

Добро дня, шановна комісіє!

Тема моєї курсової роботи : «Розробка програмного забезпечення для тестування знань студентів». Переді мною постали такі задачі :

- дослідити та провести аналіз предметної області;
- провести огляд і аналіз існуючих ПО, призначених для тестування;
- визначити функціональні вимоги до програмного забезпечення;
- вивчити відповідну технічну літературу по мовам програмування: C#, TypeScript;
- спроектувати та розробити web – додаток;

Актуальність моєї теми у тому, що комп'ютерне тестування зараз розвивається з шаленою швидкістю та все частіше використовується під час навчального процесу, але річ в тому, що існує небагато гідних Web або десктопних додатків, які б повністю були зручними у використанні та надавали потрібний функціонал.

Як ми знаємо, сьогодні тести активно використовуються в освітньому процесі. Це є гарний інструмент дистанційного контролю знань, а також поточної перевірки повноти засвоєння студентами понять, істотних положень окремих тем.

Як я вже казала, тестування є одним з різновидів контролю. Наявність механізму контролю в навчальному процесі відіграє значну роль у пізнавальній діяльності студентів.

Розрізняють дві функції контролю:

а) Для учнів – контроль забезпечує якість засвоєння знань, дає можливість зрозуміти помилки, вчасно їх виправити і краще розуміти наступний навчальний матеріал, а також формувати здатність до самоконтролю;

б) Для викладача – контроль дає інформацію про хід та якість засвоєння матеріалу, типові помилки, увагу та інтереси учнів, що дозволяє бачити свої дидактичні промахи і своєчасно вносити корективи в процес викладання.

Розглядаючи тестування, як метод педагогічного контролю, доцільно зупинитися на поняттях «тест», «тестове завдання», які нерідко ототожнюються, хоча є абсолютно різними видами педагогічної продукції.

Тест – система завдань специфічної форми, що застосовується в поєднанні з певною методикою вимірювання і оцінки результату.

Тестове завдання – це діагностичне завдання у вигляді питання з чіткою інструкцією до виконання і обов'язково з еталоном відповіді або алгоритмом необхідних дій.

Розглянемо деякі види тестових завдань :

### **Завдання закритого типу.**

До такого типу завдань відносять завдання з вибором однієї правильної відповіді, кількох правильних відповідей та вибір однієї, найбільш точної, відповіді.

### **Завдання на встановлення відповідності.**

Це завдання, де потрібно елементам однієї множини поставити у відповідність елементи іншої. Цей формат належить до категорії логічних пар і називається ще форматом розширеного вибору.

### **Завдання на встановлення правильної послідовності**

Завдання на встановлення правильної послідовності використовують в тих випадках, коли потрібно перевірити знання студентів правильної послідовності подій, дій, слів у визначеннях, правильний порядок членів у реченнях.

Найбільш широко в сучасній освіті застосовуються саме комп'ютерні тести. Застосування програм тестування та комп'ютерних тестів при перевірці знань є економічно вигідним і забезпечує підвищення ефективності навчального процесу. Важливу роль, при використанні комп'ютерного тестування, грає вибір програм для тестування знань, складання і створення тестів.

Розглянемо такі аналогічні Web – додатки.

Першим ми розглянемо web – додаток під назвою «Мастер-Тест». Використання програми дозволяє спростити роботу педагога при перевірці завдань, оскільки програма має можливість автоматично виставляти оцінки за п'ятибальною шкалою. Крім іншого, ресурс забезпечує перегляд вирішених завдань учнем з можливістю виявлення помилок і вірних відповідей. Але на жаль додаток має примітивний та не дуже зручний інтерфейс. Також для мене виявилась незрозумілою функція «надіслати запрошення», адже я реєструвалася під статусом «учня» і нащо мені відправляти запрошення моєму студенту, що більш властиве для вчителів?

Надалі розглянемо дещо цікавіший Web – додаток під назвою «ProProfs». «ProProfs QuizMaker» – це безкоштовний сервіс для створення тестів та інших навчально-контролюючих матеріалів.

