

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

«До захисту в ЕК»

Директор інституту(декан факультету)

Андрій ФОРСЮК

(ім'я та прізвище)

«13» грудня 2024р.

«До захисту допущено»

Завідувач кафедри

Сергій ГРИБКОВ

(ім'я та прізвище)

«13» грудня 2024р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА

зі спеціальності 122 «Комп'ютерні науки»

(код та назва спеціальності)

освітньо-професійної програми Управління інформацією та аналітика даних

на тему: «Дослідження та розробка аналітичної системи результатів тестування здобувачів освіти»

Виконав: здобувач 2 курсу, групи КН-2-4М

Шидловський Артем Петрович

(прізвище, ім'я, по батькові повністю)

(підпис)

Керівник Литвинов Валерій Андроникович

(прізвище, ім'я та по батькові повністю)

(підпис)

Консультанти

(ім'я та прізвище)

(підпис)

(ім'я та прізвище)

(підпис)

(ім'я та прізвище)

(підпис)

Рецензент

Мелішук ГІЙДА

(ім'я та прізвище)

(підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) незарядженої допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач

(підпис)

(підпис)

Київ - 2024р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
 Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки
 Освітній ступінь магістр
 Спеціальність 122 «Комп'ютерні науки»
(код і назва)
 Освітньо-професійна програма Управління інформацією і аналітика даних
(назва)

ЗАТВЕРДЖУЮ

Завідувач

кафедри Інформаційних технологій,
штучного інтелекту і кібербезпеки

Грибков С.В.

07 жовтня 2024 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Шидловського Артем Петровича

(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження та розробка аналітичної системи результатів тестування здобувачів освіти»

керівник роботи Литвинов Валерій Андроникович, проф, док. тех. наук,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 7 жовтня 2024 року №884-кк

2. Строк подання здобувачем роботи 04 грудня 2024

3. Вихідні дані до роботи Дані про навчальний заклад «Кадиевецький ліцей», Інформація про автоматизацію закладу, Дані про існуючі системи тестування, Вимоги до систем тестування, Вимоги до вхідних та вихідних даних

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) Дослідження та розробка аналітичної системи результатів тестування здобувачів освіти, аналіз вимог для створюваної системи, функціональні можливості системи, архітектура системи, вибір компонентів, додатків та пакетів даних для створювання системи, розробка бази даних, розробка інтерфейсу користувача

5. Перелік графічного матеріалу

Структура навчального закладу «Кадиевецький ліцей», скріншоти роботи програми, скріншоти інтерфейсів програми, скріншоти баз даних, ілюстрації створення аналітичних звітів

6. Консультанти розділів роботи

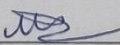
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
	Литвинов В.А, проф	07.10.2024	07.10.2024
	Литвинов В.А, проф	23.10.2024	23.10.2024
	Литвинов В.А, проф	30.10.2024	30.10.2024
	Литвинов В.А, проф	20.11.2024	20.11.2024

7. Дата видачі завдання 7 жовтня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз діяльності Кадиєвецького ліцею, та процесу тестування	07.10.2024-13.10.2024	Виконано
2	Вивчення існуючих програмних додатків та систем тестування	13.10.2024-18.10.2024	Виконано
3	Вивчення та аналіз об'єктів дослідження, створення плану їх реалізацій	18.10.2024-23.10.2024	Виконано
4	Визначення вимог та функцій до системи тестування	23.10.2024-28.10.2024	Виконано
55	Розробка архітектури системи та створення бази даних	28.10.2024-11.11.2024	Виконано
6	Створення інтерфейсу користувача, вікон програми	11.11.2024-24.11.2024	Виконано
7	Виправлення багів та неполадок програмного додатку	24.11.2024-27.11.2024	Виконано
8	Написання звіту з дипломної роботи	27.11.2024-05.12.2024	Виконано

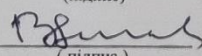
Здобувач


 (підпис)

Шидловський А.П.

(прізвище та ініціали)

Керівник роботи


 (підпис)

Литвинов В.А.

(прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему: «Дослідження та розробка аналітичної системи результатів тестування здобувачів освіти».

Автор роботи: Шидловський А.П.

Керівник роботи: Литвинов В.А

Спеціальність – 122 Комп'ютерні науки

Обсяг – 83 сторінок, 29 рисунків, 1 таблиця, 26 джерел

Мета кваліфікаційної роботи: розробка програмного додатку для автоматизації навчальних процесів та створення аналітичних звітів по результатах тестування.

У дослідженні було виконано аналіз інформаційної підтримки предметної області та потоків даних, розроблено структуру та схеми освітніх процесів. Програмний додаток, створений у рамках роботи, дозволяє перевіряти залишкові знання здобувачів освіти, автоматизує процес тестування та спрощує діяльність викладачів.

Під час виконання роботи згенеровано базу даних у СУБД MS SQL Server 2022 на основі SQL-файлу, створеного за допомогою AllFusion ERWin Data Modeler r7. Розроблено саму структуру програмного додатку, створено аналітичні звіти за результатами тестування. На завершальному етапі база даних була інтегрована з клієнтським додатком, розробленим у Visual Studio 2019 мовою програмування C#, а також створено функціональні вікна програми.

Ключові слова: C#, MICROSOFT VISUAL STUDIO 2019, ADOBE PHOTOSHOP 2018, INNO SETUP COMPILER, MS SQL SERVER 2022, ALLFUSION ERWIN DATA MODELER, СУБД, БД, ФОРМА.

ANNOTATION

Qualification work on the topic: "Research and Development of an Analytical System for Testing Results of Education Applicants".

Author: Shydlovskyi A.P.

Supervisor: Lytvynov V.A.

Specialty: 122 – Computer Science.

Volume: 83 pages, 29 figures, 1 tables, 26 references.

Objective: to develop a software application for automating educational processes and generating analytical reports on testing results.

The study includes an analysis of the information support of the subject area and data flows, as well as the development of structures and schemes for educational processes. The software application created as part of the work enables the evaluation of residual knowledge of education applicants, automates the testing process, and simplifies educators' workflows.

During the research, a database was generated in MS SQL Server 2022 based on an SQL file created with AllFusion ERWin Data Modeler r7. The application structure was developed, and analytical reports on test results were created. At the final stage, the database was integrated with a client application developed in Visual Studio 2019 using the C# programming language, and functional program windows were implemented.

Key words: C#, MICROSOFT VISUAL STUDIO 2019, ADOBE PHOTOSHOP 2018, INNO SETUP COMPILER, MS SQL SERVER 2022, ALLFUSION ERWIN DATA MODELER, DBMS, DATABASE, USER INTERFACE.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ ОБ’ЄКТА ДОСЛІДЖЕННЯ ТА ВИЯВЛЕННЯ ЗАДАЧ АВТОМАТИЗАЦІЇ.....	11
1.1 Загальна інформація про тестування в закладах освіти.....	11
1.2 Сучасні методи обробки результатів тестування та їх проблематика	12
1.3 Методи статистичного аналізу, що підвищують точність обробки освітніх даних, дослідження методів аналізу результатів тестування	14
1.4 Аналітика в тестуванні здобувачів освіти.....	16
1.5 Роль автоматизації в аналітиці.....	17
1.6 Загальна характеристика навчального закладу «Кадієвецький ліцей»	20
1.7 Організаційна структура навчального закладу «Кадієвецький ліцей».....	22
1.8 Аналіз нинішнього стану комп’ютеризації Кадієвецького ліцею	24
1.9 SWOT-аналіз діяльності Кадієвецького ліцею	25
1.10 Доцільність проектування та розробки системи.....	26
1.11 Економічний ефект від провадження системи для тестування.....	27
1.12 Висновок до першого розділу	28
РОЗДІЛ 2. ДОСЛІДЖЕННЯ ПРОБЛЕМИ ТА ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ	29
2.1 Науково-методичні основи розробки.....	29
2.2 Методи аналізу результатів тестування.....	32
2.3 Індивідуальні особливості здобувачів освіти.....	34
2.4 Загальні відомості про зміст роботи	35
2.5 Сучасні технології для дослідження та розробки аналітичної системи результатів тестування здобувачів освіти	37
2.6 Огляд існуючих рішень	39
2.7 Порівняння із запропонованими рішеннями	46
2.8 Загальні вимоги до програмного додатку.....	48
2.9 Архітектура рішення.....	49
2.10 Вимоги до гарантійної підтримки	50

2.11	Вимоги до інформаційного обміну	50
2.12	Режими роботи	51
2.13	Лінгвістичні вимоги	51
2.14	Вимоги до програмного забезпечення	51
2.15	Організаційні вимоги	52
2.16	Вимоги до розробки та передачі додатку	52
2.17	Вимоги до структури системи	52
2.18	Вимоги до функціоналу системи	53
2.19	Вимоги до збереження даних	54
2.20	Висновок до другого розділу	55
	РОЗДІЛ 3 ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ.....	56
3.1.	Основні задачі автоматизації	56
3.2	Автоматизація та алгоритмізація комплексу задач	57
3.3	Розробка програмного додатку	58
3.4	Інструкція користувача	67
3.5.	Технічне та системне забезпечення розробки	74
3.6.	Обґрунтування вибору технічних засобів	75
3.7	Обґрунтування вибору ОС та протоколу обміну даними	76
3.8.	Топологія комп'ютерної мережі	77
3.8	Висновок до третього розділу	78
	ВИСНОВОК.....	79
	ВИКОРИСТАНІ ДЖЕРЕЛА	81
	ДОДАТКИ.....	84

ВСТУП

Актуальність теми. Якість освіти сьогодні напряму залежить від наявності ефективних інструментів для оцінки знань здобувачів освіти. Зростання обсягів освітніх даних та необхідність адаптації навчальних програм до індивідуальних потреб студентів створюють потребу у розробці аналітичної системи для обробки результатів тестування. Така система здатна не лише автоматизувати процес оцінювання, але й надати комплексний аналіз навчальних досягнень, дозволяючи визначити прогалини у знаннях та побудувати індивідуальні навчальні траєкторії. Актуальність цієї теми підтверджується вимогами сучасного освітнього процесу, який прагне до використання технологічних інновацій для забезпечення високої якості освіти.

Застосування такої системи дозволить навчальним закладам підвищити ефективність процесу оцінювання знань, зокрема завдяки автоматичному виявленню проблемних місць у навчанні студентів та наданню персоналізованих рекомендацій для подальшого розвитку. Окрім того, система сприятиме підвищенню прозорості та об'єктивності оцінювання, що важливо для формування довіри серед студентів та викладачів.

Зв'язок роботи з актуальними освітніми програмами. Дослідження пов'язане з сучасними вимогами до забезпечення якості освіти, зокрема в рамках загальноосвітніх та спеціалізованих програм, які впроваджуються на національному та міжнародному рівнях. Потреба у створенні такої аналітичної системи відповідає стандартам підвищення якості оцінювання та забезпечення безперервного моніторингу навчального процесу.

Зв'язок роботи з науковими програмами, планами, темами. Наукова робота виконувалася згідно з планом та програмою наукових досліджень на кафедрі інформаційних технологій, штучного інтелекту і кібербезпеки Національного університету харчових технологій (НУХТ).

Об'єкт дослідження. Процеси обробки та аналізу результатів тестування здобувачів освіти.

Предмет дослідження. Підходи та методи обробки даних у системі оцінювання результатів тестування, а також методи візуалізації та інтерпретації навчальних досягнень.

Мета та задачі дослідження. Метою дослідження є удосконалення процесу оцінювання результатів тестування здобувачів освіти на основі розробки відповідної аналітичної системи, що дозволить забезпечити своєчасне корегування навчальних програм і сприяти індивідуалізації освітнього процесу, Для досягнення цієї мети визначені наступні задачі:

1. Дослідити сучасні підходи до обробки результатів тестування у освітньому середовищі та визначити проблемні аспекти їхнього використання;
2. Дослідити методи аналізу результатів тестування, що забезпечують об'єктивне оцінювання з урахуванням індивідуальних особливостей здобувачів та їх навчальних досягнень.
3. На основі проведених попередніх досліджень розробити аналітичну систему обробки результатів тестування і оцінити її переваги.

Методи дослідження. Для досягнення поставлених задач у дослідженні використовуються методи статистичної обробки даних, системного аналізу, моделювання та візуалізації результатів. Особлива увага приділяється методам структурованого представлення великих обсягів освітніх даних та їхнього інтерпретування для розробки індивідуальних рекомендацій.

Наукова новизна. Полягає в обґрунтуванні концепції, функції і засобів реалізації аналітичної системи обробки результатів тестування, що дозволяє

сформувати персональні освітні рекомендації і сприяє побудові індивідуальних траєкторій навчання.

Практичне значення отриманих результатів. Результати роботи можуть бути використані в освітніх установах для автоматизації процесу оцінювання та аналізу навчальних досягнень студентів, забезпечуючи індивідуальний підхід до їхнього розвитку. Аналітична система також надасть можливість керівникам навчальних закладів отримувати більш глибоке уявлення про освітній процес, що сприятиме оптимізації ресурсів та поліпшенню якості освіти в цілому.

Структура та обсяг магістерської роботи.

Магістерська робота містить вступ, 4 розділи, висновок, список використаної літератури та додатки. Основна частина викладена на 73 сторінки друкованого тексту. Робота включає в себе 25 ілюстрацій, 1 таблицю та 26 використаних джерел.

РОЗДІЛ 1 АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ ТА ВИЯВЛЕННЯ ЗАДАЧ АВТОМАТИЗАЦІЇ

1.1 Загальна інформація про тестування в закладах освіти

Тестування в закладах освіти є одним із ключових інструментів оцінювання знань, умінь і навичок здобувачів освіти. Воно використовується для перевірки рівня засвоєння навчального матеріалу, моніторингу прогресу, виявлення проблем у навчальному процесі, а також для підсумкових іспитів або вступу до інших навчальних закладів

Мета тестування:

1. Забезпечення загального розуміння засвоєння навчального матеріалу
2. Стимулювання учнів до систематичного навчання.
3. Виявлення сильних і слабких сторін у навчанні на деяких предметах, окремих здобувачів освіти або груп учнів та студентів.
4. Використання результатів тестування для вдосконалення освітнього процесу.

Види тестування

1. Поточне тестування: Проводиться протягом навчального року або семестру для узагальнення вивчення навчального матеріалу.
2. Підсумкове тестування: Використовується для перевірки загального рівня знань з певного курсу чи предмета.
3. Діагностичне тестування: Допомагає визначити рівень засвоєння поточного навчального матеріалу перед вивченням нового матеріалу.
4. Вхідне тестування: Використовується для визначення початкового рівня знань при вступі до навчального закладу.

Сучасне тестування активно використовує цифрові технології, які забезпечують зручність і точність у проведенні оцінювання. Використання автоматизованих систем дозволяє:

- Створювати різноманітні типи завдань – відкриті, закриті, зацікавлюючі, інтерактивні, розважальні.

- Швидко обробляти результати та формувати різноманітні аналітичні звіти та дашборди по результатах тестування.
- Зменшити роботу вчителів та педагогів для перевірки завдань.
- Забезпечити доступність тестування в онлайн-форматі.

Популярні програмні рішення для тестування включають Google Forms, Moodle, Kahoot, спеціалізовані платформи для дистанційного навчання – Zoom та Google Meet, а також системи на базі розробленого програмного забезпечення з інтеграцією баз даних.

Переваги тестування

- Об'єктивне оцінювання всіх здобувачів освіти.
- Економія часу викладачів на перевірці та створенні завдань.
- Адаптація для використання на різних дисциплінах.
- Забезпечення прозорості та наочності результатів.

Тестування здобувачів освіти відіграє надзвичайно велику роль у освітніх процесах у будь-яких навчальних закладах, адже допомагає зрозуміти на який предмет чи дисципліну потрібно перобити навчальну програму для кращого розуміння учнями, перевірити надлишкові знання кожного учня, тощо.

1.2 Сучасні методи обробки результатів тестування та їх проблематика

Сучасні підходи до обробки результатів тестування у освітньому середовищі значною мірою ґрунтуються на використанні автоматизованих систем оцінювання, тестування, та збору аналітичної інформації після тестування здобувачів освіти. Одним із найбільш поширених підходів є використання комп'ютерних систем для автоматизованої перевірки тестів, та виконаних завдань, що значно зменшує використання великого обсягу часу для перевірки завдань та знижує вплив людського фактору. Такі системи часто інтегруються із загальними платформами управління навчанням, наприклад, Moodle або Google Classroom тощо, що забезпечує централізоване збереження даних і зручний доступ до аналітики, звітностей по пройденому тестуванні і так далі.

Крім того, активно застосовуються методи адаптивного тестування, які враховують індивідуальний рівень знань студента і розробляють складність завдань у процесі тестування. Це підвищує точність оцінки та робить процес тестування більш гнучким. Ще одним інноваційним підходом є аналіз великих даних, який дозволяє викладачам отримувати детальну інформацію про успішність студентів, визначати прогалини в знаннях та адаптувати навчальні програми відповідно до потреб учнів.

Однак поряд із перевагами існують і проблемні аспекти використання сучасних підходів. По-перше, це технічні обмеження, насамперед необхідність забезпечення високої якості обладнання та програмного забезпечення. У багатьох навчальних закладах, особливо в країнах із недостатнім фінансуванням освіти, така автоматизація навчального процесу може бути недоступною. По-друге, є ризик надмірної залежності від автоматизованих систем, що може призвести до ігнорування людського фактора в оцінюванні, наприклад, при інтерпретації відкритих відповідей, де важливим є врахування контексту. Тому для завдань з відкритою відповіддю, або подібних, потрібна обов'язкова перевірка вчителем.

Також повстає питання конфіденційності інформації яка буде зберігатися на локальних середовищах або базах даних навчальних закладів. Особливо при зберіганні великого об'єму даних буде надзвичайно важко контролювати її збереження. Потрібно розробляти системи захисту інформації для попередження несанкціонованого доступу до неї, змінення та видалення інформації, тощо. Також потрібна реалізація авторизації учнів та викладачів в ці системи та надання їм окремих прав доступу до інформації яка зберігається в сховищах даних. Є можливість використання навіть подвійної аутентифікації користувачів для роботи з такими системами.

Таким чином, хоча сучасні технології відкривають широкі можливості для вдосконалення процесу обробки результатів тестування, їх використання вимагає ретельного підходу, який враховує технічні, організаційні та етичні аспекти. Вирішення цих проблем є необхідним для забезпечення справедливості, точності та ефективності оцінювання в освітньому середовищі.

Поряд з перевагами тестового контролю слід відмітити, що серед педагогів існують і думки про те, що тести не дають об'єктивного уявлення про якість засвоєного учнями змісту навчального предмета. Таке ставлення до тестового контролю насамперед можна пояснити великою імовірністю вгадування правильної відповіді. Особливо це характерно для найпоширеніших завдань закритої форми, а серед них — завдання з альтернативними відповідями: «так — ні». За таких умов імовірність вгадування правильної відповіді справді недопустимо висока. Подолати цей недолік можна, збільшивши кількість варіантів відповідей. Тому питання повинно бути чітким, лаконічним, зрозумілим, не перевантаженим зайвими словами.

1.3 Методи статистичного аналізу, що підвищують точність обробки освітніх даних, дослідження методів аналізу результатів тестування

Методи статистичного аналізу в освіті відіграють важливу роль у забезпеченні точності обробки даних і об'єктивного оцінювання учнів та студентів. Одним із найпоширеніших є використання описової та реляційної статистики, яка дозволяє узагальнювати результати тестування за допомогою таких показників, як середнє арифметичне, медіана, мода, стандартне відхилення. Ці дані допомагають зрозуміти загальну картину успішності, визначити рівень складності завдань і виявити тенденції в навчальних досягненнях студентів.

Більш точний аналіз можливий завдяки використанню інструментів інфраструктурної статистики, таких як критерії Стюдента або Хі-квадрат, що дозволяють порівнювати студентів, або цілі групи студентів, окремі класи або групи і оцінювати ефективність різних методів навчання. Регресійний аналіз також є потужним інструментом, оскільки дозволяє встановлювати залежності між різними факторами, наприклад, як різні методи підготовки впливають на результати тестування, або як повторне вивчення матеріалу вплине на результат тестування, тощо.

Для врахування індивідуальних особливостей студентів застосовуються більш гнучкі методи, зокрема кластерний аналіз і факторний аналіз. Кластерний

аналіз дозволяє групувати студентів за схожими характеристиками, такими як стиль навчання, рівень підготовки або особливості сприйняття матеріалу. Це допомагає адаптувати тестові завдання до конкретних груп, забезпечуючи більш точне і справедливе оцінювання. Факторний аналіз дозволяє ідентифікувати приховані фактори, які впливають на результати тестування, наприклад, мотивацію або рівень стресу, також виявити процес при якому матеріал що вивчається, просто «завчається» здобувачем освіти, щоб здати окремий тест.

Адаптивне тестування це ще один сучасний підхід для тестування, яке враховує індивідуальні особливості студентів. У процесі такого тестування складність завдань змінюється залежно від правильності відповідей на попередні запитання. Це дозволяє точніше визначити рівень знань кожного студента та уникнути ситуацій, коли сильні студенти недооцінюються через прості завдання, а слабші отримують низькі оцінки через занадто складні.

Для забезпечення об'єктивності також активно використовуються методи нормалізації результатів, що дозволяють вирівнювати оцінки за різними групами або тестами. Наприклад, застосування z-оцінок або процентильного ранжування допомагає уникнути впливу різниці у складності різних тестів. Такий підхід робить оцінювання справедливішим і дозволяє краще порівнювати результати між різними групами.

