**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**імені ІГОРЯ СІКОРСЬКОГО»**

**Навчально-науковий інститут прикладного системного аналізу**

**Кафедра системного проектування**

**КУРСОВА РОБОТА**

**з дисципліни «Об’єктно орієнтоване програмування»**

**на тему «Тестування»**

Виконав: студент 2 курсу групи ДА-21

спеціальність 122 «Комп’ютерні науки»

Аносов Ілля Віталійович

Керівник: к.т.н. Булах Б. В.

Національна оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_

Прийняли: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

Засвідчую, що у цій курсовій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_

Київ – 2024 рік

Зміст

[Вступ 3](#_Toc168256073)

[РОЗДІЛ 1 ПОСТАНОВКА ЗАДАЧІ 4](#_Toc168256074)

[Огляд існуючих рішень 4](#_Toc168256075)

[Google Forms 4](#_Toc168256076)

[SurveyMonkey 5](#_Toc168256077)

[Typeform 5](#_Toc168256078)

[Microsoft Forms 5](#_Toc168256079)

[Постановка задачі проектування 6](#_Toc168256080)

[РОЗДІЛ 2 РОЗРОБКА ДОДАТКУ 7](#_Toc168256081)

[Архітектура додатку 7](#_Toc168256082)

[Види тестів 8](#_Toc168256083)

[РОЗДІЛ 3 ОПИС РОЗРОБЛЕНОГО ПРОДУКТУ 10](#_Toc168256084)

[Опис головних структур і змінних програми 10](#_Toc168256085)

[Клас «Editor» 10](#_Toc168256086)

[Клас «StudentAnswers» 12](#_Toc168256087)

[Форма StartForm 14](#_Toc168256088)

[Форма TestEditForm 14](#_Toc168256089)

[Опис інтерфейсу та функціоналу 14](#_Toc168256090)

[Не авторизований 14](#_Toc168256091)

[Звичайний доступ 15](#_Toc168256092)

[Доступ адміністратора 17](#_Toc168256093)

[ВИСНОВОК 22](#_Toc168256094)

[СПИСОК ЛІТЕРАТУРИ 23](#_Toc168256095)

# Вступ

В сучасному світі технології постійно розвиваються і інтегруються в різні сфери життя, включаючи освіту та бізнес. Одним із важливих інструментів, що сприяють оптимізації навчального процесу та оцінки знань, є застосунки для тестування. Вони дозволяють автоматизувати процеси створення, проведення тестів та аналізу результатів, що значно економить час і ресурси як для викладачів, так і для студентів. Аналогічні інструменти використовуються і в корпоративному секторі для оцінки професійних знань працівників, проведення тренінгів та сертифікацій.

Застосунок для тестування, подібний до Google Forms, є зручним і ефективним засобом для організації тестів та опитувань. Google Forms зарекомендував себе як простий у використанні і водночас потужний інструмент, що забезпечує широкий спектр можливостей для створення різноманітних форм і анкет. Незважаючи на його популярність, існує потреба у розробці аналогічних рішень, які могли б бути більш гнучкими, інтегрованими в існуючі системи організації та налаштованими відповідно до специфічних вимог користувачів.

Метою даної курсової роботи є розробка застосунку для тестування, який дозволить користувачам створювати тести, проходити їх і отримувати аналіз результатів. Для досягнення цієї мети необхідно вирішити наступні завдання:

У роботі будуть використані методи об'єктно-орієнтованого програмування (ООП), проектування та аналізу програмних систем. Використання ООП дозволить створити модульну, гнучку та легкозмінну систему, що відповідатиме вимогам сучасного програмування.

Таким чином, розробка застосунку для тестування сприятиме підвищенню ефективності процесу оцінки знань та професійних навичок, а також створенню інструменту, який можна адаптувати для різних потреб користувачів.

