

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ДЕРЖАВНИЙ ЗАКЛАД
„ЛУГАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА”**

Навчально-науковий інститут математики та інформаційних технологій

Кафедра математики та інформатики

Маніта Лідія Олександрівна

РОЗРОБКА СИСТЕМИ ОНЛАЙН ЕКЗАМЕНУВАННЯ

кваліфікаційна робота

**здобувача вищої освіти другого (магістерського) рівня
за спеціальністю 014.09 „Середня освіта. Інформатика”**

Особистий підпис _____ Лідія МАНІТА

Науковий керівник _____ Юрій КОЗУБ,
доктор технічних наук,
професор кафедри математики
та інформатики

Завідувач кафедри _____ Юрій КОЗУБ,
доктор технічних наук,
професор кафедри математики
та інформатики

АНОТАЦІЯ

Маніта Л.О.

Тема: Розробка системи онлайн екзаменування

Спеціальність: 014.09 „Середня освіта. Інформатика”

Установа: ДЗ ЛНУ імені Тараса Шевченка, 2024р.

Магістерська робота містить: 92 стор., 17 рис., 41 джерело, 2 додатки.

Об’єкт дослідження – процес створення програмного забезпечення для організації та проведення тестів і опитувань у навчальному процесі.

Предмет дослідження – технологія створення системи онлайн екзаменування.

Мета роботи - розробка автоматизованої системи онлайн екзаменування для організації та проведення тестів і опитувань у навчальному процесі. Для створення ресурсу використано мови програмування JavaScript, PHP, веб-сервер Apache, систему керування базами даних MySQL, технологію AJAX.

Результати роботи. Досліджено теоретичні аспекти проєктування автоматизованих систем тестування. Розглянуто методику використання існуючої системи тестування MyTestX, сформульовано критерії повноти та ефективності використання тестового додатку, проведено аналіз й обрано інструменти розробки. Розроблено програмний продукт, який дозволяє впроваджувати проведення тестів і опитування студентів, переглядати дані пройдених тестів і опитувань, а також створювати нові.

Ключові слова: WEB-РЕСУРС, ТЕСТИ, ОПИТУВАННЯ, НАВЧАЛЬНИЙ ПРОЦЕС, ПЗ, PHP, MySQL.

ABSTRACT

Manita L.O.

Topic: Development of online exam system

Specialty: 014.09 „Secondary education. Computer Science”

Institution: Taras Shevchenko National University of Luhansk

The master's thesis contains: 92 pages, 17 figures, 41 sources, 2 applications.

The object research – the process of creating software for organizing and conducting tests and surveys in the educational process

The subject of research – the technology of creating systems on the Internet

The purpose of the work – to develop automated online exam system for organizing and conducting tests and surveys in the educational process. Programming languages JavaScript, PHP, Apache web server, MySQL Database Management System, AJAX technology were used for creating a resource.

Results of work. The theoretical aspects of designing automated testing systems are investigated. The methodology of using the existing MyTestX testing system was considered, criteria of completeness and effectiveness of usage the test application were formulated, analysis was performed and development tools were selected. A software product that allows tests and surveys of students to view the data of passed tests and surveys, as well as to create new ones, has been developed.

Keywords: WEB-RESOURCE, TESTS, SURVEYS, EDUCATIONAL PROCESS, SOFTWARE, PHP, MySQL.

ЗМІСТ

| | |
|---|----|
| ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ | 6 |
| ВСТУП | 7 |
| РОЗДІЛ I. Дослідження теоретичних аспектів проектування автоматизованих систем тестування | 11 |
| 1.1 Основі положення тестології | 11 |
| 1.2 Вимоги до діагностичних тестів | 13 |
| 1.3 Основні форми та види тестів | 14 |
| 1.4 Складання тестів та визнання їх області змісту | 16 |
| 1.5 Принципи побудови автоматизованих систем контролю знань | 17 |
| 1.6 Системи контролю знань в умовах болонського процесу | 24 |
| 1.7 Висновок до розділу 1 | 28 |
| РОЗДІЛ II. Опис існуючої системи комп'ютерного тестування знань студентів | 29 |
| 2.1 Огляд програми для створення тестів | 29 |
| 2.2 Опис програмного забезпечення MyTestX | 33 |
| 2.3 Проектування моделі тестування | 36 |
| 2.4 Налаштування модуля тестування | 37 |
| 2.5 Роздача тестів | 38 |
| 2.6 Централізоване отримання результатів | 40 |
| 2.7 Збереження результатів | 42 |
| 2.8 Створення відгуку | 43 |
| 2.9 Висновок до розділу 2 | 44 |
| РОЗДІЛ III. Розробка веб системи екзаменування | 45 |
| 3.1 Загальний опис та аналіз архітектури | 45 |
| 3.2 Мова програмування PHP | 48 |
| 3.3 Опис веб-додатку | 50 |
| 3.4 Висновки до розділу 3 | 59 |

| | |
|---------------------------------|----|
| ВИСНОВКИ..... | 60 |
| ДОДАТКИ..... | 66 |
| Додаток А. Код account.php..... | 66 |
| Додаток Б. Код index.php..... | 84 |