Дослідження методів аналізу результатів тестування з урахуванням індивідуальних особливостей студентів є важливим завданням сучасної педагогіки. Окрім уже згаданих підходів, слід звернути увагу на методи, які спрямовані на поглиблене розуміння освітніх даних. Наприклад, використання багаторівневого моделювання дозволяє враховувати як індивідуальні фактори, так і зовнішні впливи, зокрема соціально-економічний статус, рівень підтримки з боку батьків або особливості навчального середовища. Це дає змогу робити висновки не лише про знання студентів, але й про ефективність різних методів викладання.

Серед сучасних процесів тестування також виділяється застосування навчальної аналітики, яка базується на аналізі великих масивів даних. Ця методика дозволяє відстежувати динаміку навчальних досягнень кожного студента,

передбачати потенційні труднощі й розробляти індивідуальні схеми навчання. Застосування алгоритмів машинного навчання в аналізі освітніх даних стає дедалі популярнішим. Такі алгоритми здатні виявляти закономірності в даних, які залишаються непомітними при використанні традиційних статистичних методів.

Окрему увагу слід приділити психометричним підходам, які враховують не лише результати тестів, але й інші аспекти особистості студента. Наприклад, оцінка когнітивних стилів або рівня емоційної стійкості допомагає краще зрозуміти, чому певний студент досягає успіху або стикається з труднощами. Завдяки таким підходам можна розробляти тести, які не лише вимірюють знання, але й стимулюють розвиток критичного мислення, творчих здібностей і мотивації.

Також важливо зазначити роль формувального оцінювання, яке зосереджується не лише на кінцевих результатах, але й на процесі навчання. Використання методу порфоліо, де студент збирає свої досягнення за певний період, дозволяє враховувати індивідуальний прогрес. Це підходить для тих студентів, чий темп навчання може відрізнятись від середнього, але вони демонструють значний ріст і зусилля.

Таким чином, аналіз результатів тестування сьогодні включає широкий спектр інструментів, що враховують не лише стандартизовані показники, але й унікальні особливості кожного студента або учня. Впровадження цих методів у освітню практику дозволяє зробити систему оцінювання більш гнучкою, справедливою та орієнтованою на розвиток кожної особистості.

1.4 Аналітика в тестуванні здобувачів освіти

Аналітика в освітньому тестуванні є важливим етапом у забезпеченні ефективності навчального процесу та тестуванні загалом. Вона дозволяє не лише оцінювати результати тестів, але й глибоко аналізувати дані для виявлення закономірностей та проблем у засвоєнні матеріалу. Основними завданнями аналітики у тестуванні є оцінювання успішності, яке включає підрахунок середніх балів, визначення рівня знань і прогресу здобувачів освіти, а також виявлення слабких місць, що передбачає ідентифікацію тем чи розділів, які вимагають

додаткової уваги. Важливим аспектом є моніторинг динаміки, який дозволяє аналізувати зміни у результатах тестування з часом для оцінки ефективності викладання. Груповий та індивідуальний аналіз спрямований на порівняння результатів окремих учнів або груп для виявлення відмінностей у навчанні, тоді як формування рекомендацій базується на розробці стратегії покращення навчального процесу на основі зібраних даних.

Для здійснення аналітики використовуються різноманітні інструменти, серед яких програмне забезпечення для аналізу даних, наприклад, MS Excel, яке дозволяє будувати діаграми, таблиці та виконувати статистичні розрахунки, а також BI-платформи, такі як Power BI або Tableau, які забезпечують візуалізацію великих масивів даних. Серед спеціалізованих платформ для тестування можна виділити Moodle, Google Forms та Microsoft Forms, які дозволяють автоматизовано збирати й аналізувати результати тестування. Крім того, використовуються власноруч розроблені системи, які інтегрують бази даних, наприклад, MS SQL Server, із програмними інтерфейсами для аналітичної обробки даних.

Аналітика у тестуванні має значні переваги, серед яких об'єктивність, що мінімізує суб'єктивний фактор у визначенні результатів, прогнозування, яке дає змогу передбачати потенційні успіхи чи труднощі учнів, оптимізація навчання, яка вдосконалює методики викладання на основі отриманих даних, і підтримка адаптивного навчання, що забезпечує індивідуалізацію навчальних планів залежно від результатів тестування. Практичні приклади застосування аналітики включають формування рейтингу учнів на основі балів, аналіз частотності відповідей для виявлення найпоширеніших помилок, візуалізацію результатів за допомогою графіків і діаграм для швидкого аналізу успішності та складання докладних звітів для адміністрації закладів освіти чи викладачів.

1.5 Роль автоматизації в аналітиці

Сучасні системи тестування інтегрують аналітичні модулі, які дозволяють у реальному часі отримувати статистику та візуалізовані дані. Наприклад, програмні додатки, розроблені з використанням таких інструментів, як MS SQL Server, Visual

Studio та C#, дозволяють автоматично обробляти результати тестів і створювати аналітичні звіти.

Аналітика в тестуванні здобувачів освіти допомагає зробити освітній процес прозорішим, ефективнішим і орієнтованим на потреби учнів.

Автоматизація аналітики у тестуванні здобувачів освіти є ключовим фактором, що забезпечує швидкість, точність і ефективність аналізу великих обсягів даних. Завдяки автоматизованим системам освітні заклади можуть отримувати інформативні звіти та рекомендації для вдосконалення навчального процесу без значних витрат часу та людських ресурсів. Основні аспекти автоматизації:

1. Обробка великих обсягів даних:

У великих закладах освіти кількість результатів тестування може досягати тисяч. Автоматизація дозволяє обробляти ці дані без помилок і в короткий термін.

- Приклад: Використання СУБД (систем управління базами даних), таких як MS SQL Server, для зберігання та обробки результатів.

2. Швидке створення звітів:

Автоматизовані системи формують звіти одразу після завершення тестування. Це дозволяє миттєво оцінювати рівень знань учнів і приймати відповідні рішення.

- Приклад: Інтеграція з BI-платформами (Power BI, Tableau) для візуалізації результатів у вигляді графіків і діаграм.

3. Зниження людського фактору:

Автоматизація зменшує ризик помилок, які виникають під час ручного введення або аналізу даних. Це забезпечує більш точні результати та підвищує довіру до системи тестування.

4. Масштабованість:

Системи автоматизації легко адаптуються до зростання кількості учнів і обсягів тестування. Це особливо важливо для університетів, де одночасно навчаються тисячі студентів.

5. Реалізація складних алгоритмів:

Автоматизовані системи можуть використовувати складні аналітичні алгоритми для глибокого аналізу.

- Наприклад, аналіз когорт (груп учнів із схожими характеристиками), визначення трендів у результатах тестування.

Інструменти автоматизації

1. СУБД для зберігання даних:

- MS SQL Server, MySQL, PostgreSQL – системи для зберігання результатів тестів і швидкого доступу до них.

2. Програмне забезпечення для аналізу:

- BI-платформи: Power BI, Tableau.

- Інструменти аналізу даних: Python (Pandas, NumPy), R.

3. Розробка клієнтських додатків:

- Visual Studio, C# для створення інтерфейсів, які автоматично підключаються до баз даних і генерують звіти.

Переваги автоматизації у аналітиці

1. Швидкість: Час на обробку даних скорочується в десятки разів порівняно з ручним аналізом.

2. Точність: Автоматизовані системи виключають помилки під час підрахунків, аналізу чи створення звітів.

3. Універсальність: Можливість одночасного оброблення даних для багатьох предметів або груп.

4. Доступність: Автоматизовані системи часто інтегруються з хмарними платформами, що дозволяє викладачам і адміністраторам отримувати доступ до даних з будь-якого пристрою.

5. Гнучкість: Системи можуть адаптуватися до різних форматів тестування (закриті питання, відкриті відповіді, інтерактивні завдання).

Розширені можливості автоматизації

1. Прогнозування:

Використання алгоритмів машинного навчання для передбачення успішності учнів на основі їхніх попередніх результатів.

2. Адаптивне тестування:

Автоматизовані системи можуть динамічно підлаштовувати рівень складності питань залежно від попередніх відповідей здобувача освіти.

3. Інтеграція з іншими системами:

Синхронізація з електронними журналами, системами управління навчанням (LMS) та іншими освітніми платформами для централізованого зберігання даних.

4. Аналітика в реальному часі:

Викладачі отримують актуальну інформацію про прогрес учнів, що дозволяє швидко вносити корективи у навчальний процес.

Приклад автоматизованого рішення

Програмний додаток, розроблений за допомогою Visual Studio і MS SQL Server, може:

- Збирати результати тестування з різних джерел.
- Автоматично формувати аналітичні звіти, що включають показники успішності, помилки та середній час виконання тестів.
- Генерувати рекомендації для викладачів щодо покращення викладання певних тем.

Автоматизація аналітики у тестуванні є важливим кроком до підвищення якості освіти, оскільки вона дозволяє робити процес навчання більш об'єктивним, прозорим та адаптивним до потреб здобувачів освіти.

1.6 Загальна характеристика навчального закладу «Кадієвецький ліцей»

Програмний додаток, розроблений для аналізу та створення аналітичної системи результатів тестування здобувачів освіти, був впроваджений і протестований у «Кадієвецькому ліцеї». Нижче наведено коротку інформацію про цей навчальний заклад.

У 1842 році в селі Кадіївці було відкрито однокласне училище для хлопців, яке розміщувалося в холодній дерев'яній будівлі. У 1842–1844 роках тут навчалося близько 60–70 учнів, однак з 1885 року їх кількість значно зменшилася до 13 осіб.

22 листопада 1891 року в селі відкрили двокласну міністерську школу, до якої вже приймали й дівчат. До 1917 року кількість учнів становила 12–15 осіб. У 1920–1930 роках функціонувала двокласна початкова школа, яка поступово розширювалася:

- У 1930–1931 роках додано 4-й клас.
- У 1931–1932 роках відкрито 5–6 класи.
- У 1935–1939 роках школа стала семирічною.

У 1951 році школа отримала статус середньої, а в 1975 році переїхала до нового приміщення. З 12 січня 2021 року «Кадіївецька ЗОШ І–ІІІ ступенів» офіційно перейменована на «Кадіївецький ліцей».

Серед випускників ліцею є:

- 95 педагогів,
- 25 лікарів,
- 52 інженери,
- 45 працівників сільського господарства,
- 25 військовослужбовців,
- 10 працівників лісового господарства та представники інших професій.

Заклад регулярно дбає про безпеку та охорону праці. Раз на три роки проводиться навчання та перевірка знань працівників ліцею з охорони праці, пожежної безпеки, техніки безпеки та безпеки життєдіяльності. Форма перевірки – письмовий іспит за екзаменаційними білетами, результати якого оформлюються протоколом.

У ліцеї ведуться всі необхідні журнали для реєстрації інструктажів із питань охорони праці. Для працівників і учнів розроблено та впроваджено програму вступного інструктажу.

Контакти:

Директор – Серeda Інна Миколаївна

Електронна адреса: kadyivic_08.kpr@i.ua

КОАТУУ 6822483301 (КАДІЇВЦІ)

Графік роботи ліцею:

Таблиця 1.1 Графік роботи ліцею

Понеділок	8:00-17:00
Вівторок	8:00-17:00
Середа	8:00-19:00
Четвер	8:00-17:00
П'ятниця	8:00-17:00
Субота	Вихідний
Неділя	Вихідний

1.7 Організаційна структура навчального закладу «Кадисевецький ліцей»

Організаційна структура управління — це впорядкована система служб, відділів, підрозділів і посадових осіб, які перебувають у взаємозв'язку, підпорядкованості та виконують конкретні управлінські функції. У «Кадисевецькому ліцеї» структура управління має таку організаційну схему (Рисунок 1.1):

1. **Директор:** Очолює заклад освіти, здійснює управління всіма аспектами його діяльності та відповідає за освітню, фінансово-господарську й іншу діяльність. Права, обов'язки та відповідальність директора визначені законодавством і статутними документами закладу.

2. **Заступник директора:** Допомогає директору в керуванні закладом, виконуючи специфічні обов'язки, зокрема у сферах навчальної роботи, фінансів або роботи з учнями.

3. **Педагогічна та методична ради:** Розробляють та ухвалюють плани роботи закладу, освітні програми, оцінюють їхню ефективність. Також ці ради відповідають за вдосконалення методичного забезпечення освітнього процесу, розглядають питання переведення учнів до наступних класів, випуску, нагородження за досягнення, а також затверджують заходи та плани підвищення кваліфікації педагогічних працівників.

4. **Батьківська рада:** Група батьків і вчителів, що сприяє співпраці між батьками, педагогами та адміністрацією, підтримуючи учнів і заклад загалом. Її основна мета — забезпечення добробуту та успішності учнів.

5. **Учнівське самоврядування:** Представляє інтереси учнівського колективу, захищає їхні права та сприяє залученню учнів до управління шкільним життям.

6. **Соціальний педагог:** Професіонал, який допомагає учням, їхнім сім'ям і спільноті, сприяючи соціальному, емоційному та поведінковому розвитку дітей.

7. **Педагог-організатор:** Відповідає за організацію, координацію та проведення освітніх заходів і програм для учнів.

8. **Методичне об'єднання вчителів:** Група педагогів, яка працює над розробкою та впровадженням передових методик і новацій у викладанні.

9. **Класний керівник:** Відповідає за навчально-виховну роботу з конкретним класом або групою учнів, створюючи сприятливі умови для їхнього розвитку.

10. **Викладач:** Фахівець, який забезпечує навчальний процес, допомагаючи учням досягати освітніх цілей і формувати необхідні знання та навички.

Організаційна структура управління школою

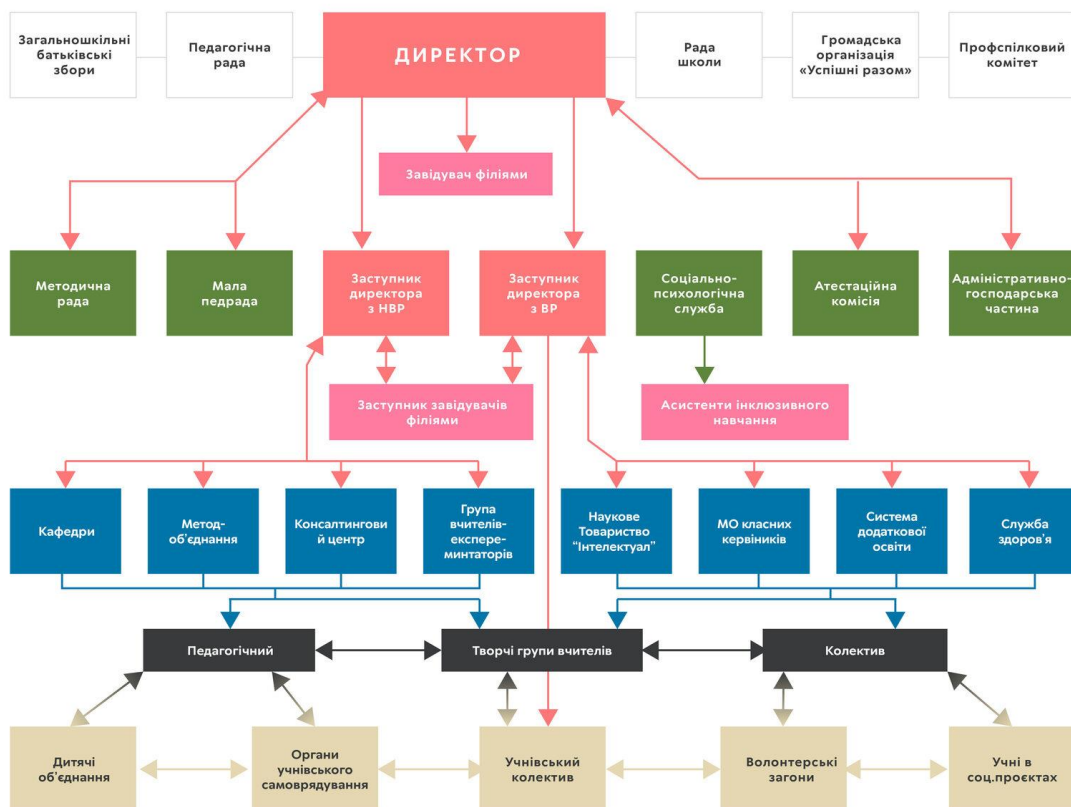


Рисунок 1.1 – Організаційна структура ліцею

1.8 Аналіз нинішнього стану комп'ютеризації Кадисєвського ліцею

Під час проходження виробничої та переддипломної практики в Кадисєвському ліцеї мною було помічено, що у закладі освіти досі не використовується програми для тестування та оцінювання учнів. Тестування та оцінювання учнів відбувається у звичайному форматі, це виконання письмових тестів, усне тестування після проходження навчальних тем і так далі. Після зібраної інформації та обговорення з керівництвом ліцею, мені було поставлене завдання створити програмний додаток, який буде забезпечувати автоматизоване тестування учнів, та їх оцінювання після проходження тесту, створення аналітичних звітів для зручного перегляду результатів тесту.

1.9 SWOT-аналіз діяльності Кадисвецького ліцею

S – Сильні сторони

Досвід та історія закладу: Ліцей має тривалу історію розвитку (заснований у 1842 році), що свідчить про стійкість і адаптацію до змін в освіті.

Кваліфікований педагогічний колектив: Випускники ліцею здобувають високий рівень знань, що підтверджується кількістю випускників із вищою освітою.

Нові приміщення: Сучасна будівля, що відповідає вимогам навчального процесу (збудована у 1975 році).

Систематичне підвищення кваліфікації: Регулярні тренінги, навчання й перевірки знань персоналу з питань охорони праці, техніки безпеки тощо.

Налагоджена структура управління: Чітка організаційна схема, включаючи педагогічну раду, батьківську раду та учнівське самоврядування.

Інтеграція інновацій: Використання сучасних технологій, зокрема програмних рішень для управління навчальним процесом.

W – Слабкі сторони

Обмежені ресурси: Недостатнє фінансування для впровадження масштабних інновацій, обладнання лабораторій та модернізації навчального середовища.

Відсутність сучасного спортивного або культурного центру: Може стримувати розвиток позашкільної діяльності.

Залежність від традиційних методів: Недостатнє впровадження інтерактивних форм навчання, таких як STEM-підходи чи проектна діяльність.

Мала кількість учнів у порівнянні з великими міськими школами: Це може зменшувати конкурентоспроможність ліцею.

Недостатня цифровізація: Потреба в більш широкому впровадженні електронних освітніх платформ та аналітичних інструментів.

O – Можливості

Впровадження IT-технологій: Використання програмних продуктів для аналізу результатів тестувань та управління навчальним процесом.

Партнерство з іншими освітніми установами: Співпраця з університетами чи освітніми платформами для залучення нових методик навчання.

Залучення додаткових ресурсів: Отримання грантів, благодійних внесків або державної підтримки на модернізацію матеріально-технічної бази.

Розвиток дистанційного навчання: Забезпечення доступу до якісної освіти навіть для тих учнів, які знаходяться поза межами громади.

Акцент на підвищення якості освіти: Можливість покращення результатів за рахунок роботи з обдарованими учнями та додаткових занять.

T – Загрози

Скорочення кількості учнів: Демографічна криза в сільській місцевості може вплинути на наповнюваність ліцею.

Конкуренція з міськими школами: Урбанізація та міграція родин до міст можуть зменшувати кількість учнів.

Недостатнє фінансування освіти: Зменшення державного або місцевого фінансування може негативно вплинути на розвиток закладу.

Застаріла матеріально-технічна база: Відсутність інвестицій у сучасне обладнання та технології може перешкоджати впровадженню інноваційних методів навчання.

Погіршення рівня освіти через соціальні фактори: Зниження мотивації учнів до навчання через економічні труднощі в регіоні або інші соціальні виклики.

1.10 Доцільність проектування та розробки системи

Розробка програмного додатку для дослідження та аналітики результатів тестування здобувачів освіти є актуальною і доцільною з огляду на сучасні потреби освітнього процесу. Ось ключові аспекти, які підтверджують її значущість:

- Підвищення ефективності управління освітнім процесом: Аналітична система дозволяє автоматизувати обробку результатів тестування, що значно зменшує час, витрачений на ручну роботу, та виключає помилки, пов'язані з людським фактором. Це дає можливість керівникам закладу швидше приймати обґрунтовані рішення щодо якості освітніх програм і процесів.

- **Об'єктивність та прозорість оцінювання:** Додаток забезпечує неупереджений аналіз результатів тестування. Використання автоматичних алгоритмів виключає суб'єктивний підхід до оцінювання знань учнів. Це сприяє формуванню довіри серед учнів, батьків і педагогів до результатів тестування.
- **Адаптація навчальних програм:** На основі аналітичних даних освітні програми можуть бути адаптовані до потреб учнів, що дозволить приділити більше уваги складним для них темам. Це сприяє підвищенню якості освіти загалом.
- **Забезпечення доступності аналітичної інформації:** Додаток може надавати зручні аналітичні звіти у вигляді графіків, діаграм і таблиць. Ці звіти є зрозумілими для всіх учасників освітнього процесу – від адміністрації закладу до вчителів, учнів та їхніх батьків.
- **Підтримка дистанційного навчання:** У сучасних умовах дистанційне навчання стало невід'ємною частиною освіти. Система аналітики дозволяє ефективно проводити тестування в онлайн-форматі, автоматизуючи збір і обробку результатів.
- **Підтримка академічної доброчесності:** Система допомагає виключити можливість підробки або маніпулювання результатами тестів, гарантуючи чесність оцінювання.