# РОЗДІЛ 1 ПОСТАНОВКА ЗАДАЧІ

## Огляд існуючих рішень

### Google Forms

Google Forms, рисунок 1, є одним із найпопулярніших інструментів для створення форм та опитувань[1]. Основні переваги Google Forms включають:

1. Простота у використанні: Інтуїтивний інтерфейс дозволяє користувачам легко створювати форми без необхідності мати спеціальні технічні знання.
2. Інтеграція з іншими сервісами Google: Форми можуть легко інтегруватися з Google Sheets для аналізу даних, а також з іншими сервісами Google.
3. Багатофункціональність: Підтримка різних типів питань, включаючи текстові поля, варіанти вибору, списки, шкали та інші.
4. Безкоштовність: Більшість функцій доступні безкоштовно, що робить Google Forms доступним для широкого кола користувачів.

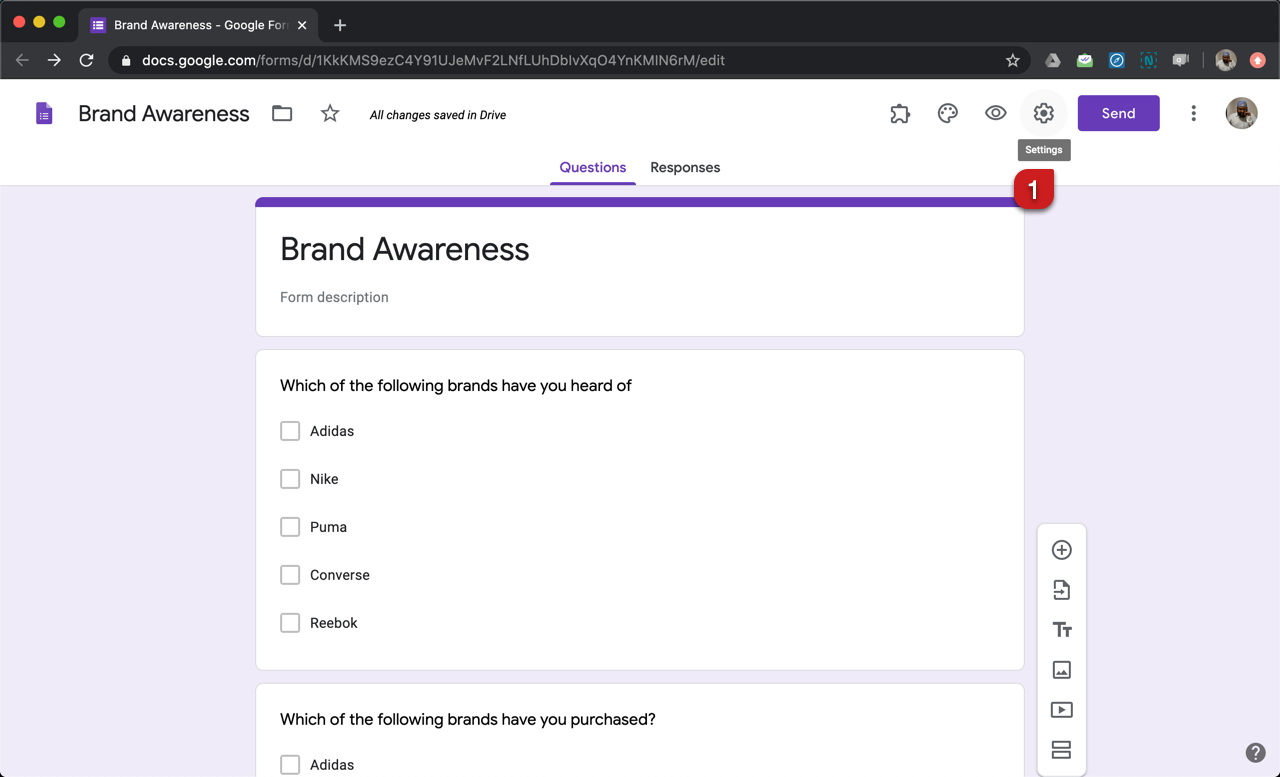


Рисунок 1 – Google Forms

Однак, Google Forms має деякі обмеження, зокрема обмежені можливості налаштування інтерфейсу та недостатню підтримку складних логічних операцій між питаннями.

