ЗМІСТ

[ВСТУП 3](#_Toc8546821)

[1. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 6](#_Toc8546822)

[1.1. Деталізація коду. 6](#_Toc8546823)

[1.1.1. Графічний інтерфейс. 6](#_Toc8546824)

[1.1.2. База даних. 6](#_Toc8546825)

[1.1.3. Регистрація та ідентифікація. 7](#_Toc8546826)

[1.1.4. Розподіл керування частинами проекту. 8](#_Toc8546827)

[1.1.4.1. Контролер проходження тестів. 8](#_Toc8546828)

[1.1.4.2. Контролер додавання матеріалів для тестів. 12](#_Toc8546829)

[1.1.4.3. Контролер рейтингової таблиці. 13](#_Toc8546830)

[1.1.4.4. Контролер налаштувань. 14](#_Toc8546831)

[2. ОПИС КОМП’ЮТЕРНОГО ТЕСТУВАННЯ ЗНАНЬ СТУДЕНТІВ 15](#_Toc8546832)

[2.1. Сторінка входу. 15](#_Toc8546833)

[2.2. Сторінка регистрації. 16](#_Toc8546834)

[2.3. Основна форма. 17](#_Toc8546835)

[2.3.1. Проходження легкого тесту. 18](#_Toc8546836)

[2.3.2. Проходження середнього тесту. 20](#_Toc8546837)

[2.3.3. Проходження складного тесту. 22](#_Toc8546838)

[2.4. Формування категорій для тестів. 24](#_Toc8546839)

[2.5. Рейтингові списки. 25](#_Toc8546840)

[2.6. Налаштування програми. 26](#_Toc8546841)

[ВИСНОВОК 28](#_Toc8546842)

[СПИСОК ЛІТЕРАТУРИ 29](#_Toc8546843)

# ВСТУП

Важливим аспектом, що визначає характер змін у системі освіти, є науково-технічний прогрес та його вплив на соціальні та суспільні відносини. Комп’ютерні технології постійно вдосконалюються, стають більш насиченими, ємними, гнучкими, продуктивними, націленими на різноманітні потреби користувачів. Промисловість та мас-медіа стали першими «сферами-користувачами» мультимедійних розробок. Навчання з використанням мультимедійних засобів також уперше здійснювалося у сфері виробництва для підвищення кваліфікації персоналу. Американські дослідники дійшли висновку, що використання засобів мультимедійних технологій у процесі навчання студентів дозволяє істотно підвищити показники змістового розуміння та запам’ятовування запропонованого матеріалу. Серед причин, частіше за інші, називали можливість синкретичного навчання (одночасно зорового та слухового сприйняття матеріалу), активну участь в управлінні поданням матеріалу, легке повернення до тих розділів, які потребують додаткового аналізу.

Сучасні освітні комп’ютерні програми (електронні підручники, комп’ютерні задачники, навчальні посібники, гіпертекстові інформаційно-довідкові системи – архіви, каталоги, довідники, енциклопедії, тестуючі та моделюючі програми-тренажери тощо) розробляються на основі мультимедійних технологій, які виникли на стику багатьох галузей знань. На нових витках прогресу відстань між новими технічними розробками та освітою скорочується.

Швидкість зміни інформації у сучасному світі висока, тому гостро постає питання формування у студента оптимальних комплексів знань і способів діяльності, формування інформаційної компетентності, що забезпечить універсальність її освіти. У розв’язанні цих проблем важливе місце займає комп’ютерне програмне забезпечення освітнього процесу.[1]

В даний час бурхливого розвитку комп'ютерних технологій усе більш актуальною задачею є розробка і використання в навчальному процесі матеріалу, що розробляються з застосуванням гіпертекстових і інших технологій. Такі системи називаються інтерактивними навчальними матеріалами та можуть використовуватися не тільки для денної, заочної та самостійної форм навчання, але і знайти широке застосування в дистанційній формі навчання .

Однією з гілок використання мультимедійних продуктів є контроль знань (тестування). Це являється однією з розповсюджених завдань буденної вчительської роботи. Багато ресурсів та часу в наші часи вчитель витрачає на проведення контролю знань, що негативно відображається і на вчителі ,і на учнях. Багато проблем існують і при усному опитування ,і при письмовому.

Тестування стає найбільш ефективним типом контролю знань в сучасний час. Тестування у відмінності від інших методів займає якомога менше часу та ресурсів. Однією з переваг тестування це те ,що результати можна отримати дуже швидко, а при комп’ютерному тестуванні відразу після закінчення тесту. Прикладами таких програм є: Айрен, MyTest, Quizlet, Proprofs. Кожна з цих програм має свої недоліки та переваги, але всі вони не надають можливість генерувати різні тести кожного разу при вході, що не надає унікальності.

Тестування – найсправедливіший метод, воно ставить усіх учнів в рівні умови, як в процесі контролю, так і в процесі оцінки, практично виключаючи суб’єктивізм викладача.

Тестові завдання можуть складатися з використанням різноманітних комп’ютерних інструментів, починаючи від різних редакторів і програм для розробки презентацій і до використання мов програмування і можливостей мережі Інтернет.

Комп'ютерне тестування повністю реалізується при проведенні контрольних робіт, контролю над самостійною роботою студентів, частково (комбінована форма: комп'ютерне тестування + традиційна співбесіда) – колоквіумів, заліків та іспитів.

Метою впровадження в Україні модульно-рейтингової системи є альтернативної форми контролю навчального процесу, шляхом формування системи внутрішнього контролю успішності студентів і оцінки рівня підготовки фахівців при освоєнні освітніх програм.

Актуальність застосування мультимедійних технологій в освітньому процесі обумовлена тим, що на сучасному етапі нашого суспільного розвитку відбувається інформатизація суспільства і широке поширення глобальної комп’ютерної мережі Інтернет.

Актуальність створення проекту обумовлена необхідністю створення сучасних комп’ютерних систем тестування з метою удосконалення навчального процесу для різних форм навчання.

## РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Деталізація коду.

Для створення програми була використана мова C#, так як вона надає можливість легкої інтеграції з графічним інтерфейсом, з використанням технології ADO.NET, а також систему керуванням базами даними MySQL.