1.11 Економічний ефект від провадження системи для тестування

Оцінка економічного ефекту від використання програмного додатку у ліцеї включає аналіз як можливих вигод, так і витрат, пов'язаних із його впровадженням:

1. Зниження витрат.

Використання додатку дозволяє значно скоротити паперову документацію та ручну працю, зокрема у процесі тестування. Це сприяє зменшенню витрат на канцтовари, папір та інші ресурси, а також економії часу співробітників.

2. Додаткові джерела доходів.

Програмний додаток може стати платформою для залучення фінансової підтримки, наприклад, через онлайн-пожертви від випускників або зацікавлених осіб. Такі механізми дозволяють зміцнити бюджет закладу.

3. Можливості монетизації через рекламу.

Додаток може використовуватися для розміщення реклами місцевого бізнесу або партнерських організацій, створюючи нові потоки доходів для ліцею.

4. Підвищення ефективності роботи.

Автоматизація та централізований доступ до даних для учнів, батьків та персоналу дозволяють швидко знаходити потрібну інформацію, зменшуючи час на виконання рутинних завдань. Це підвищує загальну продуктивність закладу.

5. Зміцнення іміджу та репутації.

Використання сучасного програмного забезпечення покращує сприйняття ліцею серед учнів, батьків та потенційних партнерів. Успішне впровадження системи тестування може підвищити конкурентоспроможність ліцею, роблячи його привабливішим для майбутніх учнів.

Розробка та впровадження програмного додатку не лише оптимізує роботу ліцею, але й сприяє формуванню його сучасного іміджу, зміцнюючи економічні показники та розширюючи можливості для розвитку.

1.12 Висновок до першого розділу

У першому розгляді було виконано дослідження процесів тестування, виявлено його результативність та недоліки. І все ж, незважаючи на недоліки, можна говорити про те, що сьогодні стрімкого поширення набуває тестовий контроль навчально-пізнавальної діяльності учнів, оскільки даний метод перевірки знання потребує менше часу для написання, а при правильній його структурованості повноцінно розкриває набуті учнями знання. Перехід від традиційних форм освіти до пошуку нових, більш ефективних, потребує створення якісно нової моделі системи контролю, яка б дала змогу об'єктивно оцінювати знання, уміння та навички учнів, виключаючи суб'єктивізм.

РОЗДІЛ 2. ДОСЛІДЖЕННЯ ПРОБЛЕМИ ТА ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ

2.1 Науково-методичні основи розробки

Науково-методичні основи розробки аналітичної системи результатів тестування здобувачів освіти базуються на комплексному підході, що включає методологічні, технічні та педагогічні аспекти. Така система створюється для автоматизації процесу оцінювання, підвищення об'єктивності та точності аналізу, а також забезпечення ефективного зворотного зв'язку для студентів і викладачів.

Методологічний аспект передбачає розробку моделі оцінювання, яка враховує стандарти навчальних програм, визначає критерії та рівні компетентностей, а також забезпечує валідність і надійність тестових завдань. Технічна складова включає впровадження цифрових платформ для проведення тестувань, збереження даних і застосування алгоритмів для аналізу результатів. Наприклад, статистичні методи та машинне навчання використовуються для виявлення прогалин у знаннях і аналізу успішності студентів.

У педагогічному аспекті важливими є такі елементи, як індивідуальні рекомендації для студентів, візуалізація результатів через діаграми й графіки, а також надання викладачам інструментів для аналізу ефективності навчання. Завдяки такій системі можна вдосконалити навчальний процес, своєчасно виявляти проблемні зони в знаннях і забезпечувати студентам можливості для самовдосконалення.

У ході дослідження процесу тестування було розроблено функціональну модель алгоритму тестування та виконано його декомпозицію у програмному засобі ERwin Data Modeler, модель алгоритму зображено на (Рисунок 2.1 та Рисунок 2.2).

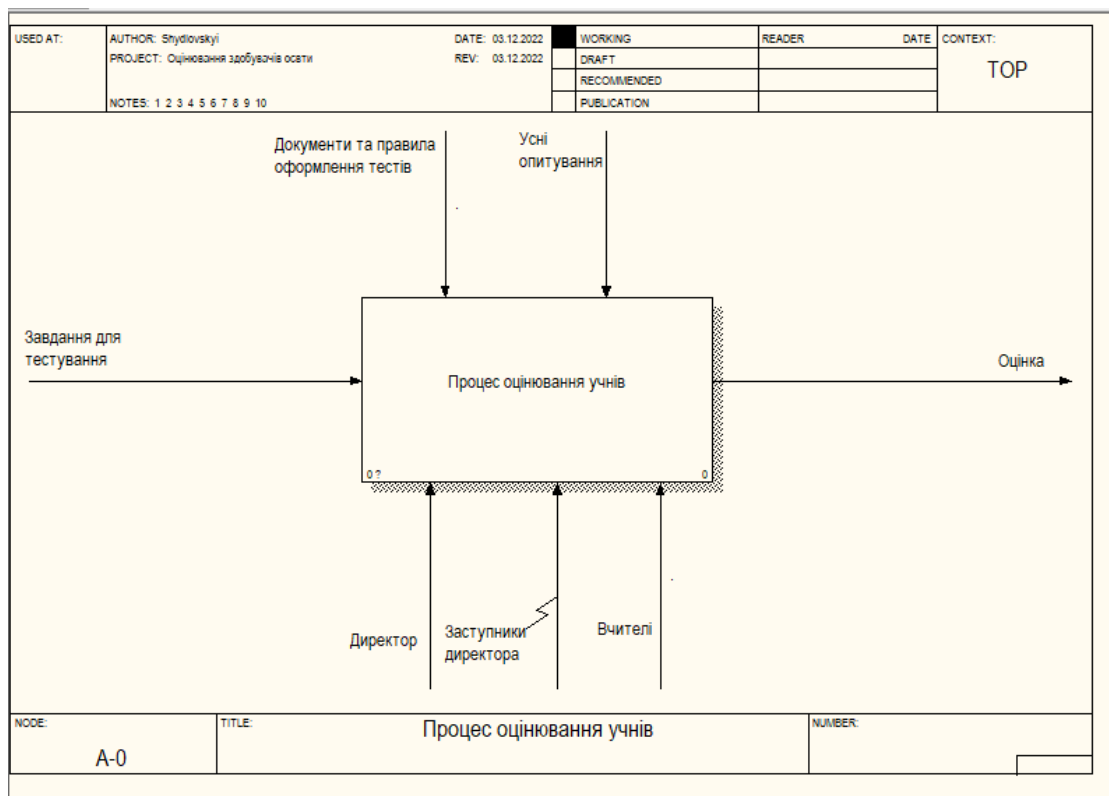


Рисунок 2.1 – Функціональна модель процесу тестування

Ця функціональна модель описує процес оцінювання учнів у навчальному закладі, зображений у вигляді діаграми уніфікованого процесу (IDEF0). Ось як можна інтерпретувати її основні компоненти:

1. Вхідні дані:

- Завдання для тестування: це підготовлені тестові матеріали, що використовуються для перевірки знань учнів.
- Документи та правила оформлення тестів: нормативно-методичні документи, які регламентують процедуру проведення тестувань (методичні рекомендації, вимоги до тестів).
- Усні опитування: альтернативний спосіб оцінювання, який включає запитання та відповіді в усній формі.

2. Головний процес:

- У центрі моделі розташовано блок "Процес оцінювання учнів", який відповідає за виконання оцінювання відповідно до завдань, документів і наданих інструментів.

3. Ресурси (підтримка):

- Директор: як керівник навчального закладу, він може надавати адміністративну чи організаційну підтримку процесу оцінювання.
- Заступники директора: координують і контролюють процес оцінювання, забезпечують методичну допомогу.
- Вчителі: безпосередньо виконують функцію оцінювання, проводячи тестування чи усні опитування.

4. Вихідні дані:

- Оцінка: результат оцінювання знань учнів, який є кінцевим продуктом процесу.

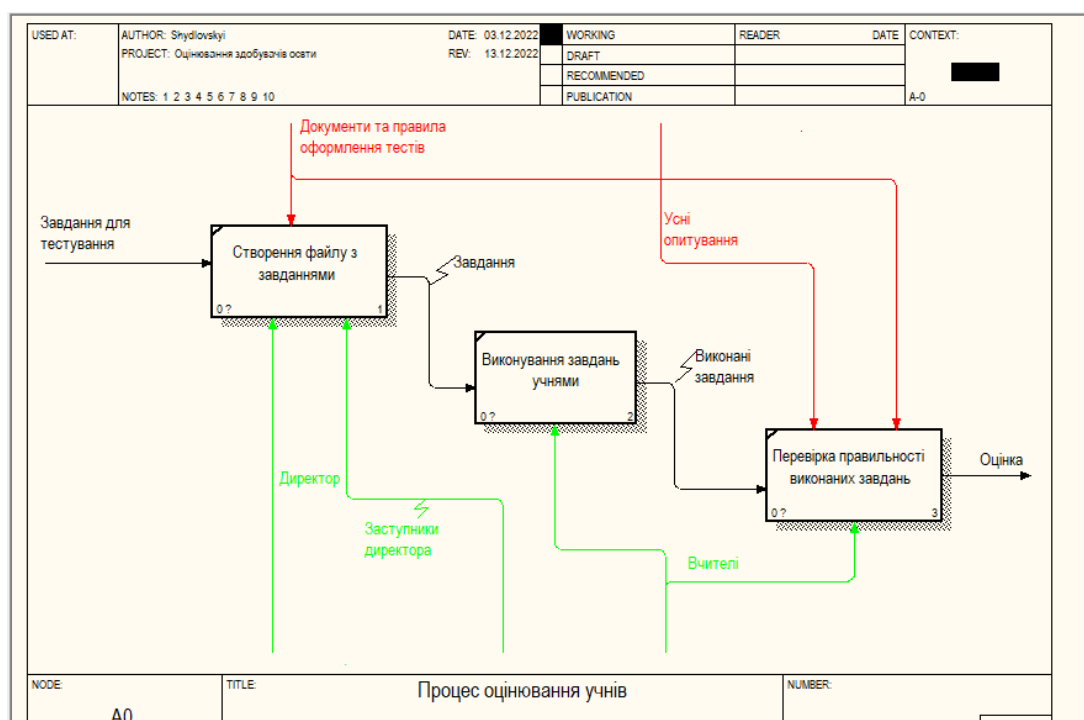


Рисунок 2.2 – Декомпозиція моделі процесу тестування

На представленій моделі показана декомпозиція процесу оцінювання учнів, яка деталізує основні етапи:

1. Створення файлу з завданнями: на основі завдань для тестування та документів із правилами оформлення формується набір завдань. Цей етап передбачає участь директора і заступників директора для контролю відповідності завдань нормативним вимогам.

2. Виконання завдань учнями: підготовлені завдання передаються учням для виконання. Учні працюють над ними відповідно до вказівок і формату.

3. Перевірка правильності виконаних завдань: викладачі оцінюють виконані учнями роботи (включаючи результати усного опитування) та формують підсумкову оцінку.

Ця декомпозиція відображає логічний ланцюг процесу оцінювання: підготовка завдань → виконання → перевірка, із залученням відповідних ролей (директор, заступники, вчителі).

Досліджуючи процес тестування здобувачів освіти було розроблено схему алгоритму системи для виконання тестування.

2.2 Методи аналізу результатів тестування

Серед основних підходів до аналізу результатів тестування виділяються статистичні методи, теорія Item Response Theory (IRT), когнітивно-орієнтований і психометричний аналіз.

1. Статистичні методи забезпечують вимірювання надійності та валідності тестів, дозволяють оцінити середній бал, дисперсію, внутрішню узгодженість тесту та виявляти зв'язки між результатами тестування та іншими показниками.

2. Item Response Theory (IRT) дає змогу оцінювати складність завдань і рівень компетенції здобувачів, враховуючи їх індивідуальні особливості. Цей підхід

допомагає налаштовувати тести так, щоб вони відповідали рівню підготовки кожного учасника.

3. Когнітивно-орієнтований аналіз використовує завдання, що оцінюють не лише знання, але й мисленнєві процеси, наприклад, за таксономією Блума. Він дозволяє адаптувати тестування під рівень здобувача для більш точного оцінювання.

4. Психометричний аналіз враховує психологічні фактори, такі як мотивація, стресостійкість та умови проведення тестування, що впливають на результати.

Для врахування індивідуальних особливостей здобувачів використовуються адаптивне тестування та формувальне оцінювання.

Адаптивне тестування автоматично налаштовує складність завдань відповідно до рівня знань учасника, що дозволяє підвищити точність оцінювання та зменшити стрес.

Формувальне оцінювання забезпечує постійний зворотний зв'язок, дозволяючи аналізувати помилки, знаходити прогалини у знаннях і вдосконалювати навчальний процес. Сучасні платформи використовують Learning Analytics для персоналізації навчання.

Інструменти аналізу результатів тестування

Системи управління навчанням, такі як Moodle чи Canvas, пропонують інструменти автоматичного аналізу результатів, визначення прогалин у знаннях і побудови звітів. Використання технологій великих даних і машинного навчання допомагає виявляти індивідуальні патерни успішності та прогнозувати майбутні досягнення.

На основі дослідження та аналізу процесу тестування було сформовано блок-схему алгоритму цього процесу. Сама блок-схема зображена у (Додаток В), та повністю описує процес при якому відбувається тестування знань та оцінювання здобувачів освіти.

2.3 Індивідуальні особливості здобувачів освіти

Індивідуальні особливості здобувачів освіти є важливим фактором, що визначає ефективність і об'єктивність тестування. Врахування цих особливостей дозволяє створити більш справедливую систему оцінювання і точніше відобразити реальний рівень знань і вмінь учнів. Кожен учень має унікальний набір характеристик, які можуть вплинути на його результати тестування.

Однією з ключових характеристик є рівень підготовки та знань. Учні можуть мати різний досвід у вивченні предметів, що впливає на їх здатність швидко засвоювати нову інформацію та ефективно виконувати завдання. Це також стосується їхніх умінь працювати з навчальним матеріалом, розуміти теоретичні концепції або застосовувати знання на практиці.

Когнітивні особливості учнів, такі як рівень уваги, швидкість обробки інформації та здатність до аналітичного мислення, можуть суттєво вплинути на тестування. Деякі учні можуть швидко обробляти інформацію та ефективно виконувати завдання, а інші – потребують більше часу для роздумів. Важливим є і вміння розв'язувати нестандартні задачі, що є важливим компонентом більшості тестів.

Мотивація та емоційний стан учнів також відіграють важливу роль. Учні з високою внутрішньою мотивацією зазвичай досягають кращих результатів, оскільки активно залучені в процес навчання. Водночас, емоційний стрес, пов'язаний із тестуванням, може знижувати результати, особливо у тих учнів, які мають тривожність або занадто сильно переживають через оцінки.

Врахування особливостей розвитку також важливе для об'єктивного тестування. Учні з фізичними або психологічними потребами можуть мати індивідуальні вимоги до тестування, наприклад, необхідність адаптації завдань чи додаткового часу. Це стосується не тільки фізичних обмежень, а й психологічних труднощів, таких як дислексія або аутизм.

Також важливими є культурні та соціальні фактори. Учні, які не є носіями мови тестування, можуть зустрітися з труднощами, які знижують точність оцінки їхніх знань. Культурні різниці можуть проявлятися у незрозумілості термінів або прикладів, що використовуються в тестах.

Зрештою, стиль навчання учнів також впливає на їхню здатність успішно пройти тестування. Деякі учні краще засвоюють інформацію візуально або через практичні вправи, інші – через слухове сприйняття. Врахування цих особливостей допомагає створити адаптовану систему оцінки, що дозволяє кожному учневі показати свій реальний рівень знань.

Враховуючи всі ці індивідуальні характеристики, можна створити більш справедливу систему тестування, яка точно відображає навчальні досягнення учнів. Це може бути досягнуто через адаптацію тестів, індивідуалізацію підходів до оцінювання, використання різних форматів тестування та надання додаткового часу або підтримки учням, які цього потребують.

2.4 Загальні відомості про зміст роботи

Дослідження та розробка аналітичної системи результатів тестування здобувачів освіти спрямована на оптимізацію освітніх процесів в навчальних закладах. Допоможе ефективно проводити процес тестування здобувачів освіти, збирати аналітичну інформацію по результатах тестування, автоматично визначати кількість правильних відповідей та обраховувати оцінку для кожного учня.

1. Предмет дослідження:

Процеси аналізу даних про результати тестування здобувачів освіти, включаючи збір, обробку та подання інформації для викладачів, адміністраторів закладів освіти та самих здобувачів.

2. Об'єкт дослідження:

Освітній процес, пов'язаний із проведенням, оцінкою та аналізом результатів тестувань здобувачів освіти в середніх та вищих навчальних закладах.

3. Основні задачі роботи:

- Аналіз поточного стану обробки результатів тестування в освітніх установах.
- Виявлення проблем і потреб в автоматизації процесів оцінки тестів.
- Обґрунтування вибору технологій для створення аналітичного інструменту.
- Розробка прототипу аналітичної системи для моніторингу результатів тестувань.
- Тестування додатку на реальних даних та оцінка його ефективності.

4. Очікувані результати:

- Зменшення часу на оцінку та аналіз результатів тестувань.
- Покращення точності, об'єктивності та доступності аналітичних звітів.
- Створення інструменту для виявлення прогалин у знаннях здобувачів освіти.
- Підвищення ефективності викладання через застосування даних для персоналізації навчання.

5. Методи дослідження:

- SWOT-аналіз існуючих методів аналізу результатів тестування.
- Вивчення сучасних підходів до розробки аналітичних систем в освіті.

- Проєктування алгоритмів автоматизованого збору та аналізу результатів тестів.

- Тестування системи на основі реальних даних із навчальних закладів.

6. Практичне значення роботи:

Розроблена система дозволить адміністраторам освітніх закладів отримувати комплексну та наочну інформацію про рівень знань здобувачів освіти, своєчасно виявляти проблемні аспекти в навчальному процесі та приймати обґрунтовані рішення щодо його покращення.

2.5 Сучасні технології для дослідження та розробки аналітичної системи результатів тестування здобувачів освіти

Сучасні технології для створення аналітичних систем, які аналізують результати тестування здобувачів освіти, охоплюють низку інструментів та підходів, що допомагають автоматизувати процеси збору, обробки і подання інформації. Розглянемо основні технології, які можна використати.

1. Інструменти для збору даних

Для збору результатів тестів використовуються прості й доступні сервіси, як-от:

Google Forms та Microsoft Forms: забезпечують зручний спосіб створення тестів і опитувань. Дані автоматично зберігаються у вигляді таблиць, які легко аналізувати.

Освітні платформи (Moodle, Canvas, Blackboard): системи управління навчанням, що мають вбудовані модулі для тестування, автоматичного оцінювання та збору статистики.

2. Зберігання даних

SQL-бази даних (MySQL, PostgreSQL): ідеальні для зберігання структурованих даних, наприклад, оцінок і метаданих про студентів.

NoSQL-бази даних (MongoDB, Firebase): використовуються для гнучкого зберігання великих обсягів даних, які можуть постійно змінюватися.

3. Обробка даних

Python: популярна мова програмування для роботи з даними. Її бібліотеки, такі як Pandas і NumPy, дозволяють аналізувати інформацію, а Matplotlib і Seaborn — створювати візуалізації.

R: ще один інструмент для статистичного аналізу, особливо якщо потрібні складні розрахунки або побудова прогнозів.

Big Data технології (Hadoop, Apache Spark): підходять для аналізу великих масивів даних, наприклад, коли потрібно обробляти результати тестів сотень тисяч студентів.

4. Візуалізація

Tableau та Power BI: ці інструменти допомагають створювати інтерактивні звіти та дашборди для відображення результатів тестування.

Google Data Studio: безкоштовний і простий інструмент для візуалізації даних з різних джерел.

5. Розробка веб-додатків

React.js, Vue.js або Angular: дозволяють створювати інтуїтивно зрозумілий інтерфейс для роботи з результатами тестувань.

Django (Python) або Node.js: забезпечують серверну частину, яка обробляє запити і забезпечує доступ до даних.

6. Хмарні технології

Google Cloud Platform, AWS, Microsoft Azure: ці сервіси пропонують хмарне зберігання, аналітику та навіть інструменти для машинного навчання.

Firebase: використовується для реального часу, наприклад, щоб одразу бачити результати тестів.

7. Штучний інтелект та адаптивне тестування

Системи адаптивного тестування, як-от Computerized Adaptive Testing (CAT), змінюють складність завдань залежно від відповідей студента, забезпечуючи індивідуальний підхід.

Штучний інтелект дозволяє аналізувати відкриті відповіді студентів за допомогою обробки текстів (NLP) або прогнозувати їхню успішність.

8. Мобільні технології

Flutter і React Native: ці інструменти використовують для створення мобільних додатків, які дозволяють студентам і викладачам працювати з результатами тестів зі смартфонів.

9. Безпека даних

Використання стандартів, таких як GDPR, забезпечує збереження конфіденційності даних. Шифрування за допомогою SSL/TLS захищає інформацію від несанкціонованого доступу.

Ці технології дозволяють створювати сучасні, зручні й ефективні системи для аналізу тестових результатів, що сприяють підвищенню якості навчання..

2.6 Огляд існуючих рішень

На сьогоднішній день програми для контролю знань здобувачів освіти представлені у доволі обмеженій кількості. Більшість із них реалізовані у форматі

вебсайтів, кожен із яких має свої переваги та недоліки. Одним із найпоширеніших програмних продуктів для тестування є MyTestX.