Додаток надає багато можливостей, таких як : створення тестів різної складності та формату, проходження готових тестів, проходження різноманітних ігор для розвитку мозку, складання опитувань, вікторин та багато іншого. Також додаток підтримує функцію створення Classroom, підключення інструкторів та створення окремих груп для проходження тесту. Його недолік у тому, що немає розмежування між студентами та викладачами. Обидва можуть створювати, корегувати свої тести і ніхто не застрахований від помилково пройдених тестів.

Plickers – це сервіс, що дозволяє реалізувати швидкий зворотній зв'язок від класу (аудиторії, слухачів), підтримує мобільні голосування і фронтальні опитування під час навчального заняття з пройденого або поточного матеріалу, миттєвий облік відвідуваності заняття.

Особливість цього додатку у тому, що створення та перевірка результатів тесту виконується у Web – сторінці, а проведення самого тесту – безпосередньо за допомогою мобільного додатку. Для мене він виявився найбільш сучасним та зручним у використанні, але пропонує лише проведення тестів з однією правильною відповіддю або true/false. Для розширення можливостей треба купляти преміум аккаунту.

Розглянувши вищезгадані аналоги програмного забезпечення для тестування знань студентів та проаналізувавши їх функціональні сторони, я зробила висновки щодо функціональної частини свого Web – додатку.

Для розробки клієнтської частини своєї програми я використовую мову програмування TypeScript на базі фреймворку Angular. У майбутньому планується, для написання серверної частини, використання мови C# на базі фреймворку ASP.Net Core. Дизайн графічного інтерфейсу розроблявся у графічному редакторі Figma.

Розбір інтерфейсу почнемо з форм реєстрації та входу. Форма входу передбачає введення імені користувача та паролю. Якщо хоча б одне з двох полів заповнене не коректно, то програма видасть сповіщення о помилці.

Якщо користувач вперше користується Web – додатком, то йому необхідно зареєструватися. Для цього заповнити такі поля як ім'я користувача, електронна пошта, пароль та обрати собі роль : студент або викладач. Від цього надалі буде залежати наданий функціонал додатка.

Якщо користувач обрав роль «учень», то він має змогу проходити різноманітні тести з предметів, слідкувати за своєю успішністю за допомогою статистичних даних та приймати участь в обговореннях тестів.

Функціональні можливості вчителів трохи відрізняються. Вони можуть створювати тести. Тести діляться на три рівні : легкий, середній та важкий. Також викладач має змогу контролювати успішність цілої групи або конкретного студента у розділі «Статистика». Спільною є можливість додавання тестів у «збережені», щоб згодом повертатися до них через деякий час.

Легкий тест вміщує в себе максимум п'ятнадцять різноманітних тестових завдань легкого рівня, які розміщуються у задовільному порядку. Кількість завдань залежить від побажань викладача (від 5 до 15). Щоб створити тест треба обрати предмет за яким проводиться тест, додати назву та опис тесту і час, за який треба його виконати. Кожен тест «Легкого» рівня передбачає одну правильну відповідь з трьох.

Середній тест складається з трьох частин. Перша частина являє собою набір з тестових завдань «легкого» типу (максимальна кількість п'ять). Друга частина представлена у вигляді тестів з кількома варіантами відповідей (максимальна кількість п'ять). Третя частина – тести на встановлення послідовностей. Студентові надається чотири пункти які треба впорядкувати. Кількість таких тестів : три.

Важкий тест складається з усіх попередніх видів тестових завдань і до них ще додається тестове завдання на відповідність. Користувачеві пропонується з'єднати дві колонки (по чотири терміна у кожній) між собою. Кількість таких тестових завдань : два.

## **Висновки.**

Ознайомилася з різновидами тестових завдань та особливостями реалізації кожного з них.

Провела аналіз аналогічних програмних забезпечень. Склала функціональні вимоги для свого Web – додатку.

Детально розібрала теоретичний матеріал, що стосується мови програмування TypeScript та фреймворку Angular. Розробила інтерфейс клієнтської частини Web – додатка.

Надалі планується створення серверної частини, вдосконалення клієнтської, розширення функціоналу та можливостей користувачів.