### Графічний інтерфейс.

Windows Forms – інтерфейс програмування додатків (API), відповідальний за графічний інтерфейс користувача і є частиною Microsoft.NET Framework. Він спрощує доступ до елементів інтерфейсу Microsoft Windows за допомогою створення обгортки для WIN32 API в керованому коді. Всередині .NET Framework, Windows Forms реалізується в межах простору імен System.Windows. Forms.[2][3]

### База даних.

MySQL є системою керування базами даних (СУБД). На сьогоднішній день це одна з найпопулярніших систем управління базами даних.[4]

Вона має кроссплатформенність, є дистрибутивною під найрізноманітніші ОС, в тому числі найбільш популярні Linux, Windows, MacOS.

Гнучкість MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці виду MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, котрі підтримують транзакції на рівні окремих записів. Більш того, СУБД поставляється із спеціальним різновидом таблиць EXAMPLE, що демонструє принципи створення нових таблиць. Завдяки відкритій архітектурі і GPL-ліцензуванню, постійно з'являються нові типи таблиць.

Максимальний розмір таблиць в MySQL v.3.22 складає до 4 гігабайт, в наступних версіях обмежень немає.

Розмір таблиці обмежений її типом. У загальному випадку тип MyISAM відподіває граничному розміру файла в файлової системі. Наприклад, в NTFS цей розмір теоретично може бути до 32 ексабайт. У разі InnoDB одна таблиця може зберігатися в декількох файлах, які мають єдиний табличний простір. Розмір останнього може досягати 64 терабайт.

### Регистрація та ідентифікація.

Щоб зробити форму регистрації треба підключити базу даних, аби мати змогу перевірити унікальність людини чи ідентифікувати вже існуючу. Для цього використаємо технологію ADO.NET. Мова C# вимагає підключення такої бібліотеки: MySql.Data.MySqlClient.[5]

Для налаштування серверного підключення використовуємо рядок:

"server=localhost; user=root; database= database; password= password ", де

* *server* – розміщення сервера на комп’ютері;
* *user* – ім’я користувача бази даних;
* *database* – назва бази даних з якою будемо оперувати;
* *password* – пароль входу до бази даних.

Тепер ми можемо передати рядок підключення до об’єкту MySqlConnection, який дозволяє створити зв’язок між даними програми та базою даних.

Для передачі інформації ми будемо використовувати такі загальновживані команди:

* SELECT – отримання даних;
* INSERT – додавання даних;
* DELETE – видалення даних;
* UPDATE – редагування даних.

Для отримання вірного користувача робимо перевірку через логін, який не можливо змінити:

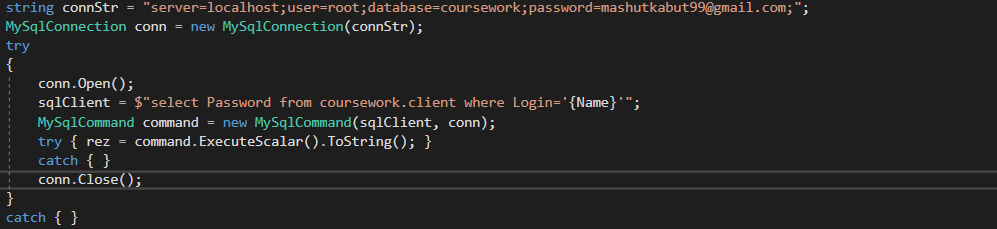


Рисунок 1. Ідентифікація користувача.

### Розподіл керування частинами проекту.

В графічному інтерфейсі присутні контролери, які дозволяють зробити програму більш привабливою та легкою у використанні. Кожен контролер має своє призначення (простий користувач бачить їх при натисканні кнопок меню):

* Проходження тестів;
* Додавання матеріалів для тестів;
* Рейтингова таблиця;
* Налаштування.

Розробнику це дає можливість структурувати код та розділити його на більш керовані частини.

Для такої реалізації в кожній кнопці меню основної форми будемо викликати метод BringToFront() для об’єкту цього контролера.[6]

### Контролер проходження тестів.

В цьому контролері присутні два ListBox, які відображають категорії, що беруться з бази даних, та рівні складності, котрі закладені програмно, а також кнопка для початку проходження тесту.

При натисканні кнопки треба зробити перевірку на заповнення таблиці, яка повинна мати більше десяти записів. В разі невдачі, видати помилку та надати можливість вибрати іншу таблицю чи перейти до іншого пункту меню.

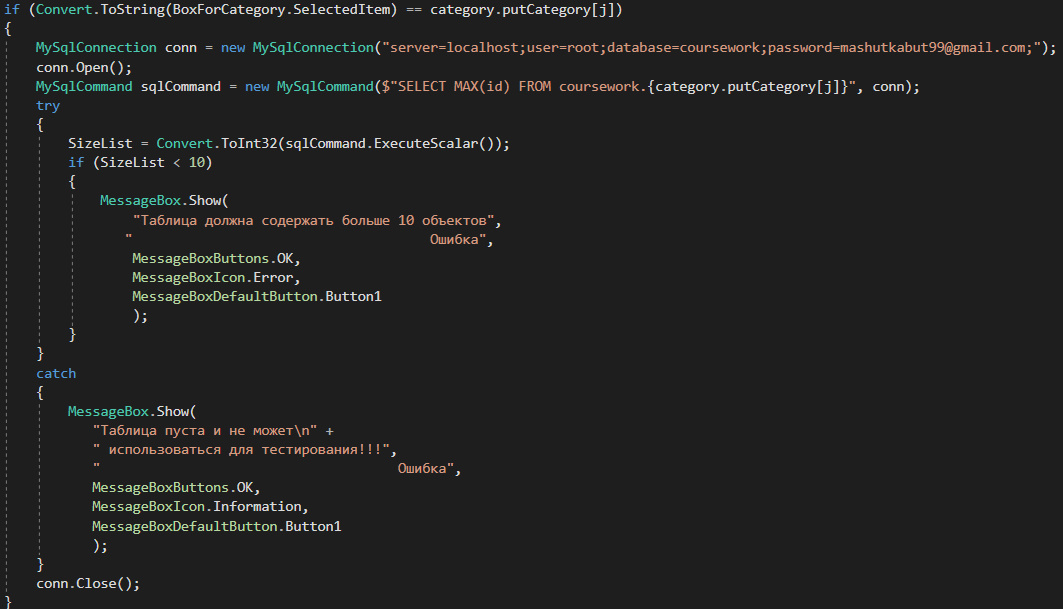


Рисунок 2. Обробка помилок вибору таблиці з даними.