MyTestX – це комплексна система, яка включає програму для тестування учнів, редактор тестів і журнал результатів. Вона дозволяє створювати, проводити тестування, аналізувати отримані результати та виставляти оцінки за заданою шкалою.

Програма підтримує дев'ять типів завдань:

- одиночний вибір,
- множинний вибір,
- встановлення порядку,
- встановлення відповідності,
- визначення істинності/хибності тверджень,
- введення чисел,
- введення тексту,
- вибір місця на зображенні,
- перестановка букв.

Завдання формату "так/ні" легко реалізуються через одиничний вибір. У тестах можна комбінувати будь-які типи завдань у довільній кількості або використовувати лише один тип.

Програма складається з трьох основних модулів:

MyTestStudent – модуль для проходження тестів учнями. Цей інструмент має інтуїтивно зрозумілий інтерфейс і дозволяє зручно організувати тестування, зберігати результати та передавати їх учителю.

MyTestEditor – редактор для створення та редагування тестів. Завдання можуть містити текст питань, варіанти відповідей, правильні відповіді та ілюстрації.

MyTestServer – серверний модуль для обробки результатів, роздачі тестів через мережу, прийому й обробки даних за допомогою протоколу TCP/IP.

Окрім основних модулів, програма пропонує MyTestBuilder – додатковий інструмент для створення автономних тестів у форматі EXE.

Режими роботи програми:

Навчальний: учень отримує повідомлення про помилки, може побачити вступ або пояснення до завдань.

Штрафний: за неправильні відповіді знімаються бали, але є можливість пропустити завдання без штрафу.

Вільний: учень має змогу самостійно переходити між питаннями у довільному порядку.

Монопольний: програма займає весь екран, не дозволяючи згорнути або відкрити інші вікна.

Можливості використання:

Програма MyTestX підтримує локальне та мережеве тестування. У випадку мережевої роботи результати передаються до модуля Журнал через локальну мережу або електронну пошту. Це забезпечує гнучкість і зручність у використанні як в окремих класах, так і в масштабах навчального закладу.

Результаты 5A+.mtr - Журнал тестирования MyTest

Файл Сервер Вид Анализ Настройка Справка

Результаты Монитор Сообщения Лог Раздать тест Раздать список

Анализ тестирования по заданиям

Тест: "Измерение информации".
Кол-во тестируемых: 10.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Башлакова Саша	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Иванов Коля	+	+	+	+	+	+	-	-	+	-	+	-	+	-	+	+
Петров Вася	+	-	+	-	+	+	+	+	+	+	+	+	+	+	+	+
Путин Вова	+	+	-	-	+	+	-	-	+	+	+	+	+	+	+	-
Кинчев Костя	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Кошкина Настя	+	-	+	+	+	+	+	+	+	-	-	-	+	+	+	+
Мионов Андрей	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Орлова Люба	-	-	-	-	+	+	+	+	+	-	-	+	+	-	-	+
Зелёная Рина	+	+	+	+	+	+	-	-	+	-	+	-	+	-	+	+
Папанов Толик	+	-	+	-	+	+	+	+	+	+	+	+	+	+	+	+
Правильно	8	5	8	6	10	10	7	7	10	6	8	7	10	7	9	9
Частично	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ошибка	2	5	2	4	0	0	3	3	0	4	2	3	0	3	1	1
Пропущено	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Подсказок	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ср. время	00:03	00:02	00:01	00:01	00:01	00:01	00:01	00:01	00:01	00:01	00:01	00:01	00:01	00:01	00:01	00:01
Ср. балл	0,8	0,5	0,8	0,6	1,0	1,0	0,7	0,7	1,0	0,6	0,8	0,7	1,0	0,7	0,9	0,9
Рез-сть	80%	50%	80%	60%	100%	100%	70%	70%	100%	60%	80%	70%	100%	70%	90%	90%

Закреть

Рисунок 2.1 – Интерфейс программы MyTestX

Розмір інсталяційного файлу з програмою менше 3 Мбайт, а разом з довідкою, зразками тестів, близько 5 МБ (залежить від кількості тестів в комплекті).

Перевагами програми є:

- Малий розмір інсталяційного файлу.
- Широкий спектр вибору завдань.
- Підтримка декількох режимів роботи.
- Підтримка мережевого тестування.

Недоліками програми є:

- Розробка якісних тестових завдань — тривалий, трудомісткий процес у діяльності викладача.

○ Використання програми в навчальному режимі для повідомлення правильності / неправильності його відповіді також сприяє психологічному напруженню студента (декілька неправильних відповідей викликають паніку й формують впевненість у незадовільному результаті).

Наступною найбільш вживаною є програма – Майстер-Тест (Рисунок 1.6).

Безкоштовний освітній сервіс, орієнтований на створення тестів і проведення онлайн тестування в навчальних цілях. Дуже добре підходить для організації тематичного та контрольного оцінювання знань. Інтерфейс ресурсу представлений українською, англійською, російською мовами.

Процес реєстрації гранично простий - вписуєте ім'я, прізвище, електронну пошту і пароль, далі підтверджуєте реєстрацію, вказуєте роль (викладач або студент) - і можете починати створення першого тесту.

Натискаєте на «Створити перший тест зараз» або в розділі «Мої тести» на «Створити новий тест». У новому вікні вгорі записуємо текст питання і при необхідності додаткову роз'яснювальну інформацію. Далі вибираємо тип питання, а саме: одну правильну відповідь, кілька відповідей, слово, номер або відповідність.

Кількість рядків з варіантами відповідей можна збільшувати, натиснувши на «Додати відповідь». «Вага питання» означає кількість балів за одне питання. У віконцях навпроти за бажанням позначаємо пташкою правильну відповідь. В кінці натискаємо «Готово» і переходимо до наступного питання.

Щоб зареєструвати учнів, в розділі «Мої студенти» клікаєте на зелену кнопку вгорі з написом «Додати студента (отримати код)». Кожен учень повинен заздалегідь зареєструватися за допомогою своєї електронної пошти. Учитель надішле запрошення в свою групу на email учня або може надати свій код викладача.

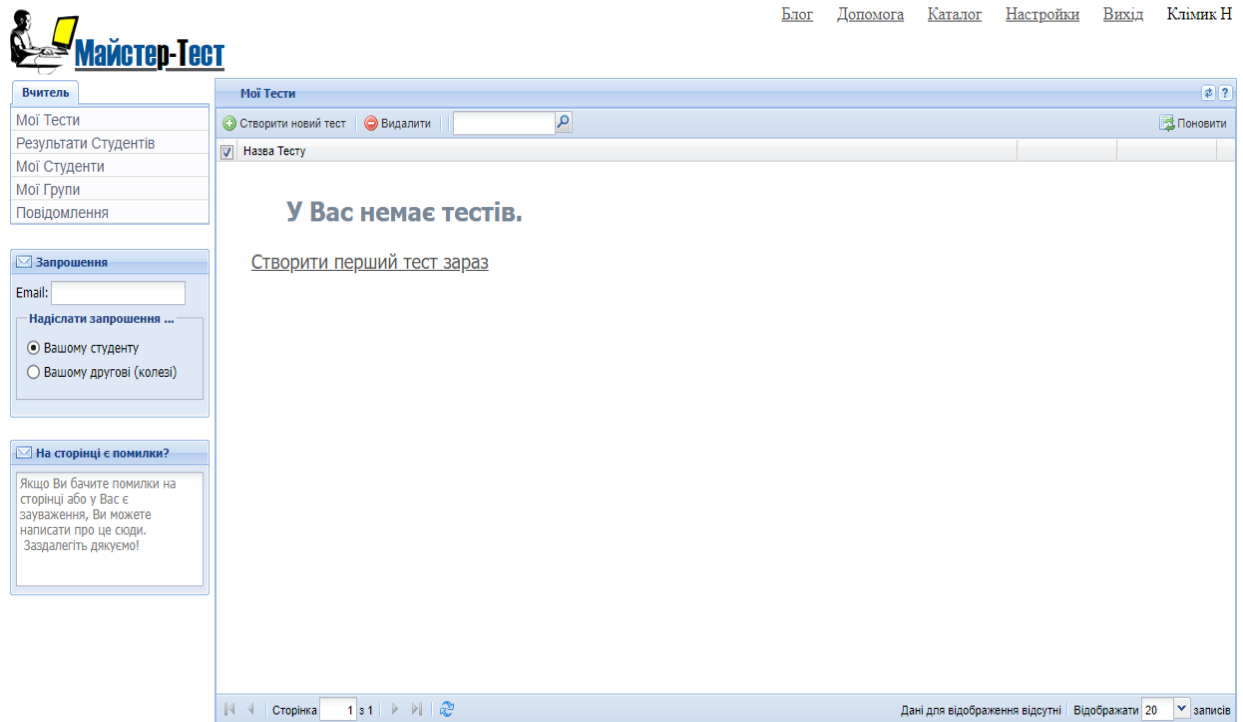


Рисунок 2.2 – Інтерфейс програми Майстер-Тест

Перевагами програми є

- Безкоштовність.
- Наявність реєстрації користувача.
- Стабільність роботи.

Недоліками програми є:

- Труднощі виконання завдань перехресного вибору.
- Немає можливості багаторазового виконання тесту.

CLASSMARKER – англomовний сервіс для швидкого конструювання тестових завдань і опитувань з найбільш широким форматом відповідей (Рисунок 1.7). Педагог може створювати і редагувати свої тести, зберігаючи їх у банк завдань в своєму профайлі. Учитель створює свій клас за допомогою розсилки реєстраційних даних учнів на email кожного учня або відправки реєстраційного коду.

Безкоштовний варіант користування допускає виробництво не більше 100 тестів. Розширені функціональні пакети коштують 16.50 \$ за 400 тестів на місяць або 33 \$ за 1000 тестів в місяць.

За умови використання браузера Google Chrome текст платформи можна автоматично перекласти російською.

Даний ресурс також дозволяє продавати свої тести онлайн з оплатою комісії в розмірі 10% від зазначеної вами суми.

Для створення облікового запису натискаємо на кнопку «Реєстрація / Register» вгорі праворуч. Виберіть вікно «Адміністратори, Для освіти / Administrators, For Education» і натисніть на значок оранжевого круга.

Спочатку пропонується вибрати тип питання - множинний вибір, правда / брехня, відповідність, текстовий рядок, виправлення помилок або есе. До питання можемо додати зображення, аудіофайл, формулу або відео. Далі визначаємося з кількістю варіантів відповіді. Є можливість також включити питання для зворотного зв'язку, кількість балів, оцінну шкалу.

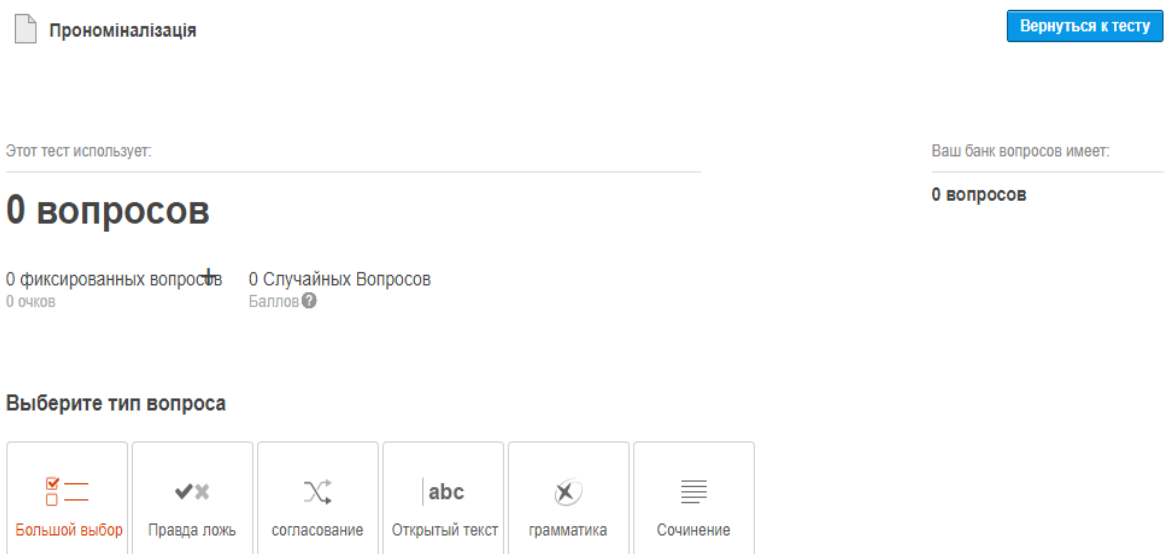


Рисунок 2.3 – Головне вікно програми CLASSMARKER

Перевагами програми є:

- Надійність роботи.
- Постійна підтримка програми розробниками.
- Великий вибір різноманітності тестів.

Недоліками програми є:

- Велика вартість програми (33 \$ за 1000 тестів в місяць).
- Обов'язкова реєстрація як і вчителя так і учнів.

2.7 Порівняння із запропонованими рішеннями

Виходячи з порівнянь програмних додатків які були описані вище, розроблювальний додаток для тестування та аналітики результатів здобувачів освіти буде більш гнучким в використанні, буде мінімізовано процес реєстрації учня, швидкий запуск тесту, додаток буде працювати офлайн без доступу до Інтернет.

Переваги запропонованого рішення:

- Спрямованість на освітні потреби:

Система орієнтована на базові процеси роботи з тестовими завданнями: створення тестів, аналіз результатів, формування звітів. Це робить її зручною у використанні як для викладачів, так і для студентів.

- Гнучкість налаштувань:

Користувачі матимуть змогу адаптувати функціонал під свої освітні потреби, від обрання типів завдань до створення індивідуальних шкал оцінювання, уникаючи переплат за зайвий функціонал.

- Візуалізація освітніх даних:

Система інтегрує інструменти візуалізації, що дозволяє викладачам та адміністрації аналізувати результати тестувань за допомогою графіків, діаграм і дашбордів. Це значно спрощує процес оцінки навчального прогресу.

- Доступність і простота використання:

Завдяки веб-додатку, система буде доступною через будь-який пристрій із доступом до інтернету. Інтуїтивно зрозумілий інтерфейс полегшить навчання та адаптацію користувачів.

Недоліки порівняно з ринковими аналогами:

- Обмежена інтеграція з зовнішніми платформами:

У той час як більші системи пропонують інтеграцію з LMS чи іншими освітніми платформами, запропоноване рішення орієнтоване переважно на внутрішнє використання.

- Початкове налаштування:

Хоча система передбачає високу гнучкість, її налаштування може займати більше часу, ніж готові ринкові рішення.

- Доцільність впровадження:

На ринку існує ряд систем для освітньої аналітики, однак більшість із них або занадто складні, або перевантажені функціоналом, який не завжди відповідає потребам конкретного навчального закладу.

Запропонована система стане оптимальним рішенням для закладів освіти, які потребують простого інструменту для створення тестів і аналізу результатів, з урахуванням їхніх індивідуальних потреб.

Переваги запропонованої аналітичної системи:

- Автоматизація процесів тестування:

Система дозволить створювати й проводити тести в цифровому форматі, з автоматичною перевіркою відповідей, знижуючи навантаження на викладачів.

- Динамічний аналіз результатів:

Викладачі отримуватимуть інтерактивні дашборди з ключовими показниками, такими як середній бал, кількість правильних відповідей, рівень успішності кожного студента.

- Персоналізація навчання:

Завдяки аналітичним даним система допоможе визначити слабкі місця у знаннях студентів, що сприятиме їхньому індивідуальному розвитку.

- Масштабованість:

Система буде придатною як для невеликих шкіл, так і для великих навчальних закладів, дозволяючи поступово додавати нові функції та розширювати можливості.

- Цифрова доступність:

Зручність роботи із системою з будь-якого пристрою сприяє її використанню в умовах дистанційного навчання.

- Актуальність в умовах цифрової трансформації освіти:

У сучасному світі цифрових технологій освітні заклади активно впроваджують автоматизовані системи для поліпшення якості навчання. Розробка аналітичної системи для тестування здобувачів освіти є необхідним кроком для підвищення ефективності навчального процесу, автоматизації рутинних завдань і поліпшення прийняття рішень на основі даних.

Така система допоможе зробити процес тестування більш прозорим, швидким і об'єктивним, що важливо для розвитку сучасної освіти.

2.8 Загальні вимоги до програмного додатку

Розроблений програмний додаток для тестування має складатися з кількох функціональних форм (вікон) з можливістю швидкого перемикання між ними. Додаток створюється мовою програмування C# із використанням середовища Microsoft Visual Studio 2019. Для зберігання даних про вчителів, учнів і тести буде використовуватися база даних MySQL.

Основні компоненти та функціональні можливості додатку включають:

- Сервери для імпорту та збереження даних.
- Ресурси програми, доступні для користувачів.
- Установку через інсталяційний файл (Setup.exe).
- Стабільну роботу програми.
- Головне меню з інструментами для управління функціями додатку.
- Налаштування часу та вибір тестів для проходження.
- Додаткові вікна для введення нових даних про учнів, вчителів та створення тестів.
- Підсвічування правильних відповідей зеленим кольором, а неправильних — червоним.
- Автоматичний підрахунок правильних відповідей і відображення результатів із зазначенням ПБ учня та його оцінки.
- Генерацію звітів про результати тестування.
- Обмін даними з серверною СУБД SQL.
- Розроблення аналітичної форми для відображення результатів тестування
- Створення звітностей на аналітичній формі

2.9 Архітектура рішення

Програмний додаток передбачає наступні компоненти:

1. База даних (БД):
 - Сервери для зберігання інформації про учнів, вчителів і тестові завдання.

2. Клієнтський рівень:

- Інтерфейси для введення, редагування та збереження даних у базі.
- Вікна для перегляду та обробки інформації.

Сервери бази даних розгортаються на всіх комп'ютерах навчального закладу, де проводитиметься тестування. Прикладні сервери забезпечують реалізацію основного функціоналу програми та захищають базу даних від зовнішніх впливів.

Клієнтські робочі місця створюються на ПК або ноутбуках, підключених до локальної мережі. Для роботи з додатком потрібна наявність **NetFramework 3.2** або новішої версії.

Інтерфейс програми має підтримувати українську мову, а нормативні довідники повинні також бути українською.

2.10 Вимоги до гарантійної підтримки

Програмне забезпечення, розгорнуте в межах проекту, має супроводжуватися гарантійною підтримкою протягом трьох місяців з моменту впровадження. Підтримка включає виправлення помилок і оновлення версій додатку. Вартість гарантійної підтримки входить до загальної ціни розробки.

2.11 Вимоги до інформаційного обміну

Обмін інформацією між компонентами системи має здійснюватися за допомогою стандартних протоколів (на рівні програмного забезпечення або платформи).

2.12 Режими роботи

Додаток повинен функціонувати в режимі 24/7 без потреби регулярних зупинок для технічного обслуговування. Він має підтримувати розгортання на кількох серверах, забезпечувати резервування даних і балансування навантаження.

2.13 Лінгвістичні вимоги

Інтерфейс користувача має бути виконаний українською мовою. Для технічних процедур допускається використання англійської мови, що має бути зазначено в документації.

2.14 Вимоги до програмного забезпечення

Програмний додаток має базуватися на вільно розповсюджваному програмному забезпеченні:

- Операційна система: Windows 10 або її аналоги.
- Мова програмування: MS Visual C# або аналогічна.
- Система для створення установочних файлів: Inno Setup Compiler або її аналоги.
- Система керування БД: MySQL або її аналоги.
- Система керування репозиторіями: GitLab або інша подібна.

2.4.8 Технічні вимоги

Мінімальні параметри ПК для роботи з додатком:

- Процесор: Intel Pentium 2266 MHz або кращий.
- Графічний адаптер: SVGA або новіший.
- Оперативна пам'ять: від 256 Мб.

- Дисковий простір: щонайменше 150 Мб + 200 Мб для Java Runtime Environment і NET Framework.
- Пристрої вводу: клавіатура та миша.

2.15 Організаційні вимоги

Для розробки програмного додатку Замовник надає Розробнику всі необхідні матеріали й інформацію для інтеграції з іншими системами.

2.16 Вимоги до розробки та передачі додатку

Обов'язки Розробника:

- Підготовка технічного завдання.
- Розробка програмного забезпечення.
- Установка компонентів системи.
- Налаштування та введення в експлуатацію.
-

2.17 Вимоги до структури системи

- Модульна архітектура: Система повинна складатися з автономних модулів, кожен з яких виконує окрему функцію, наприклад, обробку даних, генерацію звітів чи управління результатами тестів. Це забезпечить гнучкість і легкість модернізації.

- Користувацький інтерфейс: Інтерфейс має бути розділений на функціональні частини, такі як аналіз результатів, адміністрування тестів та управління користувачами. Передбачена можливість налаштування інтерфейсу для різних ролей (викладач, адміністратор, учень).

- Інтеграція з API: Веб-додаток повинен мати API для зручного обміну даними з іншими освітніми системами, такими як Moodle чи Google Classroom.
- Безпека даних: Захист даних повинен включати аутентифікацію, авторизацію та шифрування, а також механізми для виявлення та попередження атак.
- Масштабованість: Система має підтримувати розширення функціоналу і обробку збільшених обсягів даних у випадку зростання навчального закладу чи введення нових дисциплін.