### SurveyMonkey

SurveyMonkey є потужним інструментом для створення опитувань і тестів, що надає широкий спектр можливостей[2]:

1. SurveyMonkey пропонує багатий набір інструментів для налаштування зовнішнього вигляду форм, логіки питань та збору даних.
2. Потужні інструменти для аналізу даних, включаючи побудову графіків та діаграм, експорт даних в різні формати.
3. Підтримка різних рівнів захисту даних, включаючи шифрування та контроль доступу.

Незважаючи на свої переваги, SurveyMonkey має досить високі ціни на розширені функції, що може бути недоступним для невеликих організацій або окремих користувачів.

### Typeform

Typeform[3] пропонує сучасний підхід до створення форм та опитувань:

1. Фокус на інтерактивному та візуально привабливому дизайні, що робить процес заповнення форм більш цікавим для користувачів.
2. Можливість створення різних типів питань з логічними переходами та умовами.
3. Підтримка інтеграції з багатьма сторонніми сервісами, такими як Slack, Mailchimp, Google Sheets тощо.

Основними недоліками Typeform є обмежені функції в безкоштовній версії та можливі труднощі з використанням деяких складних функцій для новачків.

### Microsoft Forms

Microsoft Forms є частиною екосистеми Microsoft Office 365[4]:

1. Безшовна інтеграція з іншими сервісами Microsoft, такими як Excel, SharePoint, Teams.
2. Інтуїтивний інтерфейс, зручність створення і поширення форм.
3. Доступний для користувачів Microsoft 365 без додаткової оплати.

Проте, Microsoft Forms має менш розвинений функціонал у порівнянні з SurveyMonkey або Typeform, особливо в частині налаштування дизайну та аналітики.

## Постановка задачі проектування

Головним завданням роботи було створення користувальницького інтерфейсу, UI, для раніше створеної консольної версії з усіма основними функціями: додавання, редагування, видалення наборів тестів або певних з них, можливість проходження створених тестів користувачами, перегляд ними власних оцінок, або загальних результатів адміністраторами.

Для виконання роботи була обрана високорівнева мова С++. Однією з головних причин було те, що саме її ми вивчаємо на даному курсі, а отже напрацювання навичок роботи з нею стане гарною практикою. Також ця мова надає низькорівневий доступ до ресурсів комп’ютера і одночасно має ефективну систему керування пам’яттю, також її можливості дозволяють легко та зрозуміло реалізувати потрібний функціонал [5] [6]. Це стане в нагоді при роботі з великим двовимірним масивом.

Для створення графічного інтерфейсу для користувача було обрано технологію C++/CLI, це розширення стандарту C++, яке дозволяє писати керуючий код, що працює на платформі .NET. Використання C++/CLI дозволяє поєднати високу продуктивність коду на C++ з потужністю та зручністю .NET Framework[7]. Це особливо корисно для розробки застосунків з великим користувацьким інтерфейсом та розширеними функціональними можливостями.

# РОЗДІЛ 2 РОЗРОБКА ДОДАТКУ

## Архітектура додатку

Задачу можна розділити на дві частини: *Backend*(логічна частина) та *Frontend*(UI, те, що бачить користувач)

Backend складається з декількох класів, рисунок 2, що відповідають за:

* Поточного клієнта та доступні дії
* Усі створені тести, з функціями для їх редагування, додавання, видалення
* Збереження усіх відповідей студентів, отримані оцінки, витрачений час
* Види тестів, що реалізовані

Зображення, що містить текст, знімок екрана, Шрифт, схема

Автоматично згенерований опис

Рисунок 2 ­ Діаграма класів

Таким чином ця частина зберігатиме усі необхідну інформацію, надаватиме інформацію про те, що має бачити користувач та доступні йому дії. Frontend, у свою чергу, відповідає за:

* Графічний інтерфейс, те що може бачити користувач
* Реєстрацію дій користувача та виклик функцій з Backend

Отже у його відповідальність входить показ форм з різними елементами, наприклад для додавання нових тестів, перегляд, створення чи редагування існуючих наборів тестів, перехід між формами, показ тестів або інших можливих дій, та, у разі обирання певної дії, виклик відповідної функції з Backend.