У разі вдалого вибору, користувач має можливість вибрати складність тесту, що відобразиться в новій формі, яка не дасть можливість її зсунути або згорнути, щоб переглянути відповіді.

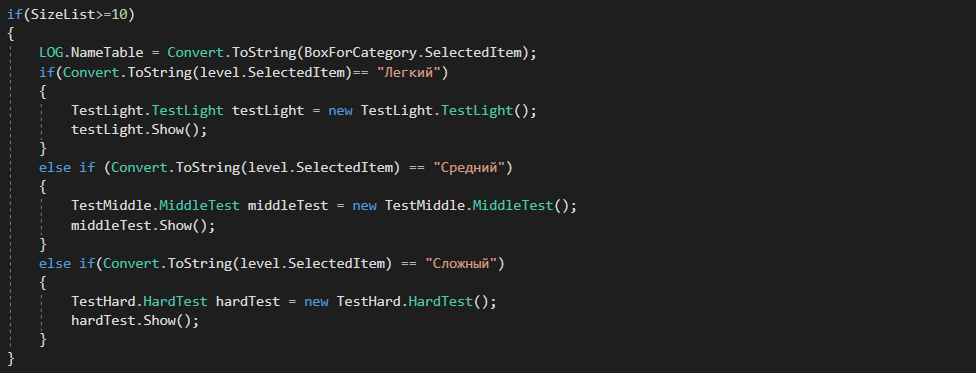


Рисунок 3. Вибір рівня складності.

***Перший тест – «Легкий».*** В ньому реалізовано десять запитань до кожного з яких маємо чотири відповіді.

Для випадкового розміщення відповідей в   
Button створимо об’єкт HashSet<int> Order, який реалізує випадкові числа, що не повторюються. За допомогою цього об’єкта генеруємо чотири числа, які відповідають за порядок розміщення відповідей до кожного питання.[7]



Рисунок 4. Генерація чисел для відповідей.

Для кожної кнопки встановлюємо єдиний колір. При натисканні він змінюється, аби користувачеві було відомо, який варіант він обрав. 

Рисунок 5. Зміна кольору при натисканні.

При завершенні тесту робимо перевірку чи на всі питання була надана відповідь. Якщо маємо помилку при заповненні - робимо сповіщення. За вдалого проходження тесту тестуючому надаються правильні відповіді та кількість балів, яку він набрав, а результати вносимо до бази даних.



Рисунок 6. Перевірка на коректність проходження тесту.

***Другий тест – «Середній».*** Середній тест складається з п’яти питань з чотирма відповідями та два блоки на відповідність по п’ять питань. Заповнення перших п’яти питань відбувається аналогічно «Легкому» тесту. Питання для блоку на відповідність беруться в порядку слідування зі списку, який взятий із бази даних. Порядок відповідей формується за допомогою вищезгаданого об’єкта HashSet<int> Order.

Аналогічно використовуємо кольори перших п’яти питань з тесту «Легкий». Для блоків на відповідність надаємо кожній кнопці власний колір. Щоб встановити відповідність треба клікнути на питання, аби отримати його колір.



Рисунок 7. Отримання кольору кнопки.

Далі цей колір буде встановлюватись на обрану користувачем клавішу блоку відповідностей.

В кінці робимо аналогічну перевірку на коректність заповнення тесту, аби не було двох відповідей на одне питання у відповідностях і була хоча б одна відповідь в перших п’яти питаннях.

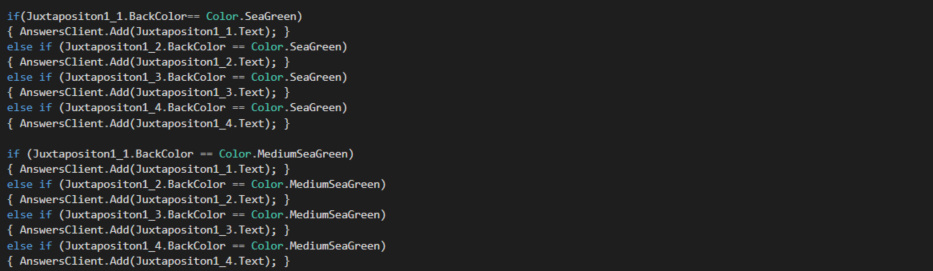


Рисунок 8. Перевірка належності відповідності.

Далі аналізуємо відповіді й видаємо користувачеві кількість запитань на які було дано вірну відповідь та відображуємо правильні значення, а в базу даних відправляється час, результат, складність тесту та прізвище тестуючого.

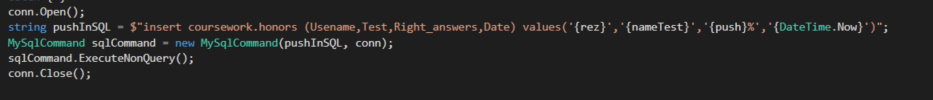


Рисунок 9. Запит на збереження даних після проходження тесту.

***Третій тест – «Складний».*** Тест представляє собою дванадцять питань, яким у відповідність ставиться двадцять вісім відповідей у випадковому порядку. Використовуємо подібну реалізацію, як в блоках на відповідність. Розміщення відповідей відбувається за алгоритмом методу роботи з об’єктом HashSet<int> Order.

Кожній кнопці встановлюємо колір і надаємо цей же колір відповіді, яку бажаємо обрати. В результаті проходження у тестуючого повинно бути на дванадцять питань дванадцять відповідей. Тоді користувачеві буде відображено правильні відповіді та кількість набраних балів. Аналогічно фіксуємо результат тестування до бази даних.

### Контролер додавання матеріалів для тестів.

Для відображення даних використовуються вже вищезгадані функції бібліотеки MySql.Data.MySqlClient.