2.18 Вимоги до функціоналу системи

- Простота використання: Інтуїтивно зрозуміле меню дозволяє викладачам швидко створювати тести, аналізувати результати й генерувати звіти.
- Розподіл за ролями: Інтерфейс налаштовується під конкретні потреби користувачів: викладачі отримують доступ до створення тестів і перегляду результатів, учні — до проходження тестування, адміністратори — до управління всією системою.
- Автоматизоване тестування: Система дозволяє створювати тести з різними типами завдань (одиначний вибір, множинний вибір, відкриті питання).
- Аналіз результатів: Інструменти для фільтрації, сортування та аналізу результатів тестів. Генерація прогнозів успішності учнів на основі їхніх результатів.
- Звіти: Підтримка створення деталізованих і узагальнених звітів із можливістю експорту в різні формати (PDF, Excel).
- Дашборди: Інтерактивні панелі з ключовими показниками (середній бал, прогрес, найчастіші помилки).

- Прогнозування: Використання історичних даних для прогнозу успішності учнів чи результатів навчання.
- Інтеграція: Можливість автоматичного обміну даними з іншими системами (ERP, CRM, LMS).
- Безпека: Система повинна забезпечувати багаторівневу авторизацію, розмежування доступу до даних та регулярне резервне копіювання.

Ці вимоги формують основу для розробки сучасної аналітичної системи, яка дозволяє ефективно проводити тестування, аналізувати його результати та підвищувати якість навчального процесу.

2.19 Вимоги до збереження даних

Структура збережених даних: Система повинна забезпечувати збереження повного набору інформації, включаючи:

- Результати тестування учнів (кількість правильних відповідей, час проходження тестів).
- Дані про учнів (ПІБ, клас, групу, індивідуальні ідентифікатори).
- Інформацію про тести (зміст питань, типи завдань, шкала оцінювання).
- Дані про викладачів (ПІБ, дисципліни, роль у системі).
- Лог-файли діяльності користувачів для відстеження активності в системі.

Типи даних: Система повинна підтримувати збереження як структурованих даних (таблиці з результатами, метадані тестів), так і неструктурованих (зображення, текстові файли, звіти у форматах PDF/Excel).

Повнота даних: Для забезпечення точності та надійності звітності слід зберігати журнали активності, такі як час входу в систему, зміни тестів або спроби проходження тестувань.

2.20 Висновок до другого розділу

В другому розділі кваліфікаційної роботи було розглянуто перелік існуючих рішень з програмних додатків які актуальні на даний час, розроблено порівняння з існуючими програмними додатками які існують на ринку. Далі було досліджено процеси та функції які мають бути реалізовані в системі тестування здобувачів освіти, прописано вимоги до додатку та його наступна реалізація.

РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ

3.1. Основні задачі автоматизації

Автоматизація освітніх процесів, зокрема аналізу результатів тестування, дозволяє значно скоротити час на рутинні завдання, підвищити точність обробки даних і забезпечити зручність роботи з ними. Враховуючи специфіку роботи закладів освіти, основними завданнями автоматизації є:

- Зменшення помилок при обробці даних:

Введення результатів тестування вручну є трудомістким процесом, який може призводити до помилок та затримок. Автоматизація цього етапу дозволить уникнути неточностей, прискорити обробку інформації та забезпечити своєчасний доступ до результатів для освітян і адміністраторів.

- Інтеграція різних систем:

Заклади освіти часто використовують окремі платформи для тестування, управління навчальними планами та ведення звітності. Об'єднання цих даних в єдину систему забезпечить повну картину успішності учнів, дозволяючи аналізувати прогрес і якість навчання на основі всіх доступних даних.

- Швидкий доступ до аналітики та звітів:

Система має надавати зручний доступ до звітів, що дозволить викладачам та адміністраторам оцінювати прогрес учнів, виявляти слабкі місця у навчальному процесі, прогнозувати результати та вчасно коригувати стратегії навчання.

- Підтримка прийняття рішень:

Використовуючи сучасні аналітичні інструменти, система допоможе викладачам розробляти ефективніші підходи до навчання, а адміністраторам — оцінювати ефективність роботи закладу в цілому.

- Зниження навантаження на викладачів:

Завдяки автоматизації рутинних задач, таких як перевірка тестів, формування звітів та введення даних, викладачі зможуть зосередитися на навчанні, а не на адміністративній роботі. Це також знизить ризик людських помилок.

- Аналіз і персоналізація навчання:

Система автоматично збирає й обробляє дані про успішність учнів, що дозволяє формувати індивідуальні рекомендації для кожного учня. Це сприяє покращенню навчального процесу та підвищенню рівня знань.

- Моніторинг якості навчального процесу:

Завдяки постійному аналізу результатів тестування та відповідності стандартам освітніх програм, система дозволяє знижувати кількість помилок у тестах, підвищувати якість завдань та забезпечувати дотримання освітніх вимог.

Впровадження автоматизованої системи для аналізу результатів тестування сприятиме зменшенню витрат часу, підвищенню ефективності управління навчальними процесами та оперативності прийняття рішень. Автоматизація дозволить оптимізувати роботу закладів освіти, покращити якість навчання та підтримувати високий рівень успішності здобувачів освіти.

3.2 Автоматизація та алгоритмізація комплексу задач

Застосування інтегрованих рішень для автоматичного отримання даних із різних джерел (систем управління навчальними процесами, баз даних учнів і тестових платформ).

Перевірка отриманих результатів на відповідність заданим стандартам для уникнення помилок. Агрегація даних у реальному часі для забезпечення актуальності звітів.

З'єднання аналітичної системи з іншими платформами за допомогою API. Реалізація ETL (Extract, Transform, Load) для обробки, трансформації та збереження даних у потрібному форматі.

Адаптація інформації до єдиного формату для подальшого аналізу й обробки, забезпечуючи сумісність між системами.

Генерація звітів за допомогою SQL-запитів, а також візуалізація даних у Power BI або Tableau для створення зрозумілих графіків і дашбордів.

Система повинна підтримувати автоматичне створення звітів у форматах PDF і Excel для зручності аналізу та подальшого використання.

Використання алгоритмів ШІ для аналізу даних про успішність учнів, прогнозування результатів і надання рекомендацій щодо покращення навчального процесу.

Реалізація шифрування даних, аудит доступу та налаштування системи резервного копіювання для захисту інформації й запобігання її втратам.

Впровадження таких алгоритмів значно підвищить ефективність автоматизації процесів тестування, забезпечить високу точність обробки даних і створить зручну систему для аналізу та прийняття рішень в освітніх установах.

3.3 Розробка програмного додатку

Для виконання поставлених завдань на кваліфікаційний проект, мною було обрано такі програмні засоби як:

1. CASE засіб AllFusion ERWin Data Modeler – створення атрибутивної моделі бази даних проекту.
2. MS SQL Server 2022 – генерація фізичної бази даних.
3. Microsoft Visual Studio 2019 – створення клієнтського додатку.
4. Adobe Photoshop 2019 – створення логотипу програми.

5. Inno Setup Compiler – створення установочного файлу.

Найпершим етапом у розробці програмного додатку було створення бази даних. Для цього було використано програмний засіб AllFusion ERWin Data Modeler для побудови логіко-фізичної моделі даних. Сама модель даних зображена в Додатку Б.

Після цього потрібно було згенерувати базу даних у СУБД MS SQL Server 2022, за допомогою SQL-коду, який був попередньо згенерований за допомогою засобу Forward Engineer/Schema Generation у ERWin (Рисунок 3.1).

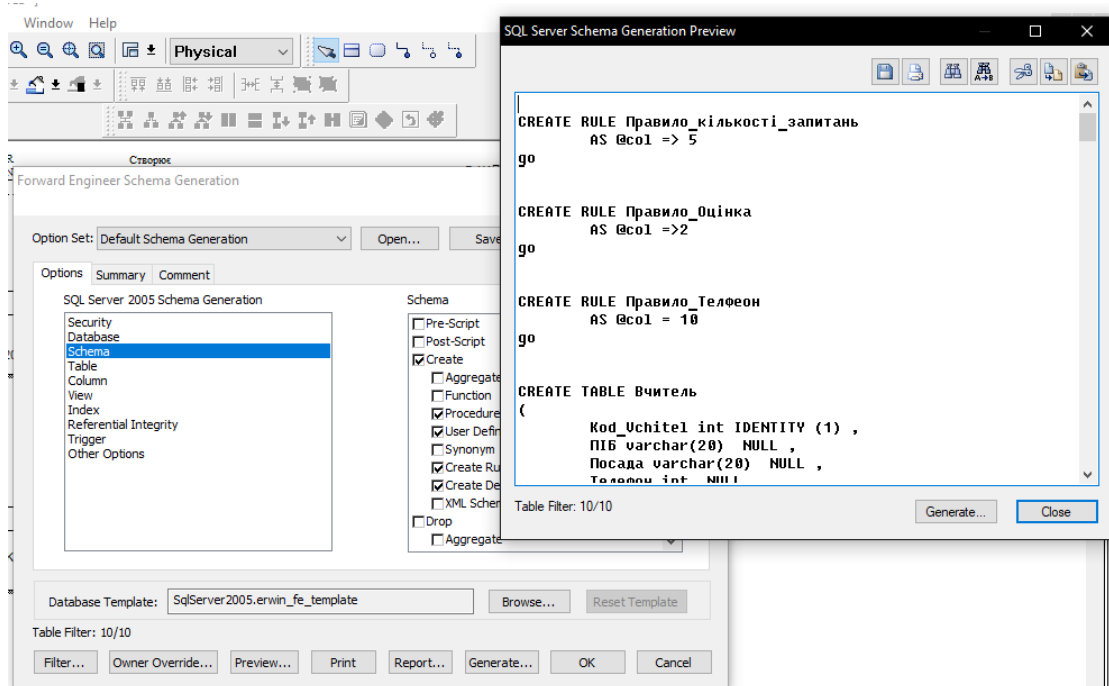


Рисунок 3.1 – Генерація SQL-коду

Генерація структури БД на основі створеного SQL коду відбувається після натиснення кнопки Generate. Діалог зв'язку з БД і виконання SQL коду відбувається в результаті натиснення кнопки Connect. Отримуємо згенеровану базу даних в середовищі MS SQL Server 2022. Вигляд схеми даних зображений у Додатку Б.

Після створення бази даних ми ініціюємо новий проєкт Windows Forms у середовищі розробки Visual Studio з використанням мови програмування C#. Усі

попередньо створені таблиці бази даних інтегруються з формами (вікнами) програми для виконання операцій із даними: додавання, видалення, редагування та перегляду записів, які зберігаються у базі даних.

Для підключення бази даних до середовища Microsoft Visual Studio 2022 застосовується компонент "Джерело даних" (див. Рисунок 3.2 і Рисунок 3.3).

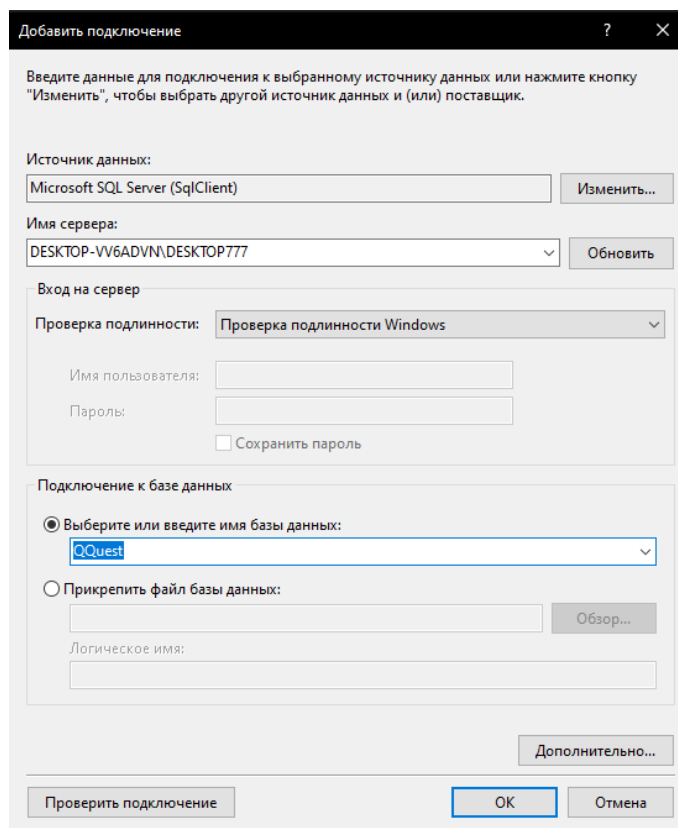


Рисунок 3.2 – Підключення до БД

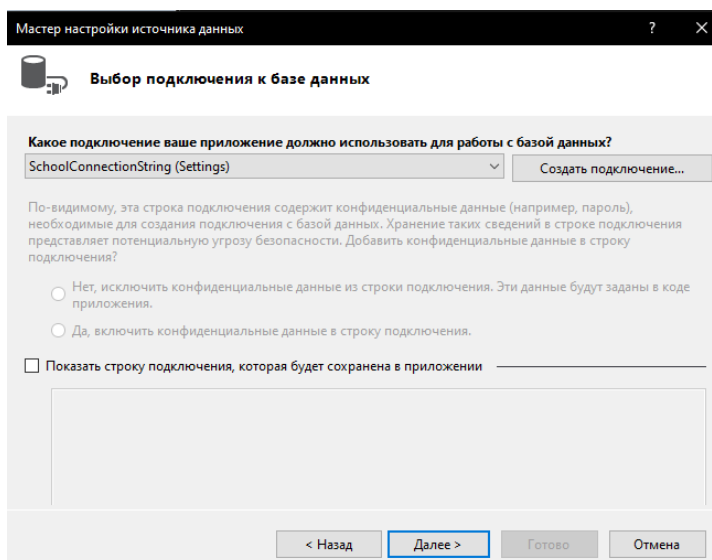


Рисунок 3.3 – Створення нового підключення

Після зроблених дій, можна створювати головне меню програми. Програма буде складатись з безлічі Windows Form (вікон програми). Для того щоб створити головне меню програми, потрібно додати на головну форму компонент - Menu Strip, та налаштувати його, відповідно до функцій програми (Рисунок 3.4)

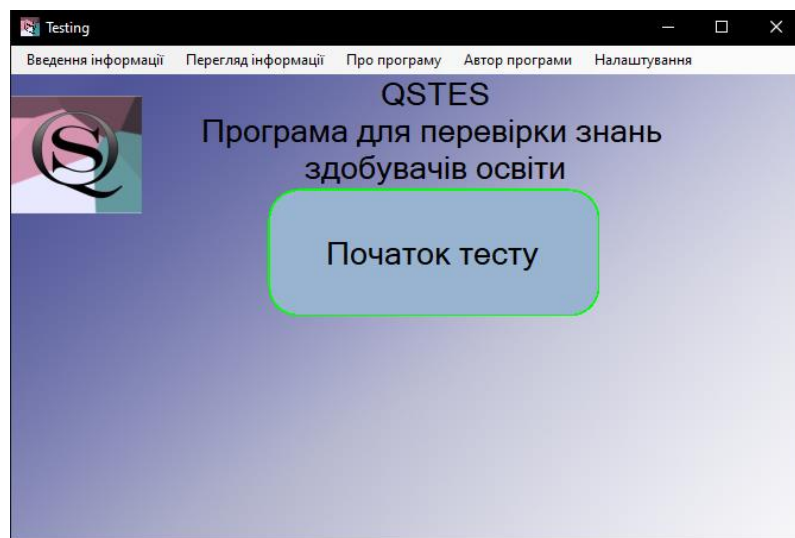


Рисунок 3.4 – Головне меню програми

Щоб додати на форму поля для введення даних, необхідно у меню «Дані» обрати створене раніше сховище даних — DataSet, знайти потрібну таблицю з бази даних, обрати формат її представлення та перенести на форму (див. Рисунок 3.5).

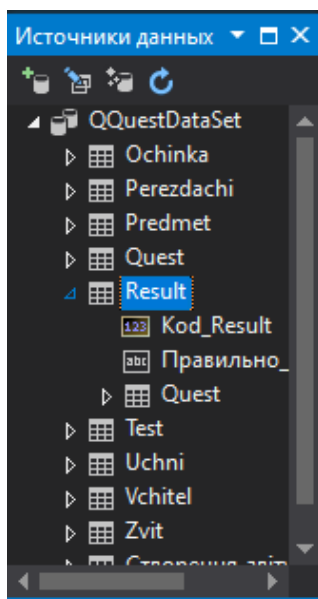
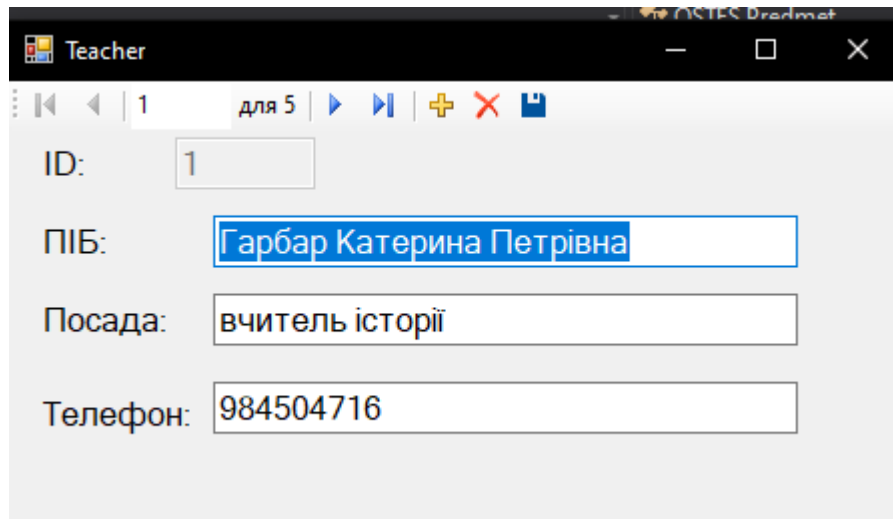


Рисунок 3.5 – Джерела даних

Після перенесення таблиці на форму, мають з'явитись поля для введення даних, також вище полів будуть автоматично створені кнопки навігації по записам: перехід між записами, видалення, додавання, збереження записів. Створена форма виглядає наступним чином (Рисунок 3.6).

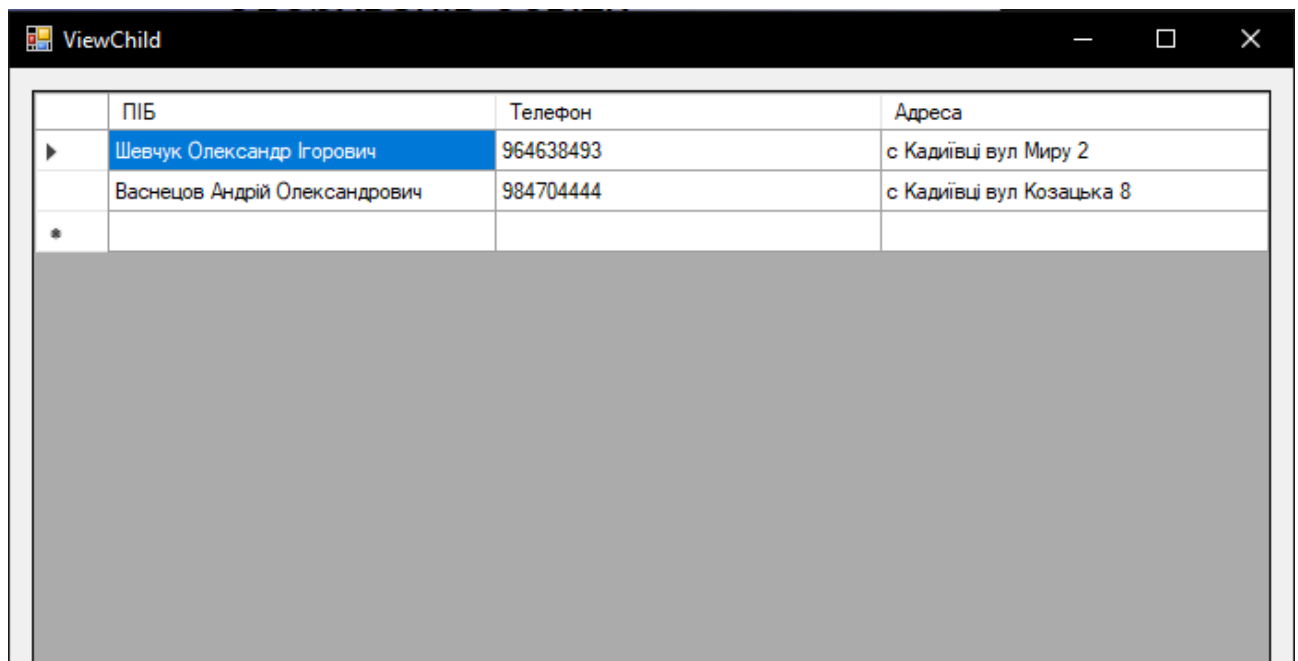


The screenshot shows a web browser window titled 'Teacher'. The address bar contains 'для 5'. Below the address bar are navigation icons: a double left arrow, a single left arrow, a vertical bar with '1', a double right arrow, a right arrow, a plus sign, a red X, and a save icon. The form contains four input fields:

- ID: 1
- ПІБ: Гарбар Катерина Петрівна
- Посада: вчитель історії
- Телефон: 984504716

Рисунок 3.6 – Форма для введення даних

Для зручності створимо вікна перегляду записаної раніше інформації, для цього використаємо елемент DataGridView. Створені вікна виглядають наступним чином (Рисунок 3.7).



The screenshot shows a web browser window titled 'ViewChild'. It displays a table with the following data:

	ПІБ	Телефон	Адреса
▶	Шевчук Олександр Ігорович	964638493	с Кадіївці вул Миру 2
	Васнецов Андрій Олександрович	984704444	с Кадіївці вул Козацька 8
*			

Рисунок 3.7 – Форма перегляду записаних даних

Для спрощення процесу пошуку та фільтрування інформації на форму додаються дві кнопки з назвами «Фільтрування» та «Пошук». Для їх функціонування розробляється наступний код.