## Види тестів

Для реалізації було обрано 2 види тестів: з однією, рисунок 3а та з множинною правильних відповідей, рисунок 3б. Діаграма створених класів наведена на рисунку 4.

|  |  |
| --- | --- |
|  |  |
| а) Одна можлива відповідь | б) Декілька можливих відповідей |

Рисунок 3 ­ Види тестів

Зображення, що містить текст, знімок екрана, число, програмне забезпечення

Автоматично згенерований опис

Рисунок 4 ­ Діаграма класів видів тестів

Для зазначених типів тестів необхідно застосування таких елементів, як Radiobutton та Checkbox. Radiobutton ­елемент з кнопкою у формі кола та текстовим полем праворуч. Такі елементи дозволяють активацію лише одного з наявних на формі або у групі, при натиску на Radiobutton попередній активований елемент деактивується. Checkbox подібний до Radiobutton, але має іншу форму кнопки та дозволяє одночасну активацію будь-якої кількості таких елементів.

Оскільки на одному екрані демонструються одразу декілька тестів, то Radiobutton працюватиме некоректно, якщо просто розміщувати його на формі. Тож для кожного окремого тесту створюється власний елемент GroupBox, який виступає в ролі контейнера і вирішує згадану проблему.

# РОЗДІЛ 3 ОПИС РОЗРОБЛЕНОГО ПРОДУКТУ

## Опис головних структур і змінних програми

У ході розробки застосунку було створено декілька класів, кожен з яких відповідає з окрему частину. Клас “ClientActions” є основним з згаданих класів для Backend, адже він поєднує усіх у собі.

Також можна виокремити інші 2 не менш важливі класи, що відповідають за Backend, у яких зберігається уся інформація та статистика: “Editor ” “StudentAnswers”

### Клас «Editor»

Цей клас слугує зберігачем створених тестів та дозволяє ними оперувати, рисунок 5. Усі тести зберігаються у змінну setOfTests типу unordered\_map, де ключем слугує назва набору, а зберігається вказівник на vector вказівників на абстрактний клас Question, від якого наслідуються усі види тестів.

Зображення, що містить текст, знімок екрана, число, Шрифт

Автоматично згенерований опис

Рисунок 5 ­ Діаграма класу Editor

Таблиця 1 ­ методи класу Editor

|  |  |  |
| --- | --- | --- |
| № | Прототип функції | Призначення функції |
| 1 | void addTest(std::string nameOfSet, Question\* question) | Додавання тесту до набору setOfTests |
| 2 | void addTest(int testSetNumber, Question\* question) |
| 3 | void addTestSet(std::string nameOfSet) | Додавання набору тестів до setOfTests |
| 4 | void changeTestSetName(std::string testSetName, std::string changedTestSetName) | Зміна назви набору тестів у setOfTests |
| 5 | void deleteSetOfTests(std::string nameOfSet) | Видалення набору тестів з setOfTests |
| 6 | void deleteSetOfTests(int setNumber) |
| 7 | void deleteTest(std::string nameOfSet, int nOfTest) | Видалення певного тесту з setOfTests |
| 8 | void deleteTest(int testSetNumber, int nOfTest); |
| 9 | bool deleteTestSetName(std::string nameOfSet) | Видалення ім’я тесту з nameOfTests |
| 10 | void editTest(std::string nameOfSet, int nOfTest, Question\* question) | Редагування певного тесту у setOfTests |
| 11 | std::vector<Question\*>\* getSetOfTests(std::string nameOfSet); | Отримання набору тестів з setOfTests |
| 12 | std::vector<Question\*>\* getSetOfTests(int setNumber) |
| 13 | Question\* getTest(std::string nameOfSet, int nOfTest) | Отримання певного тесту з setOfTests |
| 14 | Question\* getTest(int testSetNumber, int nOfTest) |
| 15 | bool replaceTestSetName(std::string nameOfSet, std::string changedTestSetName) | Заміна назви набору у nameOfTests |

### Клас «StudentAnswers»

Цей клас відповідає за збереження усіх відповідей користувачів, оцінок, та витраченого часу. Зберігаються у змінній studentAnswers типу unordered\_map. Оцінки або відповіді кожного користувача за певний тест зберігаються в окремих vector у згаданій хеш-таблиці. Завдяки цьому дані знаходяться у легкому доступі через, можна отримати через ім’я користувача та назву тесту.