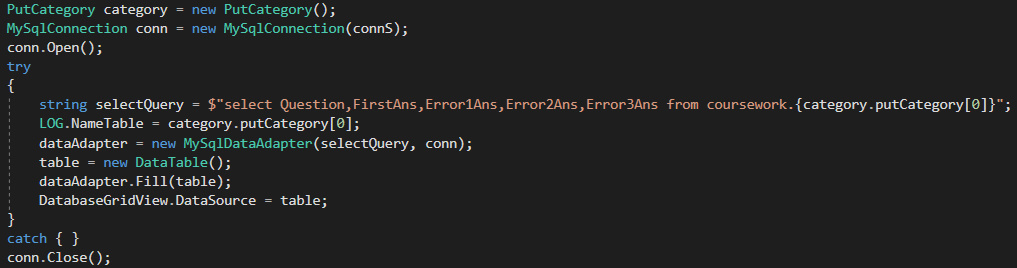


Рисунок 10. Отримання даних в DataGridView.

DataTable() відповідає за тимчасове зберігання інформації, яке буде відображено в DataGridView.

За допомогою роботи з DataTable() ми можемо додавати записи одним зручним та швидкодіючим методом:



Рисунок 11. Метод для зручної маніпуляції з базою даних.

Також користувачеві надається можливість створювати власні категорії. Для того, щоб були унікальні назви категорій, зробимо відповідну перевірку.

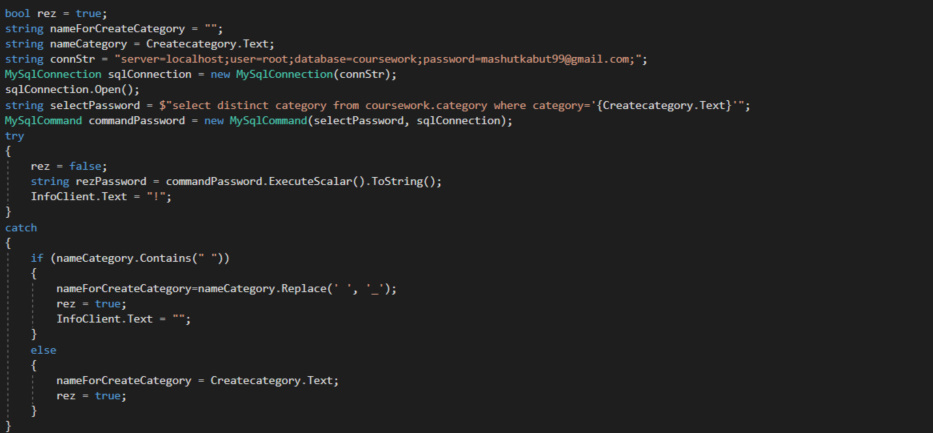


Рисунок 12. Перевірка на існування ідентичної назви категорії.

Після вдалого вводу назви, ми маємо можливість динамічно створити таблицю в базі даних.

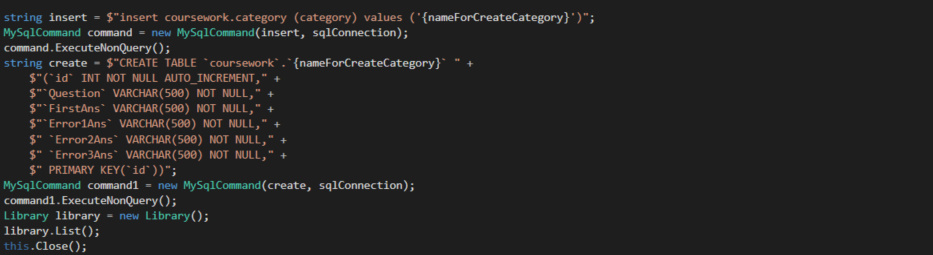


Рисунок 13. Динамічне створення таблиці для категорії.

### Контролер рейтингової таблиці.

Для відображення даних будемо використовувати також бібліотеку MySql.Data.MySqlClient та System.Data.

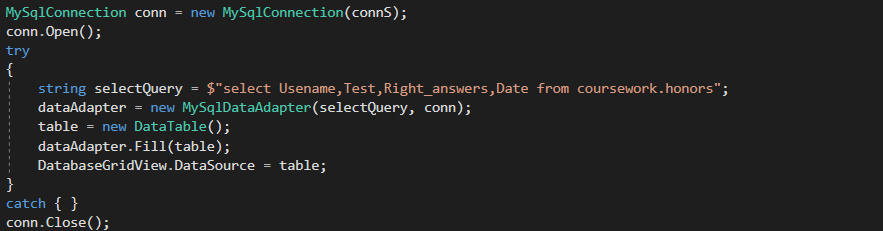


Рисунок 14. Отримання рейтингу з бази даних.

За допомогою MySqlDataAdapter(select,conn), який приймає два параметри:

* select - рядок, який визначає дані для зчитування;
* conn – відкритий сеанс роботи з базою даних.

Можемо отримати інформацію, що буде зображена користувачеві в DataGridView.

Сам рейтинг підраховується за формулою:

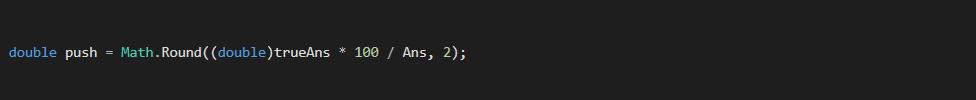


Рисунок 15. Розрахунок рейтингу.

* trueAns – кількість правильних відповідей після проходження тесту;
* Ans – загальна кількість питань.

### Контролер налаштувань.

В даному контролері реалізована взаємодія користувача з особистими даними (зміна імені, прізвища, пароля та видалення аканту). Для цього використані загальні команди SQL. Також ця сторінка надає змогу вибрати категорію та видалити з всією інформацією.

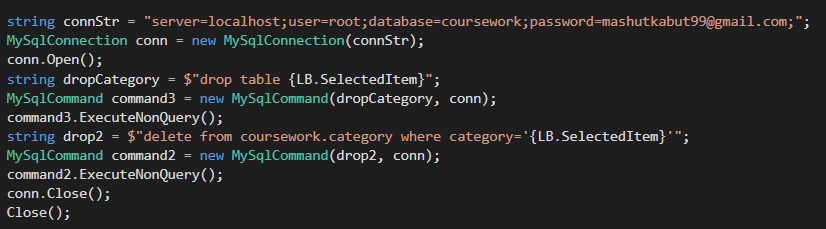


Рисунок 16. Видалення категорій та супутньої таблиці.

При видаленні профілю буде відображена початкова сторінка, де вже не можливо зайти під старим іменем.

