-сайт. URL: <https://nmu.npu.edu.ua> (дата звернення 17.05.2023).

19. Локшина О. Відкрита освіта в європейському просторі: стратегія розбудови. Педагогічні науки: теорія, історія, інноваційні технології. 2018. с.86.

20. Програмне забезпечення для вищих навчальних закладів України: веб-сайт. URL: <http://www.politek-soft.kiev.ua> (дата звернення 18.05.2023).

21. Maintaining Accurate & Systematic Student Records: веб-сайт. URL: <https://study.com/academy/lesson/maintaining-accurate-systematic-student-records.html> (дата звернення 14.04.2023).

22. Why Is It Important To Maintain A Student Tracking Record. Can Student Tracking Improve Learning: веб-сайт. URL: <https://elearningindustry.com/why-maintain-student-tracking-record-to-improve-learning> (дата звернення 16.02.2023).

23. Student Tracking System: веб-сайт. URL: <https://www.iitms.co.in/blog/student-tracking-system-improving-performance.html> (дата звернення 23.03.2023).

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

24. Student Performance Record Report: веб-сайт. URL: <https://arhelp.renaissance.com/hc/en-us/articles/12921514434971-Student-Performance-Record-Report> (дата звернення 24.03.2023).

25. OnCourse Systems for Education: веб-сайт. URL: <https://oncoursesystems.com/about> (дата звернення 25.03.2023).

26. Student Record Management Systems: Definition, Benefits, and Top Tools: веб-сайт. URL: <https://www.softwaresuggest.com/blog/student-record-management-system> (дата звернення 25.03.2023).

27. Students' Academic Performance Дата зверненняaset: веб-сайт. URL: [https://www.kaggle.com/дата\\_зверненняaset/aljarah/xAPI-Edu-Дата\\_звернення](https://www.kaggle.com/дата_зверненняaset/aljarah/xAPI-Edu-Дата_звернення) (дата звернення 26.03.2023).

28. Discuss the impact of introducing a дата зверненняabase management system in managing records and academic: веб-сайт. URL: [https://homework.study.com/explanation/discuss-the-impact-of-introducing-a-дата\\_зверненняabase-management-system-in-managing-records-and-academic-performance-for-students.html](https://homework.study.com/explanation/discuss-the-impact-of-introducing-a-дата_зверненняabase-management-system-in-managing-records-and-academic-performance-for-students.html) (дата звернення 27.03.2023).

29. Мова SQL. Основні поняття: веб-сайт. URL: [http://moonexcel.com.ua/уроки-sql1-мова-основні-поняття\\_ua](http://moonexcel.com.ua/уроки-sql1-мова-основні-поняття_ua) (дата звернення 12.03.2023).

30. Огляд основних SQL запитів: веб-сайт. URL: <https://itvdn.com/ua/blog/article/m-sql> (14.03.2023).

31. Introduction to Salted-Hashed Passwords: веб-сайт. URL: <https://medium.com/swlh/introduction-to-salted-hashed-passwords-d19bd6f92480> (дата звернення 25.05.2023).

32. Навчальні підрозділи ЗУНУ: веб-сайт. URL: <https://www.wunu.edu.ua/educational-subdivisions> (28.05.2023).

33. C++ QTableWidgetItem: веб-сайт. URL: <https://cpp.hotexamples.com/examples/-/QWidget/item/cpp-qtablewidget-item-method-examples.html> (дата звернення 02.05.2023).

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76



34. Н.Я. Савка, І.Р. Паздрій. Методичні вказівки до виконання практичних робіт з дисципліни «Техніко-економічне обґрунтування розробки комп'ютерних систем». Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. с. 40.

35. Трофименко О.Г., Прокоп Ю. В., Буката Л.М. СКБД MySQL і робота з інтернет базами даних: метод. вказівки для виконання лабораторних робіт роботи з дисципліни "Створення та опрацювання баз даних". 2021. 56 с.

36. Електронний журнал обліку успішності студентів з web–доступом: веб-сайт. URL: <https://naurok.com.ua/elektronniy-zhurnal-obliku-uspishnosti-studentiv-z-web-dostupom-220282.html> (дата звернення 18.05.2023).

37. Ткачук Ю.О. Інформаційна система обліку успішності студентів вищої освіти. Фінансове забезпечення економік: матеріали науково-практичної студентської конференції Поліського національного університету. 2021р. с.97.

38. «Електронний журнал» Національного педагогічного університету (НПУ) імені М.П. Драгоманова: веб-сайт URL: <http://nmu.npu.edu.ua> (дата звернення 01.05.2023).

39. Student performance alerts: веб-сайт. URL: <https://studentcentral.iupui.edu/grades-progress/student-performance-alerts.html> (дата звернення 09.04.2023).

40. Andersson, U. B., Löfgren, H., & Gustafson, S. Forward-looking assessments that support students' learning: A comparative analysis of two approaches. Studies in Educational Evaluation. 2019. p.116.

41. Report Card: A Print-Friendly Record of Student Performance: веб-сайт. URL: <https://support.goguardian.com/s/article/Report-Card-A-Print-Friendly-Record-of-Student-Performance-1629333204214> (дата звернення 25.04.2023).

42. Effect of school records on the academic performance of school student: веб-сайт. URL: <https://sparklyn.com.ng/effect-of-school-records-on-the-academic-24889-d> (дата звернення 03.05.2023).

43. MySQL Workbench: веб-сайт. URL: <https://www.mysql.com/products/workbench> (дата звернення 17.03.2023).

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		77

44. 17 Best Student Record Management System: веб-сайт. URL: <https://clearinfo.in/blog/student-record-management-system> (дата звернення 09.03.2023).

45. What is The Student Record Management System: веб-сайт. URL: <https://astrialearning.com/what-is-the-student-record-management-system> (дата звернення 10.03.2023).

46. 4 Benefits of Student Record Management System Institutions Must Know: веб-сайт. URL: <https://www.academiaerp.com/blog/4-benefits-of-student-record-management-system-institutions-must-know> (дата звернення 01.04.2023).

47. Performance Using Student Information System, Moodle and the Mobile Application: веб-сайт. URL: <https://www.mdpi.com/2306-5729/6/11/110> (08.04.2023).

48. Shetty I.D. Shetty D.; Roundhal S. Student performance prediction. 2019. p. 160.

49. Top 10 benefits & advantages of a student management system: веб-сайт. URL: <https://www.edsembli.com/10-benefits-of-a-student-management-system> (дата звернення 02.04.2023).

50. Top 20 Highly Efficient Features of Student Information System: веб-сайт. URL: <https://www.creatrixcampus.com/blog/top-20-highly-efficient-features-student-information-system> (дата звернення 11.04.2023).

					КР.КІ.9500102.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		78