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

| | |
|-------|---|
| АС | - автоматизована система; |
| АСНКЗ | - автоматизована система навчання і контролю знань; |
| БД | - база даних; |
| БЗ | - база знань; |
| ЕС | - експертна система; |
| КЗ | - контроль знань; |
| КНС | - комп'ютерна навчальна система; |
| ООП | - об'єктно-орієнтоване програмування; |
| ПЗ | - програмне забезпечення; |
| ПТМ | - понятійно-тезисна модель; |
| СЕТ | системи електронного тестування. |

ВСТУП

Сучасний розвиток суспільства неможливо уявити без використання комп'ютерних та мережових технологій. Технології, які раніше використовувались головним чином у військових та наукових сферах, сьогодні доступні для широкого використання і застосовуються в різних аспектах людського життя, зокрема в освітній сфері вищих навчальних закладів. З усіх комп'ютерно-орієнтованих технологій особливо слід визначити Internet-технології, які вдалим чином поєднують відкритість, універсальність і надійність. Це перспективна технологія навчання, що базується на комп'ютерних системах та забезпечує відкритий доступ до автоматизованих систем навчання та контролю, доступних для студентів та викладачів на будь-якому рівні, будь то університетський, національний чи світовий рівень інформаційних ресурсів. В мережових технологіях можна реалізувати різні методи та форми навчання, включаючи електронні посібники з віддаленим доступом, електронні бібліотеки, телеконференції для студентів та викладачів, а також системи контролю.

Тест є набором запитань, які зазвичай вимагають однозначної відповіді та складений за певними правилами та процедурами. Він передбачає попередню експериментальну перевірку та відповідає таким характеристикам ефективності, як валідність і надійність. Тестологія є міждисциплінарною наукою, яка займається створенням якісних та науково обґрунтованих методик вимірювання та діагностики для процесу тестування.

Тестування має три основні сфери застосування:

Освіта: Зростання тривалості навчання та складності учбових програм призводить до поширення тестування в освітній сфері.

Професійна підготовка і відбір: Зі зростанням темпу та складності виробництва важливим стає використання тестів у процесі професійної підготовки та відбору кадрів.

Психологічне консультування: З прискоренням соціодинамічних процесів тестування використовується для визначення рівня розвитку необхідних навичок, знань і особистих характеристик.

Використання онлайн-тестування для контролю знань набуває все більшої популярності. Автоматизовані системи контролю знань впроваджуються як окремі рішення, так і складові частини систем дистанційного навчання. В сучасній педагогічній практиці спостерігається перехід від простого оцінювання за кількістю правильних відповідей до складніших методів, що базуються на математичних моделях сучасної теорії параметризації та моделювання педагогічних тестів.

У контексті Болонської декларації контроль якості знань стає більш демократичним і якісним. Це виявляється в тому, що контролю піддаються невеликі обсяги навчального матеріалу, наприклад, модуль, що включає декілька тем навчального курсу. Загальна оцінка за навчальний курс обчислюється як середнє арифметичне оцінок за всі модулі. Це надає студентам можливість краще підготуватися до контролю, оскільки вони мають час опанувати менший обсяг навчального матеріалу перед кожним контролем, порівняно з традиційною схемою, де необхідно опанувати великий обсяг матеріалу перед підсумковим контролем у формі заліку чи іспиту.

Якщо окремі модульні оцінки не задовольняють студента, він має можливість покращити загальну оцінку за навчальний курс, подавшись на підсумковий контроль у формі заліку чи іспиту. [2].

Актуальність теми проєкту обумовлена необхідністю створення сучасних систем онлайн тестування з метою удосконалення навчального процесу для різних форм навчання, яка дозволить систематизувати та значно полегшити перевірку знань студентів з дисципліни для викладачів.

Метою виконання магістерської роботи є аналіз та розробка системи автоматизованого онлайн тестування в якому повинні максимально і лаконічно поєднуватись зручність використання, простота сприйняття

інформації і максимальне засвоєння навчального матеріалу. Таким чином, за допомогою онлайн тестування викладач може досить швидко побудувати систему контрольних або тестових завдань. При цьому значно полегшується процес обробки результатів і оцінювання.