# Опис комп’ютерного тестування знань студентів

### Сторінка входу.

При запуску програми користувачу відображається вікно персоналізації, де він може ввести свої данні для входу або зареєструватися



Рисунок 17. Вікно входу.

Якщо введені данні ,при вході, були некоректні абоненту буде видано повідомлення.



Рисунок 18. Вікно з помилкою входу.

### Сторінка регистрації.

В разі вибору регистрації клієнту відкриється вікно, де повинні бути введені ім’я , прізвище, логін та пароль, за яким він наступні рази буде входити до програми.



Рисунок 19. Форма регистрації.

Користувач може зіткнутись з двома помилками при регистрації- вже існуючий такий пароль або логін.

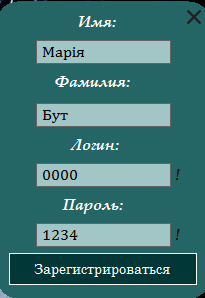


Рисунок 20. Помилки при регистрації.

### Основна форма.

При вдалому вході чи регистрації користувачеві відображається основна форма. На сам перед він бачить сторінку на якій може обрати той тест та його рівень складності, який бажає пройти.



Рисунок 21. Основна форма програми.

Якщо одна з категорій не має відповідну кількість полів для проходження тесту – буде відображене повідомлення.

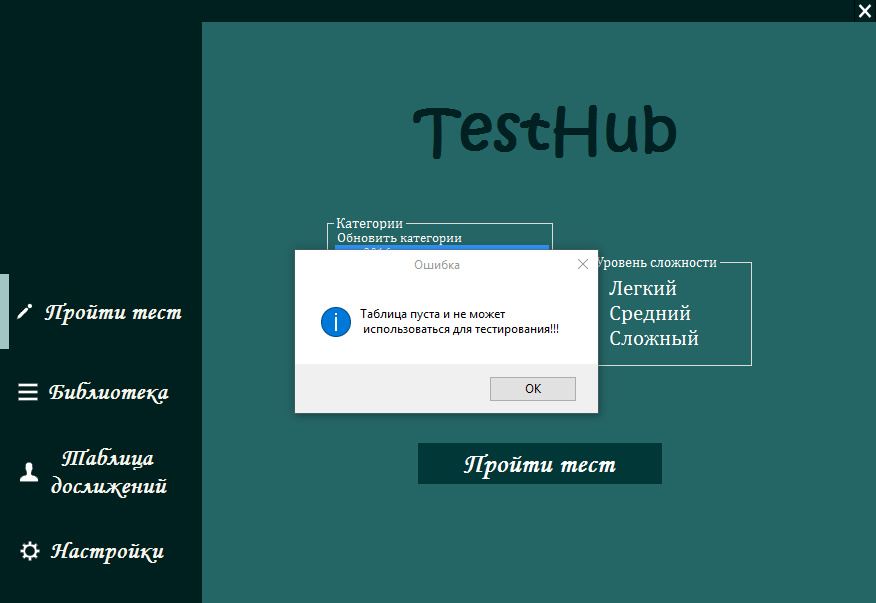


Рисунок 22. Помилка при виборі категорії.

### Проходження легкого тесту.

При виборі легкого тесту тестуючому відображається нова форма, яку не можливо усунути, аби уникнути шахрайства при проходженні. Тест складається з десяти питань з однією правильною відповіддю. При натисканні на обраний варіант кнопка змінює колір, щоб користувачеві було видно, який саме він обрав.

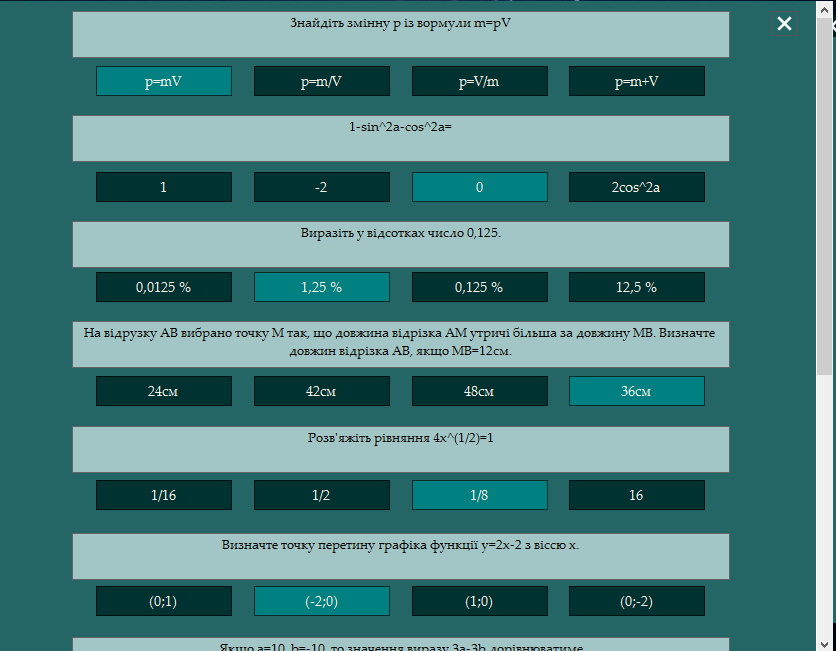


Рисунок 23. Проходження легкого тесту.

Після відповіді на всі питання буде відображено результат тесту та правильні відповіді.