Код для реалізації пошуку:

```
private void button1_Click(object sender, EventArgs e)
{
    int itemFound =
this.Назва_завданняBindingSource.Find("Назва_завдання", textBox1.Text);
this.Назва_завданняBindingSource.Position = itemFound;
```

Код роботи кнопки фільтрування:

```
private void button1_Click(object sender, EventArgs e)
{
    this.predmetBindingSource.Filter = "(Kod_Predmet=" +
comboBox1.SelectedValue.ToString() + ")";
}
private void button2_Click(object sender, EventArgs e)
{
    this.predmetBindingSource.RemoveFilter();
}
```

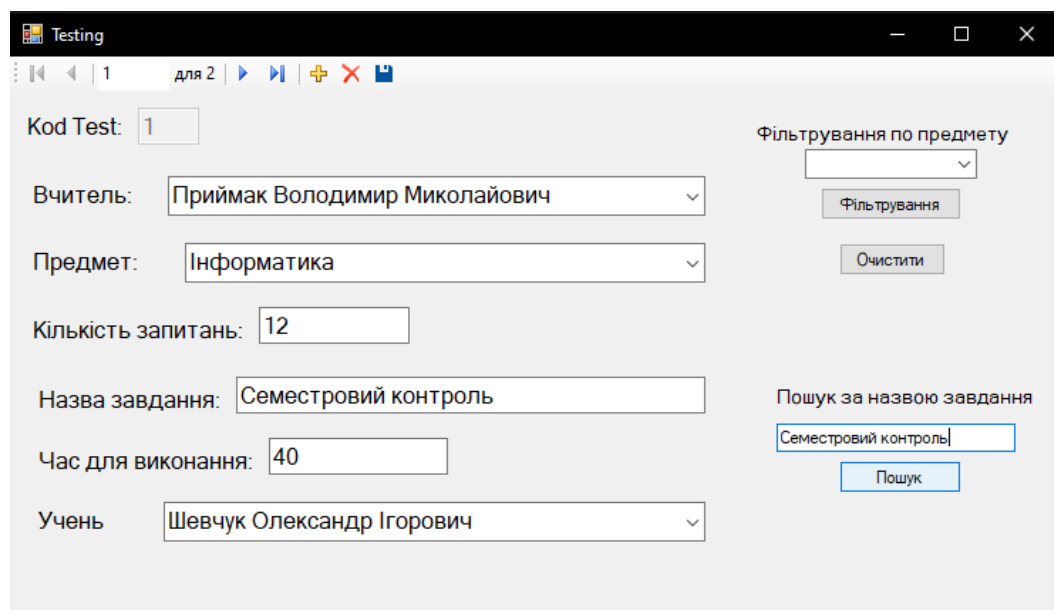


Рисунок 3.8 – Результат роботи пошуку

Наступним кроком налаштуємо правило валідації. Згідно проектової бази, маємо таке правило, що поле «ПШБ» не може бути пустим (Рисунок 3.12).

Щоб програма не зупиняла свою роботу при не дотриманні валідації було додано код:

```

private void errorHandler_DataError(object sender,
DataGridViewDataErrorEventArgs e)
{
    if (e.Exception is FormatException)
    {
        e.ThrowException = false;
    }
}

```

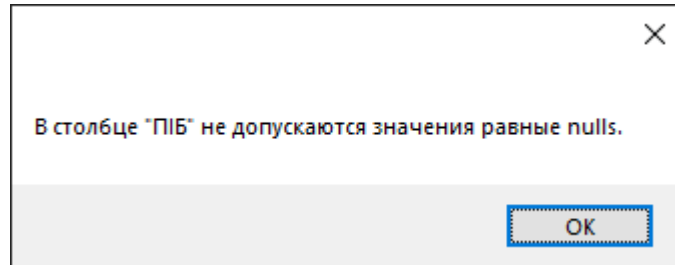


Рисунок 3.9 – Помилка при збереженні даних.

Подібні дії було пророблено для всіх важливих полів, де може статись помилка при введенні даних.

Далі була розроблена система тестування, задані назви змінних, розроблена форма для тестування, додані кнопки для відповіді, та наступного питання, кнопки для вибору правильних відповідей, та текстове поле «TextVox», де будуть відбуватись показ питань з текстового файлу (Рисунок 3.14).

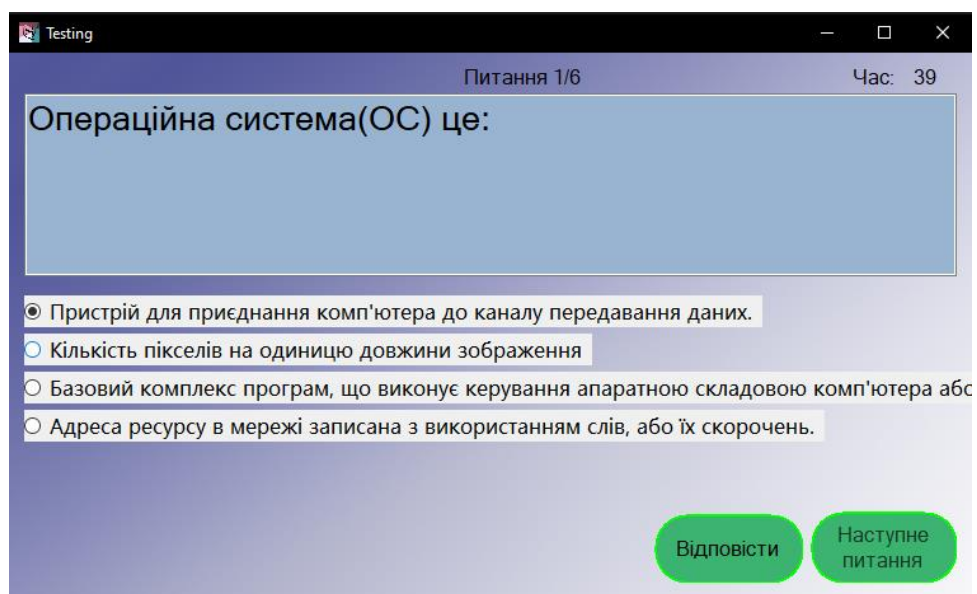


Рисунок 3.10 – Вікно для проходження тесту

Наступним етапом було створення форми для перегляду аналітичної інформації по результатах тестування здобувачів освіти. На формі було задано пошук за двома параметрами, це клас де навчається учень, та назва предмету за яким проходив тест. Код відповідної форми був адаптований під створювану раніше базу даних, з якої і буде братись інформація для перегляду аналітики, частина коду для цієї форми зображений нижче:

```

using System;

using System.Data;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

namespace AnalyticsApp

{
    public partial class MainForm : Form

        public MainForm()

            InitializeComponent();

            InitializeDataGrid();

            InitializeChart();

        private DataGridView dataGrid;

        private Chart analyticsChart;

        // Ініціалізація DataGridView

        dataGrid = new DataGridView

        {
            Dock = DockStyle.Top,

            Height = 200,

            AllowUserToAddRows = false,

            ReadOnly = true,

            AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill

```

```
this.Controls.Add(dataGrid);

analyticsChart = new Chart
    Dock = DockStyle.Fill

this.Controls.Add(analyticsChart);

ChartArea chartArea = new ChartArea("MainArea");

analyticsChart.ChartAreas.Add(chartArea);

Series series = new Series("Оцінки")
    ChartType = SeriesChartType.Column,
    XValueType = ChartValueType.String

analyticsChart.Series.Add(series);

private void LoadData()

    DataTable table = new DataTable();

    table.Columns.Add("Учень", typeof(string));

    table.Columns.Add("Оцінка", typeof(int));

    // Відображення у DataGridView

    dataGrid.DataSource = table;

    // Додавання даних до Chart

    foreach (DataRow row in table.Rows)

        analyticsChart.Series["Оцінки"].Points.AddXY(row["Учень"],
row["Оцінка"]);    } }
```

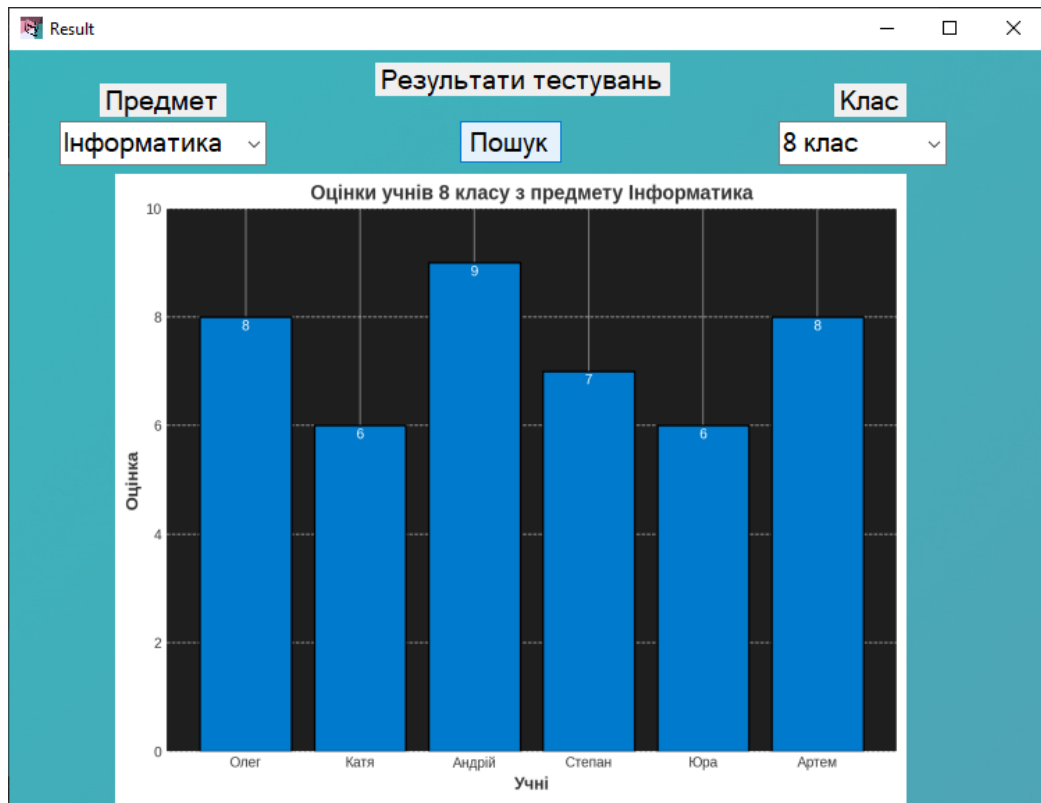


Рисунок 3.11 – Форма для аналітичної звітності

3.4 Інструкція користувача

Щоб працювати з додатком для тестування "QSTES", потрібно спочатку завантажити установочний файл під назвою "setup QSTES" і відкрити його. Після цього з'явиться вікно, де ви зможете розпочати встановлення програми. Просто дотримуйтесь інструкцій на екрані, і програма успішно встановиться та запуститься. На першому етапі ви побачите вікно, де потрібно буде вибрати мову встановлення (Рисунок 3.12).

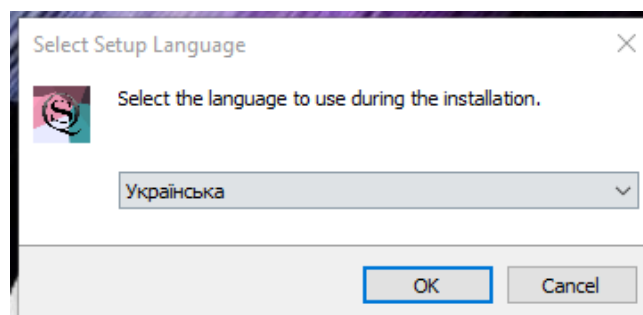


Рисунок 3.12 – Вікно вибору мови установки

Далі з'явиться вікно, в якому потрібно буде ознайомитися з ліцензійною угодою, підтвердити її прийняття, і тільки після цього можна буде перейти до наступного етапу встановлення програми. Якщо ви не погодитесь з умовами ліцензії, процес установки буде неможливим. На цьому етапі ліцензійна угода доступна лише англійською мовою. Її текст виглядає наступним чином:

Дякуємо за встановлення програми тестування - QSTES.

Програма передбачає автоматизоване тестування здобувачів освіти.

Копіювати, передавати або встановлювати окремі частини, або всю програму заборонено третім особам, без відома розробника (Рисунок 3.13).

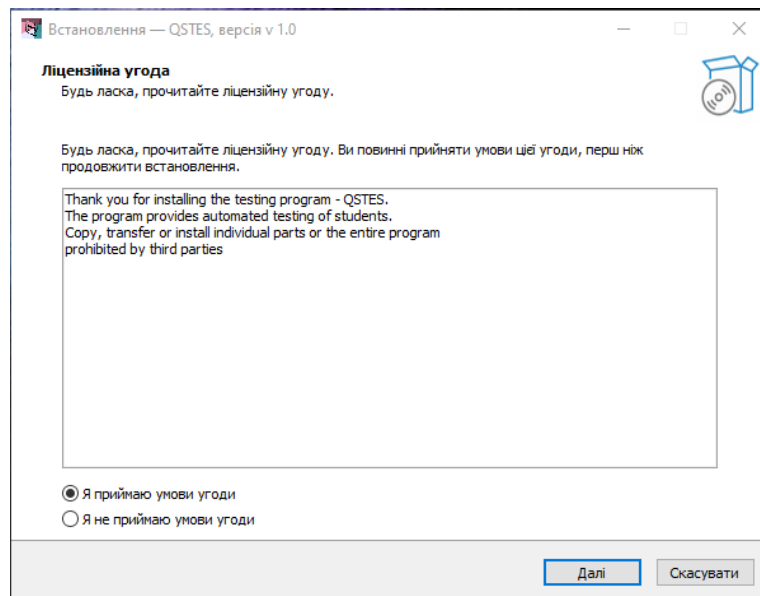


Рисунок 3.13 – Підтвердження ліцензійної угоди

Після того як ви приймете ліцензійну угоду, з'явиться вікно, де можна вибрати створення ярлика .exe файлу для запуску програми на робочому столі. Якщо ви відмовитеся від цієї опції, запуск програми буде можливий із папки, яку ви оберете під час установки (Рисунок 3.14).

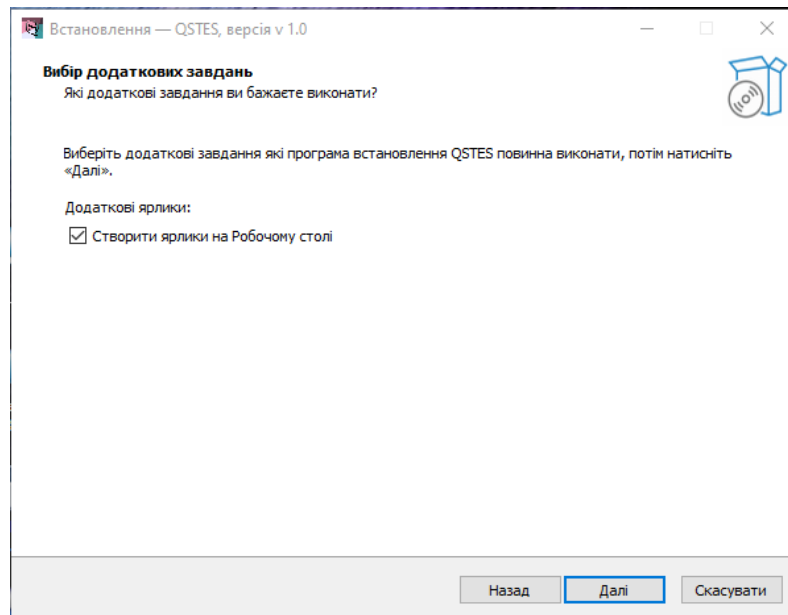


Рисунок 3.14 – Вибір створення ярлика на робочому столі

На завершальному етапі установки додатку вам буде запропоновано підтвердити установку або повернутися назад, щоб переглянути і змінити налаштування. У вікні з інформацією про встановлення буде відображено додаткові завдання, які ви вибрали на попередніх етапах (Рисунок 3.15).

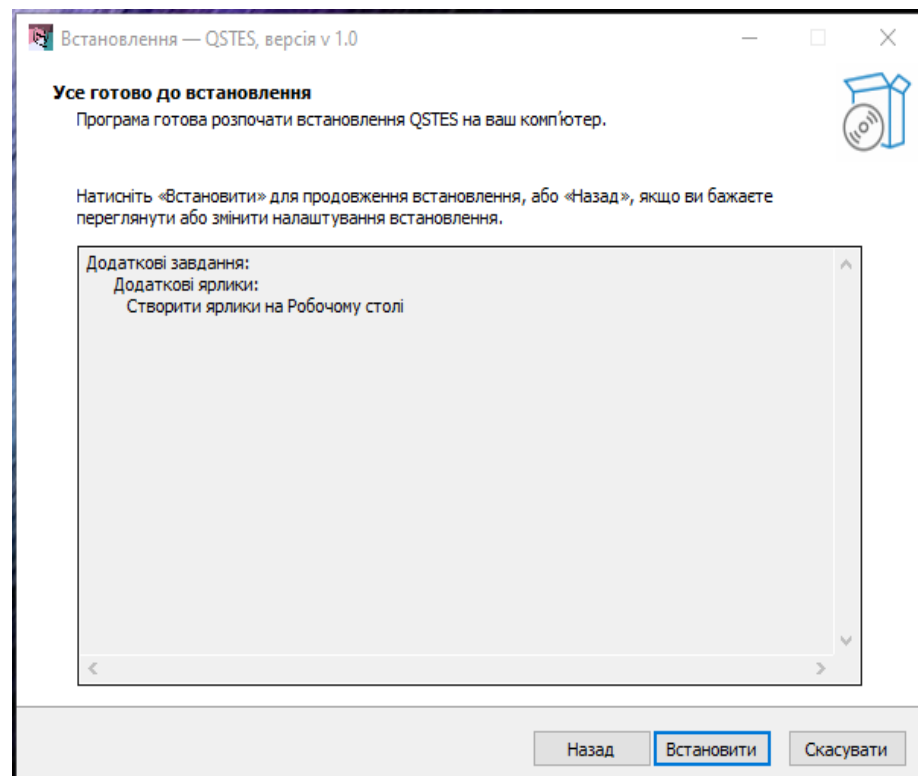


Рисунок 3.15 – Завершення установки

Після успішної установки пакетів програми на ваш комп'ютер, у вас відкриється вікно, з функцією, що дозволяє запустити програму зразу після завершення установки.

Після успішного встановлення додатку перед вами відкриється головне вікно програми та панель навігації зверху вікна. Як ознайомитесь з всіма вікнами програми можна буде безпосередньо переходити до реєстрації учня та подальшого тестування (Рисунок 3.15, 3.16).