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Рисунок 6 ­ Діаграма класу StudentAnswers

Таблиця 2 ­ методи класу StudentAnswers

|  |  |  |
| --- | --- | --- |
| № | Прототип функції | Призначення функції |
| 1 | void addNewTestMark(std::string nameOfTest) | Додає учню нову оцінку (при створені нового тесту і якщо учень його вже проходив) |
| 2 | bool addStudentAnswer(std::string studentName, std::string nameOfTest, StudentAnswerData\* marks) | Додавання нової відповіді учня |
| 3 | void changeTestSetName(std::string previousSetName, std::string newSetName) | Зміна назви набору тестів |
| 4 | bool deleteStudentAnswers(std::string studentName, std::string nameOfTest) | Видалення відповідей студента на певний набір тестів |
| 5 | void deleteStudentsAnswer(std::string nameOfTest, int testNumber) | Видалення відповідей усіх студентів на певний тест набору |
| 6 | void deleteTestAnswers(std::string nameOfTest) | Видалення відповідей усіх студентів на певний набір |
| 7 | std::vector<std::vector<std::string>\*>\* getAllTestAnswers(std::string nameOfTest) | Отримання усіх результатів тестів |
| 8 | StudentAnswerData\* getStudentAnswers(std::string studentName, std::string nameOfTest) | Отримання результатів певного учня з проходження набору тестів |
| 9 | bool isTestPassed(std::string studentName, std::string nameOfTest) | Перевіряє чи проходив учень певний набір тестів |

### Форма StartForm

StartForm є стартовою формою і фактично основною. Більшість дій відбуваються саме на ній. Так, наприклад вона активна при вході і як адміністратор, і як звичайний користувач та відображає набори тестів і кнопки, що відповідають типу входу. Також на ній відбувається проходження тестів, перегляд результатів користувача та загальних, та взаємодія з тестами наборів. Що робить цей клас величеньким та важливим.

### Форма TestEditForm

TestEditForm використовується для створення нових тестів або ж для редагування вже існуючих. При підтверджені закінчення взаємодії з новим тестом, створюється новий елемент типу Question\* відповідно до обраного типу тесту, а форма StartForm викликає функцію getQA(), завдяки чому отримує вказівник на новостворений тест.

## Опис інтерфейсу та функціоналу

У створеному застосунку використовується декілька форм, між якими відбуваються переходи під час виконання програми. У клієнта може бути 3 стани авторизації.

### Не авторизований

Перший станів авторизації ­ не авторизований. У такому стані користувачу показується форма для авторизації, рисунок 7, у якій він може ввести нікнейм.

Зображення, що містить текст, знімок екрана, Прямокутник, ряд

Автоматично згенерований опис

Рисунок 7 ­ Форма авторизації

Після отримання ім’я та натиску кнопки викликається функція logIn з класу ClientActions для відповідної спроби авторизуватися. У разі успіху ця форма змінюється наступною, інакше ж знову відкривається.

### Звичайний доступ

Другий стан ­ авторизований як звичайний користувач. Для того, щоб зайти з такими правами доступу слід ввести ім’я, що починається з “user”, наприклад “user123”. У цьому стані демонструється перелік усіх існуючих наборів тестів, рисунок 8.

Зображення, що містить знімок екрана, текст, програмне забезпечення, ряд

Автоматично згенерований опис

Рисунок 8 ­ Перелік усіх тестів

Навпроти кожної назви тесту наявна кнопка для початку проходження тесту, якщо його ще не пройдено, рисунок 9а, або кнопка для перегляду результатів, якщо тест вже пройдено, рисунок 9б.

|  |  |
| --- | --- |
| Зображення, що містить чорний, темрява  Автоматично згенерований опис | Зображення, що містить чорний, темрява  Автоматично згенерований опис |
| а)Початок проходження | б)Перегляд результатів |

Рисунок 9 ­ Кнопки навпроти назв наборів

При натиску першої з них, починається проходження відповідного набору тестів, рисунок 10, стартує відлік часу. Користувач має обрати бажані варіанти відповідей та натиснути кнопку “Finish”.