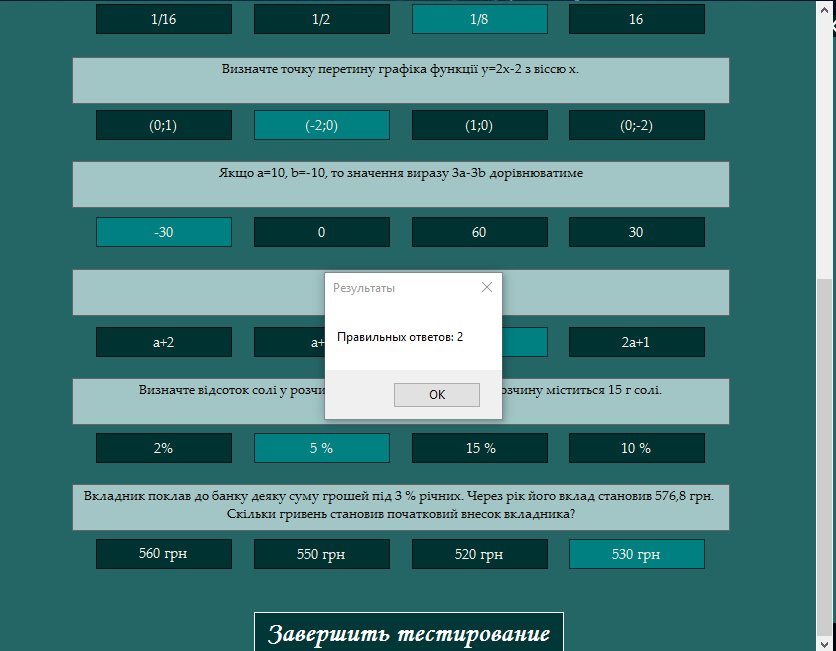


Рисунок 24. Кількість правильних відповідей після проходження тесту.



Рисунок 25. Відображення правильних відповідей.

### Проходження середнього тесту.

При проходження тесту середньої складності користувачеві відображається нова форма, аналогічна легкому тесту. На ній ми бачимо п’ять питань з однією правильною відповіддю та два блоки на відповідність. З першими п’яти питаннями ми маємо таке ж взаємодію як і з попереднім тестом.



Рисунок 26. Блоки на відповідність.

Аби поставити у відповідність питанню відповідь користувач повинен клікнути на питання на яке він хоче відповісти, а потім на відповідь, яку він вважає правильною. Таким чином колір відповіді прийме колір питання.

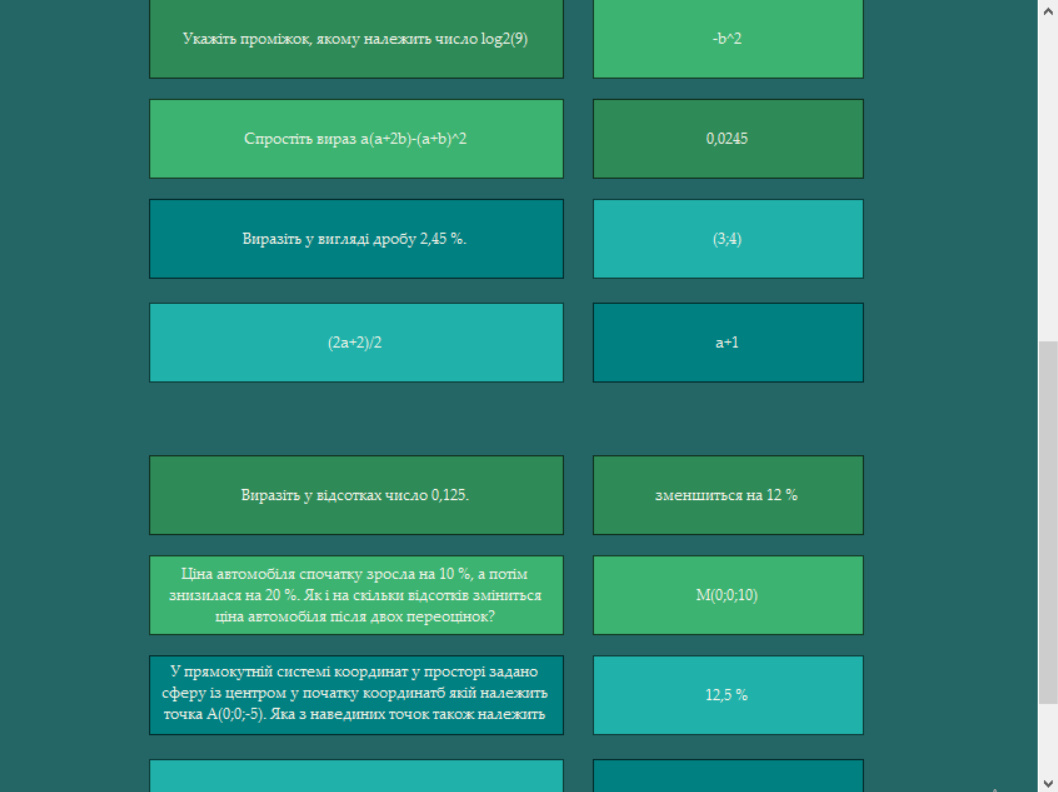


Рисунок 27. Встановлення відповідності.

Далі , як і в попередньому тесті, клієнту відображається результат. Також буде виведено правильні відповіді.