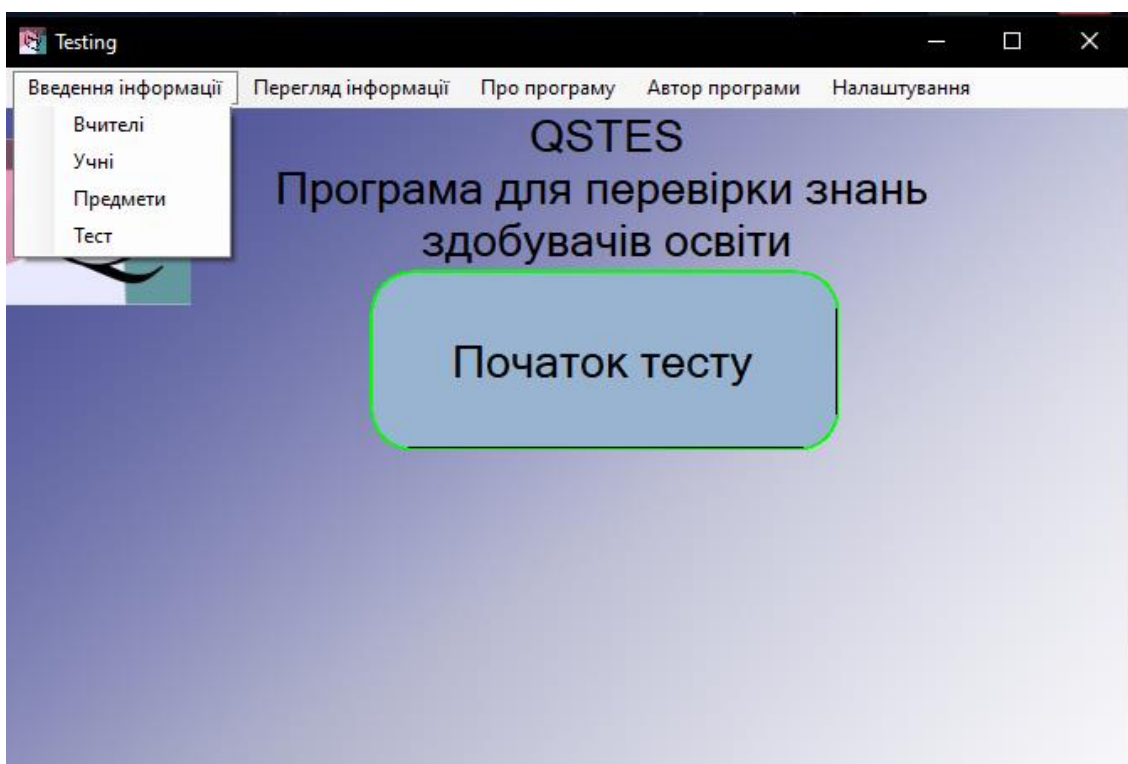


Рисунок 3.16 – Головне меню та панель навігації

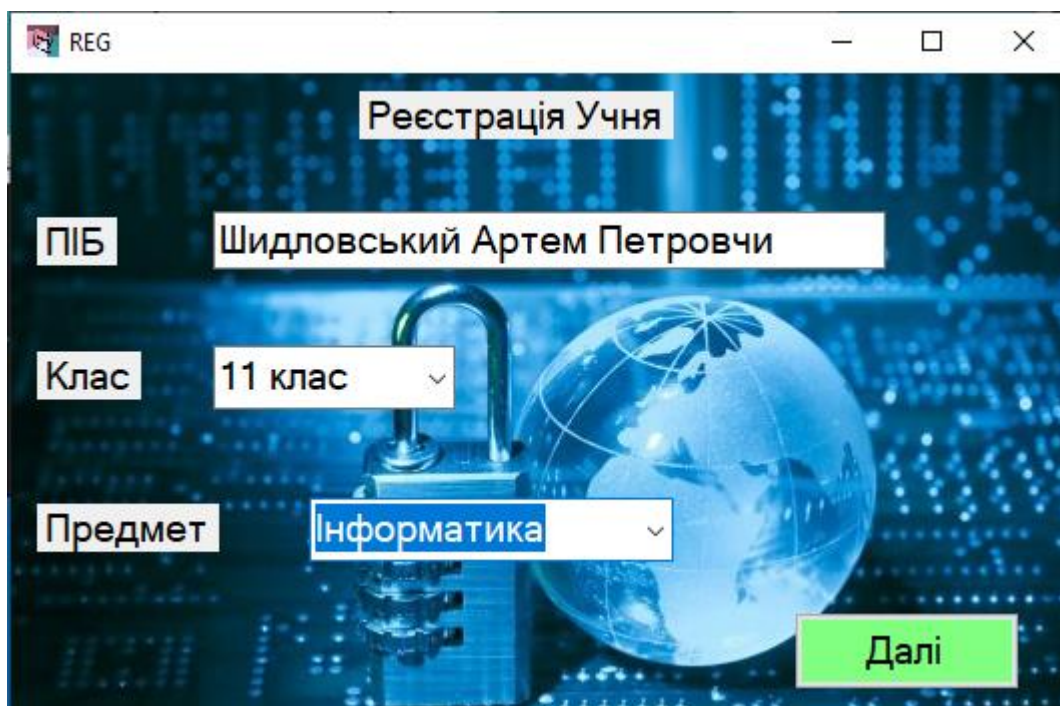


Рисунок 3.17 – Вікно реєстрації учня

Після натискання кнопки «Далі» на вікні реєстрації учня у вас відкриється вікно для початку тестування. Будьте уважні!, після натискання кнопки «ОК» автоматично запуститься таймер який буде рахувати час для кожного тесту та загалом тестування, тому розпочинайте тестування коли будете повністю до нього готові. Повідомлення про початок тестування та загальна інформація по тестування зображена на Рисунку 3.18.

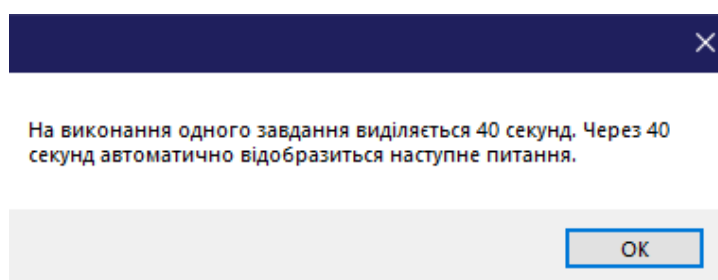


Рисунок 3.18 – Повідомлення про початок тестування

Після цього у вас відкриється вікно загалом для виконання тестів які були задані вчителем перед початком тестування. Також буде відображено у верхньому правому куті вікна таймер на виконання кожного завдання, організовано

позначення правильної та неправильної відповіді, також цю функцію є можливість вимкнути перед початком тесту в налаштуваннях на початковому вікні програмного додатку.

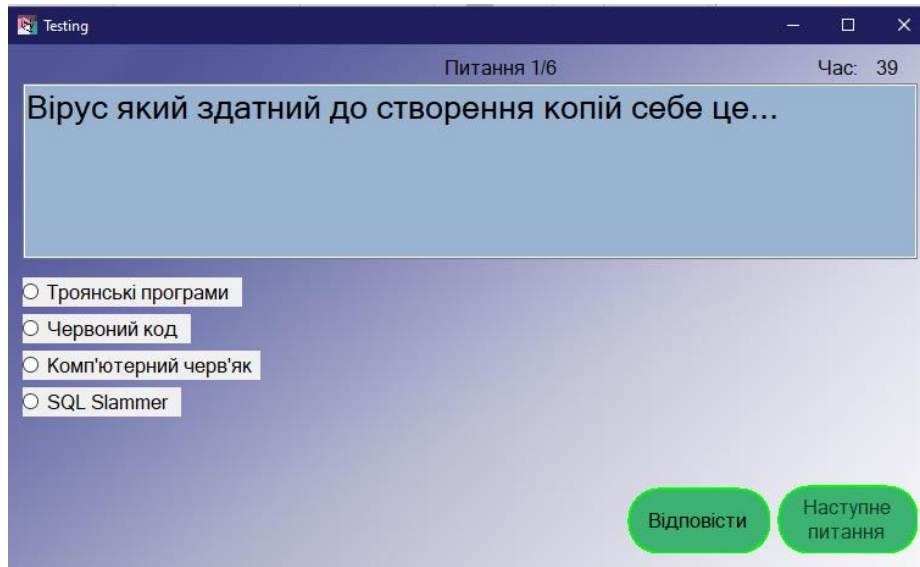


Рисунок 3.19 – Вікно для тестування

Після успішного складання тесту перед вами з'явиться вікно з інформацією про виконаний тест: Оцінка, кількість правильних відповідей, Скільки витрачено часу на тестування (Рисунок 3.20)

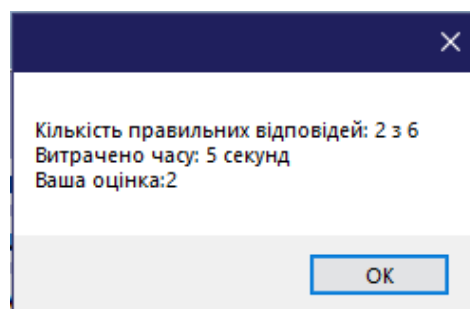


Рисунок 3.20 – Повідомлення після завершення тесту

Дані які були зафіксовані після завершення тестування автоматично зберігаються в базі даних програми для подальшого їх перегляду на вікні з аналітичними даними. На вікні з аналітичною інформацією зображено діаграму з результатами тестування кожного учня, визначено його оцінку, та додано час який

було затрачено на виконання тестування. Також додані фільтри на вікно для можливості пошуку інформації з бази даних за такими пунктами як: «Назва предмету» та «Клас». Після вибору відповідних параметрів пошуку та натискання на кнопку «Пошук» перед вами з'явиться діаграма з результатами (Рисунок 3.21).

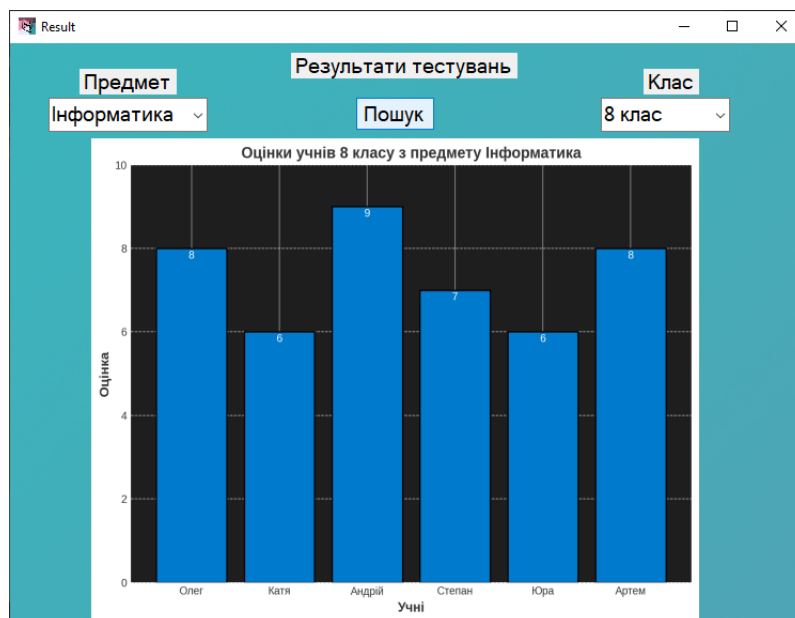


Рисунок 3.21 – Вікно для перегляду результатів тестування

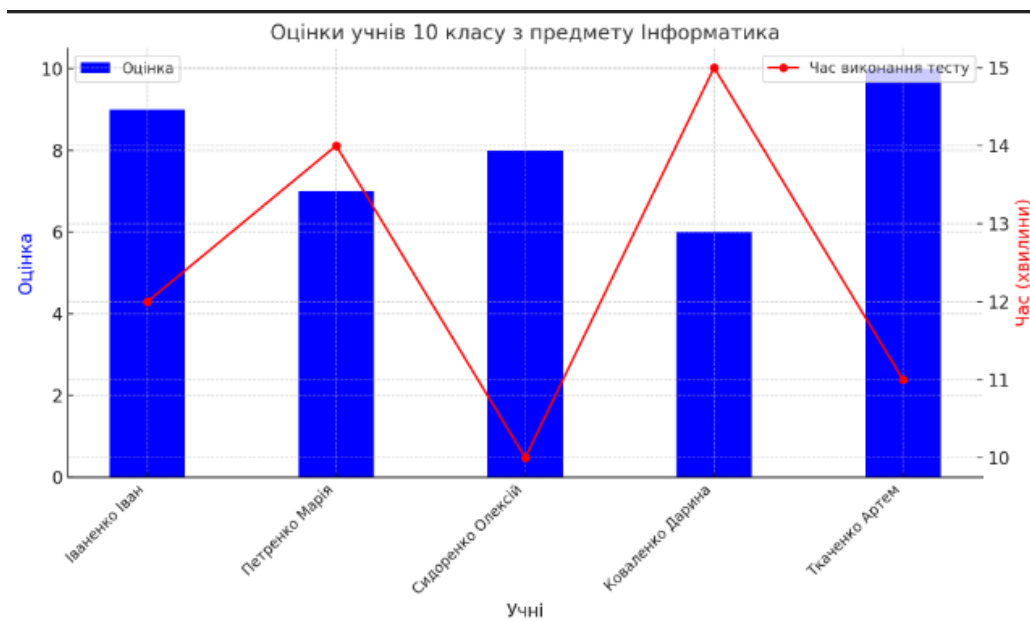


Рисунок 3.22 – Графік з успішністю учнів

Для того щоб почати нове тестування, просто закрийте вікно з результатами тестування та натисніть на кнопку «Розпочати тест». Вся інформація з тестувань зберігається в базі даних програми, тому після завершення програми буде можливість переглянути результати раніших тестувань.

3.5. Технічне та системне забезпечення розробки

Технічне та системне забезпечення додатку включає в себе використання різних технологій та інструментів для розробки та функціонування системи. Основні компоненти, що використовуються, включають: Мову програмування C#, NetFramework який вбудовано в Visual Studio, базу даних Microsoft SQL Server, різноманітні модулі та доповнення, основний – LiveCharts для створення діаграми на формі. Далі детальніше описано про кожен засіб розробки:

Мова програмування C# - була використана для цього проекту через легкість та адаптивність до візуальних елементів програми. Також можна відмітити її надзвичайну стабільність в роботі з багатьма модулями на різних формах програмного додатку. Також в неї вбудовано автоматичний розподіл пам'яті, що дозволяє без великих затрат ресурсів виконувати багато задач в реальному часі. Реалізовано також в ній кросплатформеність – можливість запускати додатки на різних системах Windows, MacOS, Linux та інші.

NET Framework — це потужна платформа для розробки, що використовується для створення додатків на базі Windows. Вона інтегрується в Visual Studio, забезпечуючи розробників інструментами для створення програмного забезпечення будь-якої складності, від простих консолей до складних веб- або десктопних додатків. Надає широкий набір класів для роботи з базами даних, обробки XML, роботи з потоками, створення графічних інтерфейсів та іншого. ADO.NET і Entity Framework - Інструменти для роботи з базами даних, які дозволяють розробляти програми з доступом до даних різного формату.

Microsoft SQL Server – один з найкращих СУБД для створення та управління базами даних. Для моєї роботи був вибраний саме SQL-Server 2022, тому що в ньому найкраща взаємодія з Visual Studio, гарна стабільність, та надзвичайно

швидко підключення до форм проекту, та зберігання даних. Він підтримує реляційну модель даних, де інформація організована в таблицях, що дозволяє створювати зв'язки між ними та виконувати складні запити для обробки даних. Однією з ключових особливостей SQL Server є підтримка транзакцій, що гарантує атомарність, консистентність, ізолюваність і довговічність операцій, що робить його ідеальним для бізнес-додатків, які потребують високої надійності та коректності даних. Крім того, SQL Server має розширені механізми безпеки, включаючи автентифікацію та авторизацію користувачів, шифрування даних, а також функції моніторингу та аудиту для захисту інформації.

LiveCharts — це бібліотека для побудови інтерактивних графіків і діаграм у .NET-додатках, яка забезпечує зручний і сучасний спосіб візуалізації даних. Вона має простий у використанні API, що дозволяє швидко інтегрувати графіки навіть новачкам. Бібліотека підтримує різні типи графіків, такі як лінійні графіки, гістограми, кругові діаграми, бульбашкові графіки та інші, що дає можливість створювати різноманітні візуалізації. Однією з ключових особливостей є підтримка анімацій, що робить графіки динамічними та привабливими для користувачів. LiveCharts також забезпечує зручну кастомізацію, дозволяючи змінювати вигляд графіків, кольори, стилі шрифтів та форматування осей, що дає можливість адаптувати графіки під дизайн програми.

3.6. Обґрунтування вибору технічних засобів

Visual Studio 2022 – середовище розробки програмних додатків. Продукт дає змогу розробляти консольні програми, програми за технологією Windows Form, вебсайти, різноманітні вебзастосунки. Це середовище програмування для моєї роботи підходить ідеально, тому що в ньому є багато засобів керування та розробки графічних елементів, та вбудована мова програмування ООП – C#. Також є можливість під'єднувати інші різноманітні модулі, наприклад для створення аналітичних звітностей такий як – LiveCharts. Однією з основних переваг Visual Studio 2022 є підтримка 64-розрядної архітектури, що дозволяє програмі використовувати більше пам'яті, покращуючи продуктивність при роботі з

великими проектами. Це особливо важливо для великих кодових баз, компіляцій або при роботі з ресурсомісткими додатками. Інтерфейс Visual Studio 2022 також був значно вдосконалений — він став більш швидким, зручним і зберігає простоту користування.

Adobe Photoshop – Надзвичайно потужний інструмент для створення графічних моделей, логотипів, фону програми, тощо. Ввесь фон програми, логотип програми, різні графічні моделі які були використані на проектуванні додатку були створені в цьому додатку. Однією з головних переваг Photoshop є його потужний набір інструментів для ретушування зображень. Це включає можливість відновлення деталей, видалення дефектів, корекцію освітлення та кольору, а також видалення об'єктів або фонових елементів. Photoshop також дозволяє працювати з масками, що дає можливість створювати складні ефекти та коригувати окремі частини зображення, не впливаючи на інші області.

Inno Setup - Цей інструмент дозволяє створювати як прості інсталятори для одиночних файлів, так і більш складні, з налаштуванням шляхів установки, створенням ярликів, реєстрацією файлів у системі та іншими етапами. Inno Setup використовує скрипти, написані на мові, схожій на Pascal, що дає можливість налаштування кожного етапу інсталяції та забезпечує гнучкість у процесі створення інсталяторів. Крім того, він підтримує багатоетапні установки, можливість вибору компонентів, налаштування параметрів та створення ярликів, а також має функції перевірки системних вимог, таких як наявність необхідних бібліотек або програмного забезпечення. Inno Setup дозволяє застосовувати шифрування та стиснення файлів, що робить інсталяційні пакети захищеними від несанкціонованого доступу та зменшує їх розмір.

3.7 Обґрунтування вибору ОС та протоколу обміну даними

Для цього проекту була обрана операційна система Windows 10. Вона є останньою версією Windows, яка підтримує 32-розрядні процесори (на базі IA-32 та ARMv7) та пристрої з BIOS (хоча оновлення від травня 2020 року припинило підтримку 32-розрядних процесорів). Для нової версії Windows 11 потрібен

пристрій з прошивкою UEFI і 64-розрядним процесором будь-якої підтримуваної архітектури (x86-64 для x86 та ARMv8 для ARM). Windows 10 була обрана для розробки через велику кількість доступних програм, протоколів і служб, що забезпечують її стабільну роботу. Вона також має високу продуктивність, що дозволило без затримок компілювати код і працювати з додатком. Крім того, Windows 10 має режим сумісності з іншими операційними системами, такими як Windows 8.1, Windows 7, Windows Vista, Windows XP, що дозволило протестувати програму на цих ОС.

3.8. Топологія комп'ютерної мережі

Програмний додаток розроблений на мультирівневій архітектурі (multi-tier architecture). Ця топологія мережі ділиться на 3 основних типи, це: Презентаційний рівень, Рівень бізнес-логіки, Рівень даних.

Презентаційний рівень (Presentation Layer): Це рівень, який відповідає за взаємодію з користувачем. Він зазвичай включає інтерфейс користувача (UI) та механізми для введення даних. На цьому рівні користувачі взаємодіють з додатком, отримують від нього інформацію або вводять дані, які будуть оброблені іншими рівнями. Це може бути веб-браузер для веб-додатків або мобільний додаток для мобільних пристроїв.

Рівень бізнес-логіки (Business Logic Layer): Цей рівень відповідає за обробку даних і виконання основних бізнес-операцій. Він містить правила та алгоритми, що визначають, як дані повинні оброблятися, перевірятися, змінюватися або взаємодіяти з іншими системами. Бізнес-логіка ізолює деталі реалізації та забезпечує, щоб усі операції виконувалися згідно з певними бізнес-вимогами.

Рівень даних (Data Layer): Це рівень, який відповідає за зберігання та доступ до даних. Це можуть бути бази даних, файлові системи або інші типи сховищ інформації. Рівень даних забезпечує збереження інформації та її обробку через операції, такі як запис, читання, оновлення або видалення даних.

Було обрано саме таку топологію через її простоту та адаптацію з розроблювальним додатком, тому що головний інтерфейс програми відповідає за

організацію роботи з основним користувачем, код та логіка програми виступає як рівень бізнес-логіки, який обробляє, фіксує та зчитує інформацію яка обробляється або завантажується в додаток. І останній рівень це рівень даних в програмному додатку виступає як база даних SQL, яка зберігає в собі дані роботи програми, результати тестувань, інформацію про учнів, викладачів, тощо.

3.8 Висновок до третього розділу

На основі дослідницьких даних які було проведені у Розділі 1 та Розділі 2 кваліфікаційної роботи, було розроблено систему для тестування та аналізу результатів тестування здобувачів освіти. У третьому розділі детально писано проектування системи, опис використаних програмних засобів та систем, їх аналіз та обґрунтування вибору. Також було описано топологію програмного засобу та введення його в експлуатацію.

ВИСНОВОК

У результаті виконання магістерської роботи було проведено дослідження підходів та методів аналізу результатів тестування здобувачів освіти та розроблено концепцію аналітичної системи підтримки прийняття рішень для оцінки ефективності навчального процесу. Метою дослідження було вивчення існуючих методів збору, обробки та аналізу даних про результати тестувань, а також створення системи, яка дозволить автоматизувати цей процес і забезпечити сучасні інструменти для прийняття обґрунтованих рішень у сфері освіти.

У ході дослідження було виявлено, що освітні заклади стикаються з низкою проблем під час використання традиційних методів роботи з даними. Зокрема, ручний збір інформації збільшує кількість помилок і витрати часу, а відсутність актуальних даних у реальному часі значно ускладнює оперативний аналіз. Крім того, традиційні методи не забезпечують можливості інтеграції з іншими освітніми платформами та системами управління навчальним процесом, що обмежує використання комплексного підходу до аналізу та оцінювання успішності учнів. Ці недоліки знижують ефективність процесу прийняття рішень і перешкоджають персоналізації навчання.

У результаті розробки аналітичної системи було запропоновано створити систему тестування, яка забезпечить автоматизований збір і обробку даних про результати тестувань, надасть можливість інтеграції з іншими платформами та забезпечить зручний доступ до актуальної інформації через інтуїтивно зрозумілий інтерфейс. Ця система дозволить керівництву освітніх закладів оперативно отримувати звіти та аналітичні дані, сприятиме вдосконаленню навчального процесу, допоможе виявляти проблемні аспекти та оцінювати прогрес учнів.

Використання сучасних методів аналізу даних, таких як візуалізація, прогнозування результатів та аналітика на основі великих даних, дасть змогу суттєво покращити процес прийняття рішень у сфері освіти. Це, своєю чергою,

сприятиме підвищенню якості навчання, більш точному оцінюванню успішності учнів і підвищенню конкурентоспроможності освітнього закладу. Проведене дослідження та розробка концепції аналітичної системи для аналізу результатів тестування є важливим кроком для покращення освітніх процесів. Впровадження такої системи дозволить автоматизувати обробку даних, оптимізувати оцінювання та забезпечити ефективне управління навчальним процесом у динамічно змінюваних умовах.