Зображення, що містить текст, знімок екрана, монітор, програмне забезпечення

Автоматично згенерований опис

Рисунок 10 ­ Проходження тесту

Після натиску кнопки завершення, відбувається збереження та аналіз усіх результатів та витраченого часу до відповідної структури класу StudentAnswers. Усі пройдені тести замальовуються зеленим, рисунок 11, або червоним кольором, рисунок 12, якщо відповідь правильна або не правильна відповідно. Біля кожного тесту зазначена отримана кількість балів.

Зображення, що містить текст, знімок екрана, ряд, Шрифт

Автоматично згенерований опис

Рисунок 11 ­ Відповідь правильна

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Рисунок 12 ­ Відповідь не правильна

Результати кожного клієнту з таким доступом зберігаються окремо та не доступні нікому, окрім “адміністратора”

### Доступ адміністратора

У такому випадку з самого початку на формі також відкрито перелік наборів тестів, рисунок 13.

Зображення, що містить текст, знімок екрана, монітор, програмне забезпечення

Автоматично згенерований опис

Рисунок 13 ­ Перелік наборів тестів для адміністратора

У кінці переліку знаходиться кнопка для додавання нових наборів, рис 14. Її натискання відкриває нову форму, рисунок 15, для введення назви створюваного набору.

Зображення, що містить чорний, темрява

Автоматично згенерований опис

Рисунок 14 ­ Кнопка створення нового набору тестів



Рисунок 15 ­ Форма редагування назви набору

На відміну від попереднього доступу, навпроти кожної назви набору перелік кнопок, рисунок 16, значно відрізняється, відповідно відрізняється функціонал.

|  |  |  |  |
| --- | --- | --- | --- |
| Зображення, що містить чорний, темрява  Автоматично згенерований опис | Зображення, що містить чорний, темрява  Автоматично згенерований опис | Зображення, що містить чорний, темрява  Автоматично згенерований опис | Зображення, що містить чорний, знімок екрана, білий, символ  Автоматично згенерований опис |
| а) Редагування назви | б) Перегляд набору | в) Перегляд результатів | г) Видалення набору |

Рисунок 16 ­ Кнопки взаємодії з наборами

Перша з кнопок, рисунок 16а, відповідає за зміну назви тесту. При натиску відкривається вікно для введення назви, аналогічне до вікна створення нового набору, рисунок 14. Після натискання кнопки для завершення редагування, стара назва замінюється новою.

Остання кнопка, рисунок 16г відповідає за повне видалення набору тестів з усіма результатами проходжень.

Друга з кнопок, рисунок 16б, відкриває перегляд відповідного набору тестів, рисунок 17. Перелік усіх тестів

Зображення, що містить текст, знімок екрана, монітор, число

Автоматично згенерований опис

Рисунок 17 ­ Перегляд обраного набору

У кінці переліку, що з’явився, знаходиться кнопка, схожа до кнопки створення набору, рисунок 14. При натисканні відкривається форма створення нового тесту, рисунок 18

Зображення, що містить текст, знімок екрана, монітор, число

Автоматично згенерований опис

Рисунок 18 – Створення нового тесту

На цій формі обирається тип створюваного тесту, вводиться питання, оцінка, можливі відповіді та обираються правильні. Приклад заповненої форми зображено на рисунку 19.

Зображення, що містить текст, знімок екрана, число, Шрифт

Автоматично згенерований опис

Рисунок 19 ­ Заповнена форма створення тесту

Після підтвердження створюється відповідний тест та додається до поточного набору, рисунок 20.

Зображення, що містить текст, Шрифт, знімок екрана

Автоматично згенерований опис

Рисунок 20 ­ Доданий тест

Навпроти кожного тесту наявні дві кнопки: редагування тесту, рисунок 21а та видалення тесту, рисунок 21б. Перша при натисканні відкриває вікно, аналогічне до вікна створення тесту, але вже із заповненими полями, тож воно має вигляд вікна з рис. 19

|  |  |
| --- | --- |
| Зображення, що містить чорний, темрява  Автоматично згенерований опис | Зображення, що містить чорний, знімок екрана, білий, символ  Автоматично згенерований опис |
| а) Редагування тесту | б) Видалення тесту |

Рисунок 21 ­ Кнопки взаємодії з тестами

Для повернення до перегляду наборів тестів використовується кнопка з рисунку 21. Вона Не доступна при проходженні тесту чи перебуванні на перегляді наборів тестів.