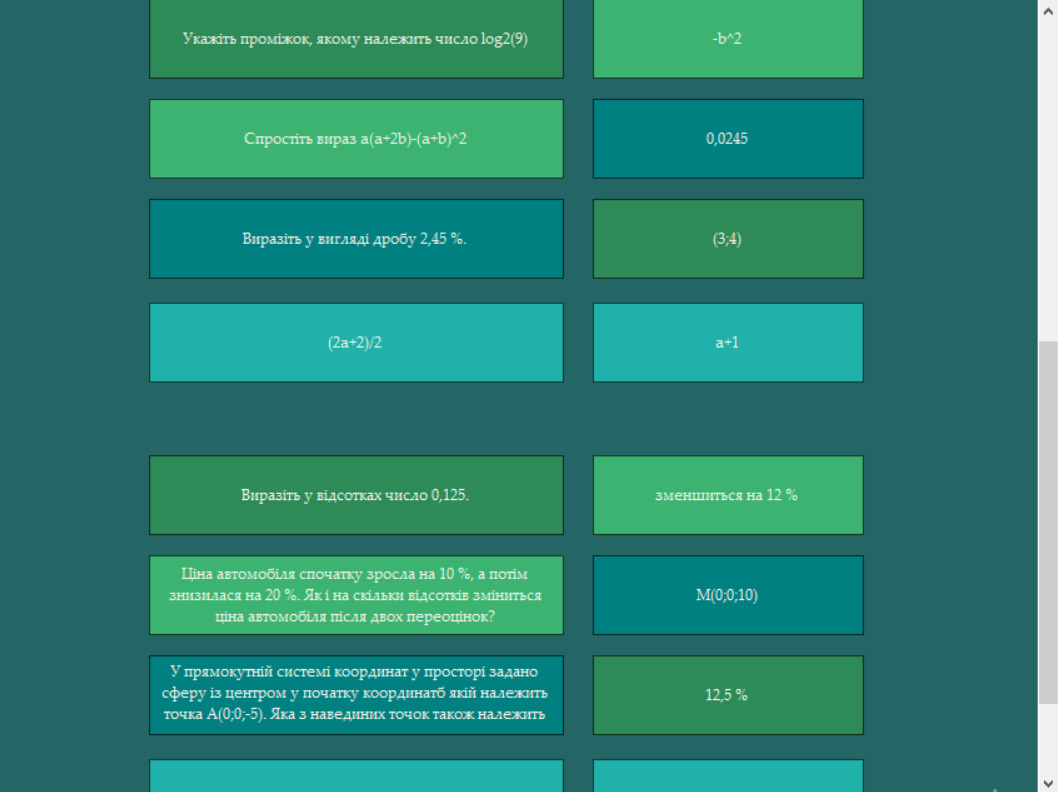


Рисунок 28. Зображення правильних відповідей.

### Проходження складного тесту.

Складний тест представляє собою дванадцять питань до яких надається двадцять вісім відповідей. На кожне питання є тільки одна вірна відповідь.



Рисунок 29. Вигляд складного тесту.

Для встановлення відповідності використовуємо аналогічний спосіб, як і в блоках тесту середньої складності.



Рисунок 30. Встановлення відповідності.

Далі надаються також можливості, як і в попередніх тестах: перевірити кількість правильних відповідей та побачити вірні варіанти.

### Формування категорій для тестів.

Для того, щоб створити категорію треба перейти до вкладки «Бібліотека». Користувачеві буде відображено поле з вже існуючими категоріями.

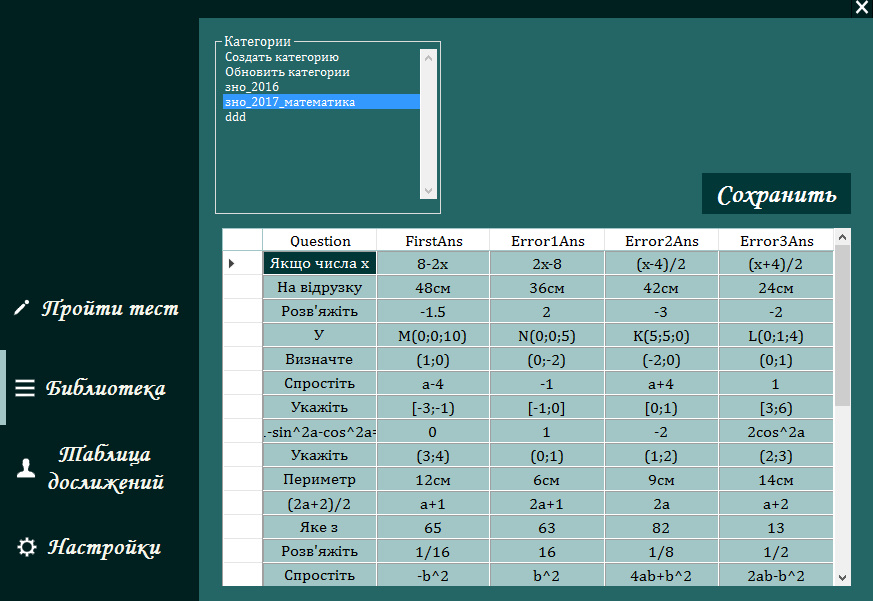


Рисунок 31. Вікно бібліотеки.

В цій формі клієнт може змінити, видалити та додати записи. Після всіх змін треба клікнути на «Зберегти», аби зафіксувати всі зміни в базі даних.

Для створення нової категорії потрібно обрати поле «Створити категорію». Після цього буде відображено вікно, до якого треба занести назву категорії.

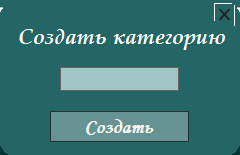


Рисунок 32. Створення категорії.

Тоді серед категорій буде відображена щойно створена, яка має поля для питання, правильної та трьох хибних відповідей.

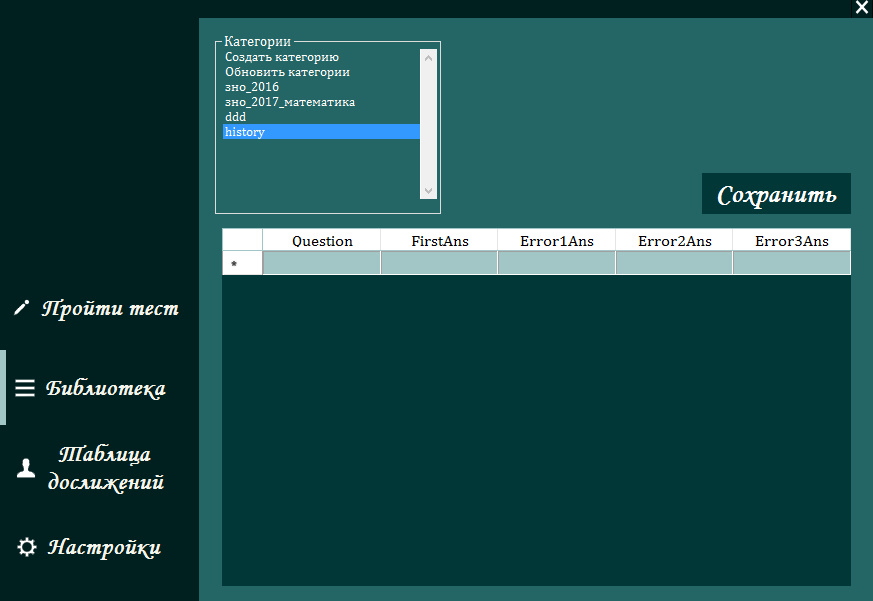


Рисунок 33. Відображення нової категорії.

### Рейтингові списки.

На момент завершення тесту всі результати зберігаються в базі даних. Їх можна переглянути за допомогою переходу на сторінку «Таблиця досягнень». Там ми можемо бачити результати всіх клієнтів, які представляють собою прізвище, складність тесту, результат у відсотках та дату і час проходження.