Наступним кроком у розвитку програмного додатку для тестування здобувачів освіти має бути створення нових модулів генерації питань, а також розширення функціональності вибору питань для тесту, та додавання різноманітних функцій для тесту. Також впродовж експлуатації додатку потрібно буде розробляти інші форми для зручного перегляду аналітичної інформації по результатах тестування.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Томашевський В.М. Інформатика. Моделювання систем ВМВ, 2005. 14с.
2. Філіп Джепікс, Ендрю Троелсен. Мова програмування С# 7.0 та платформи .NET та .NET Core: Вільямс, 2018. 1330с.
3. ChatGPT. URL: <https://chat.openai.com> (Дата звернення 18.10.2024).
4. Microsoft SQL-Server. URL: https://uk.wikipedia.org/wiki/Microsoft_SQL_Server (Дата звернення 10.10.2024).
5. Державні санітарні правила та норми. URL: <https://zakon.rada.gov.ua/rada/show/v0009588-98> (Дата звернення 12.10.2024).
6. ДСТУ ISO/IEC 12207:2014. Інженерія систем і програмного забезпечення. Процеси життєвого циклу програмного забезпечення.
7. Angular Material [Електронний ресурс] – Режим доступу до ресурсу: <https://material.angular.io> (Дата звернення 23.11.2024).
8. ДСТУ ISO/IEC 29155-1:2015. Розроблення систем і програмного забезпечення. Платформи для тестування проєктів з розроблення інформаційних систем. Частина 1. Концепції та визначення.
9. ДСТУ 3008:2015 — Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. – К.: ДП «УкрНДНЦ», 2015. – 32 с.
10. Капітальні вкладення. URL: <https://uk.wikipedia.org/wiki/Капітал> (Дата звернення 21.11.2024).
11. Куточок з охорони праці в закладі освіти. URL: <https://www.pedrada.com.ua/article/2332-kutochok-ohoroni-prats-v-zaklad-osvti> (Дата звернення 19.11.2024).
12. Методичні рекомендації до виконання КР "Комп'ютерні науки". URL: <https://cde.nuft.edu.ua/mod/resource/view.php?id=619410> (Дата звернення 10.10.2024).

13. Норми освітленості комп'ютерних класів. URL: <https://city-adm.lviv.ua/news/science-and-health/medicine/219680-sanitarno-hihienichni-vymohy-roboty-na-komp-iuteri-v-navchalnykh-zakladakh> (Дата звернення 21.11.2024).
14. Переменные-ссылки и возвращение ссылки. URL: <https://metanit.com/sharp/tutorial/2.21.php> (Дата звернення 10.10.2024).
15. Плани розміщення комп'ютерних класів. URL: <https://vseosvita.ua/library/osnovni-psihologo-pedagogicni-ta-sanitarno-igienicni-vimogi-do-provedenna-zanat-informatiki-v-pocatkovij-skoli-123683.html> (Дата звернення 17.11.2024).
16. Платформи для створення тестів. URL: <https://osvita.ua/school/method/technol/45747/> (Дата звернення 19.10.2024).
17. Пожежна безпека у закладах освіти. Рекомендації експерта. URL: <https://auc.org.ua/novyna/pozhezhna-bezpeka-u-zakladah-osvity-rekomendaciyi-eksperta> (Дата звернення 05.11.2024).
18. Програмні засоби для проведення тестового контролю знань. URL: <http://oldconf.neasmo.org.ua/node/1821> (Дата звернення 08.11.2024).
19. Розробка програм за допомогою Windows Form. URL: <https://www.bestprog.net/uk/2021/04/09/c-windows-forms-development-of-a-program-for-demonstrating-work-with-threads-of-execution-ua/> (Дата звернення 09.11.2024).
20. Справочник по C#. Оператор объединения с NULL. URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/operators/null-coalescing-operator> (Дата звернення 12.11.2024).
21. Створення бази даних SQL, покрокова інструкція. URL: <https://uk.myservername.com/mysql-create-database-how-create-database-mysql> (Дата звернення 08.11.2024).
22. Створення новго підключення бази даних в Visual Studio. URL: <https://learn.microsoft.com/ru-ru/visualstudio/data-tools/add-new-connections?view=vs-2022> (Дата звернення 06.10.2024).

23. Створення стратегії середовища. URL: <https://learn.microsoft.com/uk-ua/powerplatform/guidance/adoption/environment-strategy> (Дата звернення 16.10.2024).

24. Створення логотипів в Photoshop. URL: <https://www.logaster.com.ua/blog/create-logo-photoshop/> (дата звернення 15.10.2024)

25. Створення тестових завдань з допомогою програми MyTestX. URL: <https://dowjok2.e-schools.info/pages/stvorennja-testiv-z-dopomogoju-programy-mytestx>. (Дата звернення 09.11.2024).

26. 7 сервісів для створення освітніх тестів і завдань. URL: <https://buki.com.ua/ru/news/7-servisiv-dlya-stvorennya-navchalnykh-testiv-ta-zavdan-onlayn/> (Дата звернення 12.11.2024).

ДОДАТКИ**Додаток А**

Код програми:

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

namespace QSTES
{
    public struct Quests
    {
        public string QuestText;
        public string ans1;
        public string ans2;
        public string ans3;
        public string ans4;
        public int curRightAns;
    };
    public partial class Questions : Form
    {
        private int timeOfTest;
        private Random rand = new Random();
        private bool[] ComplQuest;
```

```

private int currentAnswer, numbOfQuest = 0, qst, mark = 0;
private int timeLeft=40,i;
private int result =0;
private Form1 _form1;
private Quests[] Squests = new Quests[30];
private StreamReader strRead;
public Questions(Form1 form1) //завантаження форми та її
елементів
{
    _form1 = form1;
    InitializeComponent();
    roundedButton2.Enabled = false;
    radioButton1.Visible = false;
    radioButton2.Visible = false;
    radioButton3.Visible = false;
    radioButton4.Visible = false;
    textBox1.Visible = false;
    roundedButton1.Enabled = false;
    label2.Text = "";
    label1.Text = "";
}
private void CheckRightAnswer(int RightAns, int curAns)
//перевірка чи правильно дали відповідь
{
    if (curAns == RightAns)
    {
        result++;
    }
    qst++; // кількість показаних питань
}

```

```

private void timer1_Tick(object sender, EventArgs e) //рахує час на
даному питанні
{
    if (timeLeft > 0)
    {
        timeLeft -= 1;
        label1.Text = timeLeft.ToString(); //виводить кількість часу що
залишився на дане питання
    }
    else
    {
        roundedButton2.PerformClick();
        Quest();
    }
}
private void button1_Click(object sender, EventArgs e) //кнопка
"наступне питання"
{
}
private void button3_Click(object sender, EventArgs e) //вибір
важкого тесту
{
    StartTest("QuestHard.txt");
}
private void button4_Click(object sender, EventArgs e) //вибір
легкого тесту
{
    StartTest("QuestEasy.txt");
}

```

```

private void button2_Click(object sender, EventArgs e) //вибір тесту
{
    StartTest("QuestMedium.txt");
    label3.Visible = true;
    roundedButton1.Visible = true;
    roundedButton2.Visible = true;
}

private void timer2_Tick(object sender, EventArgs e) //рахеє час
затрачений на весь тест
{
    timeOfTest++;
}

private void Questions_Load(object sender, EventArgs e)
{
    this.Visible = false;
    _form1.Visible = true;
}

private void roundedButton1_Click(object sender, EventArgs e)
{
    try
    {
        //визначаєм вибрану відповідь
        if (radioButton1.Checked)
        {
            currentAnswer = 1;
        }
        else if (radioButton2.Checked)
        {
            currentAnswer = 2;
        }
    }
}

```

```
}  
else if (radioButton3.Checked)  
{  
    currentAnswer = 3;  
}  
else if (radioButton4.Checked)  
{  
    currentAnswer = 4;  
}  
  
//деактивуємо радіобатони  
radioButton1.Enabled = false;  
radioButton2.Enabled = false;  
radioButton3.Enabled = false;  
radioButton4.Enabled = false;  
  
//перевіряємо чи правильна відповідь  
CheckRightAnswer(Squests[numbOfQuest].curRightAns,  
currentAnswer);  
  
//закрашуємо нашу відповідь червоним  
if (radioButton1.Checked)  
{  
    radioButton1.BackColor = Color.Red;  
}  
else if (radioButton2.Checked)  
{  
    radioButton2.BackColor = Color.Red;  
}  
else if (radioButton3.Checked)  
{  
    radioButton3.BackColor = Color.Red;  
}
```

```
else if (radioButton4.Checked)
{
    radioButton4.BackColor = Color.Red;
}
//закрашуєм правильну відповідь зеленим
switch (Squests[numbOfQuest].curRightAns)
{
    case 1: radioButton1.BackColor = Color.Green; break;
    case 2: radioButton2.BackColor = Color.Green; break;
    case 3: radioButton3.BackColor = Color.Green; break;
    case 4: radioButton4.BackColor = Color.Green; break;
}
//button1.Visible = true;
roundedButton2.Enabled = true;
roundedButton1.Enabled = false;
}
catch (Exception ex)
{
    MessageBox.Show(ex.ToString());
}
}
private void roundedButton2_Click(object sender, EventArgs e)
{
    roundedButton2.Enabled = false;
    roundedButton1.Enabled = true;
    Quest();
}
private void label2_Click(object sender, EventArgs e)
{
}
```

```

private void button5_Click(object sender, EventArgs e) //кнопка
відповісти
{
}
private void StartTest(string fName)
{
    try
    {
        qst = 0;
        timeOfTest = 0;
        numbOfQuest = 0;
        result = 0;
        i = 0;
        MessageBox.Show("На виконання одного завдання
віділяється "+timeLeft+" секунд. Через "+timeLeft+" секунд автоматично
відобразиться наступне питання.");
        roundedButton2.Enabled = false;
        roundedButton1.Enabled = true;
        strRead = new StreamReader(fName,
System.Text.Encoding.GetEncoding(1251));
        while (!strRead.EndOfStream) //читаєм питання з файла і
записуєм їх в структуру
        {
            Squests[i].QuestText = strRead.ReadLine();
            Squests[i].ans1 = strRead.ReadLine();
            Squests[i].ans2 = strRead.ReadLine();
            Squests[i].ans3 = strRead.ReadLine();
            Squests[i].ans4 = strRead.ReadLine();
            //визначаєм правильну відповідь до поточного питання

```

```
if (Squests[i].ans1.StartsWith('!.ToString()))
{
    Squests[i].curRightAns = 1;
    Squests[i].ans1 = Squests[i].ans1.Remove(0, 1);
}
else if (Squests[i].ans2.StartsWith('!.ToString()))
{
    Squests[i].curRightAns = 2;
    Squests[i].ans2 = Squests[i].ans2.Remove(0, 1);
}
else if (Squests[i].ans3.StartsWith('!.ToString()))
{
    Squests[i].curRightAns = 3;
    Squests[i].ans3 = Squests[i].ans3.Remove(0, 1);
}
else if (Squests[i].ans4.StartsWith('!.ToString()))
{
    Squests[i].curRightAns = 4;
    Squests[i].ans4 = Squests[i].ans4.Remove(0, 1);
}
i++;
}
ComplQuest = new bool[i];
radioButton1.Visible = true;
radioButton2.Visible = true;
radioButton3.Visible = true;
radioButton4.Visible = true;
textBox1.Visible = true;
button2.Visible = false;
button3.Visible = false;
```

```

        button4.Visible = false;
        timer1.Start();
        timer2.Start();
        Quest();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
}
private void Quest() // покажує наступне питання або закінчує тест
{
    try
    {
        //активуєм радіобатони і скидаємо колір
        radioButton1.Enabled = true;
        radioButton2.Enabled = true;
        radioButton3.Enabled = true;
        radioButton4.Enabled = true;
        radioButton1.BackColor = SystemColors.Control;
        radioButton2.BackColor = SystemColors.Control;
        radioButton3.BackColor = SystemColors.Control;
        radioButton4.BackColor = SystemColors.Control;
        timeLeft = 40; //секунд на поточне питання

        if (qst == ComplQuest.Length) //кількість показаних питань =
кількості всіх питань
        {
            timer1.Stop();
            timer2.Stop();

```

```

label2.Text = "";
label1.Text = "";
label3.Text = "";
mark = CheckResult(result);//рахуєм оцінку
MessageBox.Show("Кількість правильних відповідей: " +
result + " з "+i+ "\nВитрачено часу: " + timeOfTest + " секунд\n" + "Ваша
оцінка:" + mark);

//ховаєм елементи форми
radioButton1.Visible = false;
radioButton2.Visible = false;
radioButton3.Visible = false;
radioButton4.Visible = false;
textBox1.Visible = false;
roundedButton2.Enabled = false;
roundedButton1.Enabled = false;
button2.Visible = true;
button3.Visible = true;
button4.Visible = true;
strRead.Close();
this.Hide();
_form1.Show();
}
else
{
label2.Text = "Питання " + (qst + 1) + "/" +
ComplQuest.Length; //показує кількість питань
bool ОК = false;
while (!ОК) //цикл що дає змогу показувати кожен раз
питання в різному порядку
{

```

```

        numOfQuest = rand.Next(0, ComplQuest.Length);
        if (!ComplQuest[numOfQuest]) //якщо false
        {
            //показуєм питання
            string s =
Squests[numOfQuest].curRightAns.ToString();
            textBox1.Text = Squests[numOfQuest].QuestText;
            radioButton1.Text = Squests[numOfQuest].ans1;
            radioButton2.Text = Squests[numOfQuest].ans2;
            radioButton3.Text = Squests[numOfQuest].ans3;
            radioButton4.Text = Squests[numOfQuest].ans4;
            ComplQuest[numOfQuest] = true;
            OK = true;
            continue;
        }
        else
        {
            numOfQuest = rand.Next(0, ComplQuest.Length);
            OK = false;
        }
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.ToString());
}
}

```

```
private int CheckResult(int result) //повертає оцінку в залежності  
від кількості правильних відповідей
```

```
    if (result >= 10) {  
        return 10; }  
    if (result >= 9) {  
        return 9; }  
    if (result >= 8) {  
        return 8; }  
    if (result >= 7) {  
        return 7; }  
    if (result >= 6){  
        return 6; }  
    else if (result >= 5) {  
        return 5; }  
    else if (result >= 4) {  
        return 4;  
    } else if (result >= 3) {  
        return 3;  
    }else {  
        return 2; }  
using System;  
using System.Data;  
using System.Windows.Forms;  
using System.Windows.Forms.DataVisualization.Charting;  
namespace AnalyticsApp  
{ public partial class MainForm : Form  
    public MainForm()  
        InitializeComponent();  
        InitializeDataGrid();  
        InitializeChart();
```

```

private DataGridView dataGrid;
private Chart analyticsChart;
// Ініціалізація DataGridView
dataGrid = new DataGridView
{
    Dock = DockStyle.Top,
    Height = 200,
    AllowUserToAddRows = false,
    ReadOnly = true,
    AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill
};
this.Controls.Add(dataGrid);
analyticsChart = new Chart
{
    Dock = DockStyle.Fill
};
this.Controls.Add(analyticsChart);
ChartArea chartArea = new ChartArea("MainArea");
analyticsChart.ChartAreas.Add(chartArea);
Series series = new Series("Оцінки")
{
    ChartType = SeriesChartType.Column,
    XValueType = ChartValueType.String
};
analyticsChart.Series.Add(series);
private void LoadData()
{
    DataTable table = new DataTable();
    table.Columns.Add("Учень", typeof(string));
    table.Columns.Add("Оцінка", typeof(int));
    // Відображення у DataGridView
    dataGrid.DataSource = table;
    // Додавання даних до Chart
    foreach (DataRow row in table.Rows)
        analyticsChart.Series["Оцінки"].Points.AddXY(row["Учень"],
row["Оцінка"]); } }

```

База даних SQL

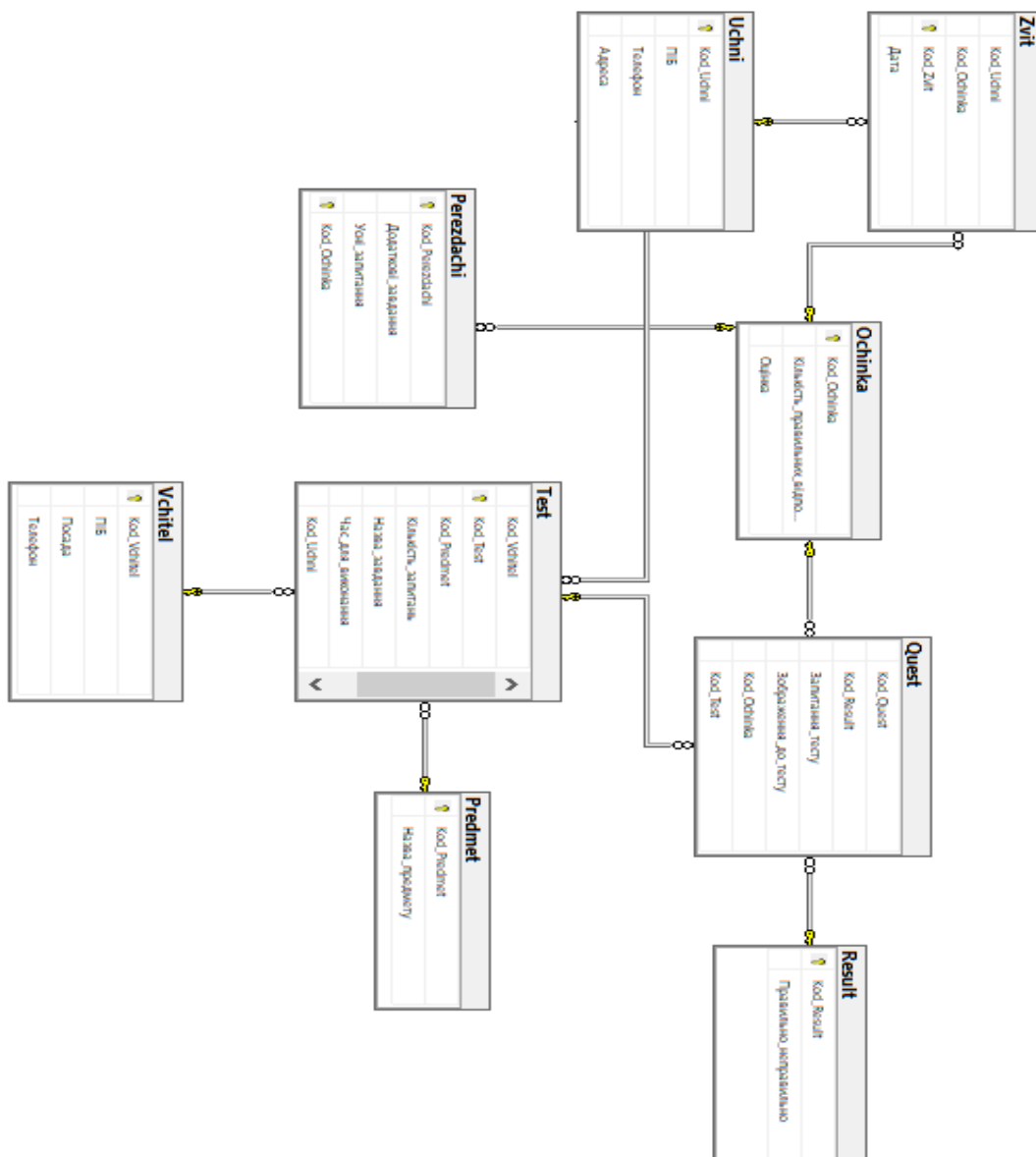


Рисунок Б.1 – Вигляд структури бази даних

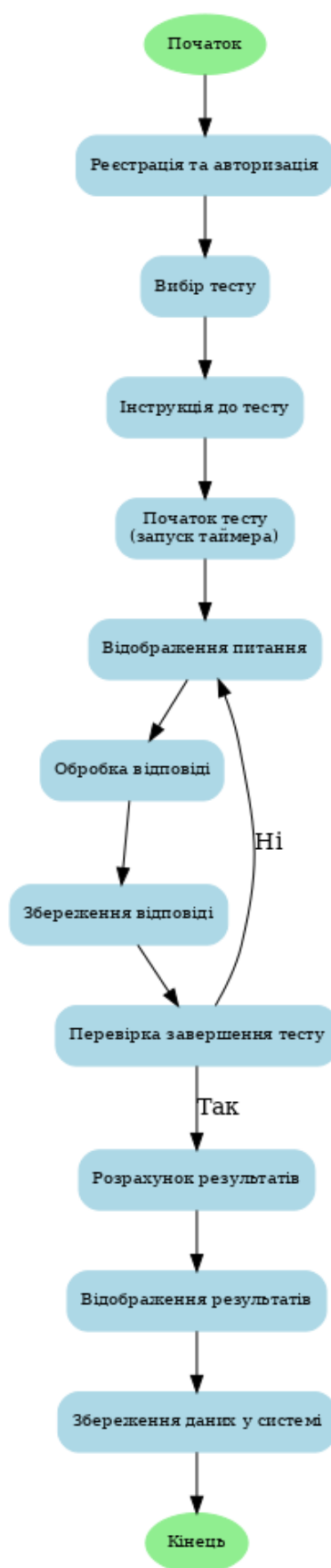


Рисунок В.1 – Схема процесу тестування