Зображення, що містить чорний, знімок екрана, темрява, білий

Автоматично згенерований опис

Рисунок 21 ­ Кнопка для повернення

При натисканні кнопки з рисунку 16в відкривається перегляд усіх результатів, тобто перегляд оцінок учнів, що вже пройшли тест, рисунок 22.

Зображення, що містить знімок екрана, текст, монітор, програмне забезпечення

Автоматично згенерований опис

Рисунок 22 ­перегляд результатів

Для зміни користувача у правому верхньому куті знаходиться елемент з ім’ям поточного користувача, при розгортанні якого з’являється кнопка виходу з акаунту, рисунок 23.

Зображення, що містить текст, Шрифт, ряд, число

Автоматично згенерований опис

Рисунок 23 ­ Вихід з акаунту

# ВИСНОВОК

У процесі виконання курсової роботи було розроблено застосунок для тестування, що має на меті спростити процес створення тестів, їх проведення та аналіз результатів. Застосунок, створений за допомогою C++/CLI, поєднує в собі продуктивність і гнучкість мови C++ з багатими можливостями платформи .NET. Для реалізації програми було розглянуто вже існуючі аналоги, наприклад Google Forms, MicrosoftForms, Typeform, SurveyMonkey. Це дозволило зрозуміти які функції необхідні для застосунку для тестування. Отримана інформація допомогла сформулювати вимоги для створюваного застосунку.

В першому розділі наведено теоретичні відомості про тести, різновиди застосунків з аналогічним призначенням. У другому розділі розглянуто основні задачі, які потрібно реалізувати, та шляхи виконання цього завдання. У третьому розділі наведено основні структури, що використовувались та задачі основних форм

Реалізовано backend, який обробляє логіку застосунку та взаємодіє з базою даних, а також frontend частину, що забезпечує зручний для користувача інтерфейс. Запрограмовано їх взаємодію з урахуванням можливих ситуацій неправильного введення.

Створено взаємодію між різними формами та передачу інформації між ними, що дозволило значно оптимізувати вигляд створеного користувальницького інтерфейсу та зробити його зручнішим. Реалізовано можливість створення нових тестів та взаємодію з вже створеними. Результати пройдених тестів наочно показуються тим, хто їх проходив за допомогою забарвлення у характерні кольори, а загальні результати доступні створювачу тестів.

Отже було створено інструмент, що може бути корисним у багатьох сферах, від освіти до бізнесу. У майбутньому можливе додавання більшого різновиду тестів та покращення аналізу результатів, що значно розширить функціонал та області застосування

# СПИСОК ЛІТЕРАТУРИ

1. ПЗ для адміністрування опитувань Google Forms

<https://www.google.com/intl/uk_ua/forms/about/>

1. ПЗ для аналізу брендів, ринку, досвіду роботи з співробітниками SurveyMonkey

<https://uk.surveymonkey.com/>

1. Платформа для онлайн опитувань Typeform

<https://www.typeform.com/>

1. ПЗ для адміністрування опитувань MicrosoftForms

<https://www.microsoft.com/uk-ua/microsoft-365/online-surveys-polls-quizzes>

1. Бьярне Стауструп Программирование: принципы и практика использования С++. Киев: Вильямс, 2011. 1225 с.

URL: <https://books.google.com.ua/books?id=kttwBgAAQBAJ&pg=PA2&hl=uk&source=gbs_selected_pages&cad=3#v=onepage&q&f=false>

1. Стенли Б. Липпман, Жози Лажойе, барбара Э. Му Язык программирования С++. Базовый курс

URL: <https://monster-book.com/yazyk-programmirovaniya-cpp-bazovyy-kurs>

1. .NET programming with C++/CLI Офіційний сайт Microsoft.

<https://learn.microsoft.com/uk-ua/cpp/dotnet/dotnet-programming-with-cpp-cli-visual-cpp?view=msvc-170>