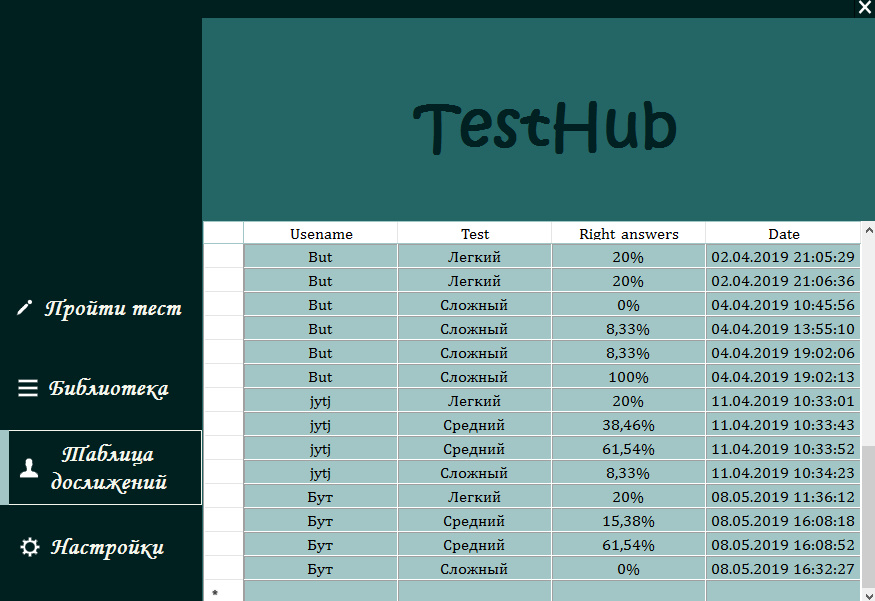


Рисунок 34. Рейтингова таблиця.

### Налаштування програми.

Користувачеві надається деякі можливості управління своїм профілем та категоріями.

Клієнт має повноваження змінювати своє ім’я, прізвище та пароль. В разі непотрібності аккауту, людина може його видалити. Після цього він буде переправлений на початкову сторінку входу.

Також надається можливість видалити категорію зі всіма її даними, яка потім відновленню не підлягає.

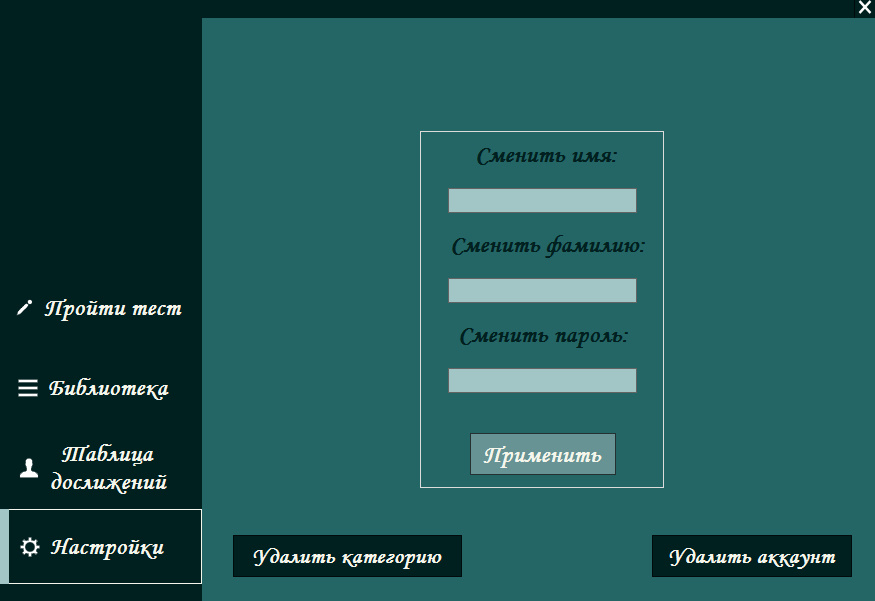


Рисунок 35. Налаштування.

# ВИСНОВОК

В даний час комп'ютери упроваджуються у всіх галузях людської діяльності. Вони допомагають автоматизувати процеси, на виконання яких потрібне багато часу. В цьому курсовому проекті я представила програму для тестування, яка може використовуватися в будь-якій сфері.

Метою даної роботи було створити програму для проведення математичного тестування та написати її якісно, тобто щоб текст програми не був занадто складним для сприйняття. Всі налаштування до завдань, кількості питань та їх зміст для зручності було винесено до бази даних, яку можливо редагувати, збільшувати чи зменшувати кількість запитань тощо.

До переваг даної програми можна віднести такі якості:

* Легка маніпуляція з даними;
* Створення власних тем для тестів;
* Абсолютно безкоштовна;
* Унікальність тестів для нового користувача;
* Необмежена кількість проходження тестів;
* Миттєві результати з відображенням помилок;
* Автономна робота без підключення до інтернету.

Недоліки:

* Тільки під платформу Windows;
* Мала кількість видів тестів;
* Низька продуктивність;
* Обмежена маніпуляція з даними.

В майбутньому планується вдосконалення програми за такими ознаками:

* Можливість працювати на різних операційних системах;
* Створення веб-додатку;
* Розширення варіантів тестів;
* Можливість формувати тести з мультимедійними додатками;
* Підвищити продуктивність;
* Вдосконалити роботу з реляційною базою даних.

В процесі написання курсової роботи я покращила свої навички написання коду на мові С# ,навчилась розробляти бази даних, зрозуміла алгоритм створення тестових вправ, розвинула аналітичні навички та концепцію створення бізнес-логіки програм.

# СПИСОК ЛІТЕРАТУРИ

1. <http://timo.com.ua/wiki/index.php/%D0%86%D0%A1%D0%A2%D0%9E%D0%A0%D0%86%D0%AF_%D0%A2%D0%95%D0%A1%D0%A2%D0%A3%D0%92%D0%90%D0%9D%D0%9D%D0%AF>
2. <https://metanit.com/sharp/tutorial/>
3. <https://metanit.com/sharp/windowsforms/>
4. <https://metanit.com/sql/mysql/>
5. <https://vscode.ru/prog-lessons/mysql-c-sharp.html>
6. <https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.usercontrol?view=netframework-4.8>
7. <https://professorweb.ru/my/csharp/charp_theory/level12/12_12.php>