Для досягнення мети необхідно вирішити наступні **завдання**:

- аналіз існуючих сучасних засобів та класифікація автоматизованих систем тестування;
- особливості існуючих сучасних системи тестування;
- аналіз критеріїв повноти та ефективності використання тестових додатків;
- огляд сучасних засобів, технологій та інструментів розробки нового програмного застосування;
- розробка програмного продукту, який дозволяє впроваджувати проведення тестів і опитування здобувачів освіти, переглядати дані пройдених тестів і опитувань, а також створювати нові.

Об’єкт дослідження – процес створення програмного забезпечення для організації та проведення тестів і опитувань у освітньому процесі.

Предмет дослідження – технологія створення системи онлайн екзаменування.

Гіпотеза – якщо в процесі навчання студенти будуть використовувати систему онлайн тестування, то це дозволить об’єктивно оцінити рівень знань та удосконалити якість навчання здобувачів освіти комп’ютерних спеціальностей.

Наукова новизна: розроблено систему комп’ютерного тестування для використання за будь-якою дисципліною засобами JavaScript, PHP, веб-серверу Apache, системи керування базами даних MySQL, технологією AJAX.

Практичне значення полягає у можливості створювати і впроваджувати проведення тестів і опитування здобувачів освіти, переглядати дані пройдених тестів і опитувань.

Особистий внесок магістра складається у:

- аналізі сучасних засобів автоматизованих систем тестування;
- виборі методів досліджень і технологій реалізації;
- створенні методики розробки програмного забезпечення;
- створенні нової комп'ютерної системи тестування.

Структура і обсяг роботи

Робота складається з вступу, трьох розділів, висновків списку використаних джерел, додатків. Обсяг роботи становить 92 сторінок, використаної літератури – 41 джерело.

В першому розділі проведено дослідження теоретичних аспектів проєктування автоматизованих систем тестування, сформульовано постановку задачі й розглянуто проблеми щодо організації контролю знань.

У другому розділі наведено аналіз сучасних технологій організації та проведення опитування і тестів. Представлено огляд Mytestx, розглянуто методику проєктування моделі тестування та технологію проведення тестів і опитування.

Третій розділ містить обґрунтування прикладних засобів програмування поставленої задачі проводиться опис програмної реалізації, зокрема надається інформація про різні програмні компоненти, які взаємодіють для забезпечення правильної роботи системи.

У додатках представлено лістинг коду розробленого програмного забезпечення.

РОЗДІЛ І

ДОСЛІДЖЕННЯ ТЕОРЕТИЧНИХ АСПЕКТІВ ПРОЄКТУВАННЯ АВТОМАТИЗОВАНИХ СИСТЕМ ТЕСТУВАННЯ

1.1 Основі положення тестології

Тестування (від англ. test - досвід, проба) - це метод психологічної діагностики, що застосовує стандартизовані тести - набори питань і завдань з певною шкалою значень. Цей метод використовується для стандартизованого вимірювання індивідуальних відмінностей. Знаходимо три основні області застосування тестування:

а) Освіта: З огляду на збільшення тривалості навчання та складності учбових програм.

б) Професійна підготовка і відбір: У зв'язку із зростанням темпу та ускладненням виробництва.

в) Психологічне консультування: З огляду на прискорення соціодинамічних процесів. Тестування дозволяє з високою ймовірністю визначити актуальний рівень розвитку необхідних навичок, знань та особистісних характеристик.

Процес тестування можна розділити на три етапи:

Вибір тесту: Це визначається метою тестування і рівнем достовірності та надійності тесту.

Проведення тесту: Цей етап визначається інструкцією до тесту.

Інтерпретація результатів: Визначається за системою теоретичних припущень щодо предмету тестування.

На всіх трьох етапах необхідна участь кваліфікованого психолога. У теорії тестів, яка є класичною як вітчизняною, так і зарубіжною, існує загальноприйнята класифікація завдань:

Завдання закритої форми (ЗЗФ): Вони також відомі як тести множинного вибору (ТМВ). Це завдання, де потрібно обрати вірну або вірні відповіді з запропонованих варіантів. Терміни "ТМВ" і "виборчі тести" є синонімами.

Завдання відкритої форми (ЗВФ): Це включає відкриті завдання, завдання на доповнення або завдання з вільно конструйованою відповіддю. Терміни "тест по методиці доповнення," "методика відновлення," і "клоуз-тест" є термінами, які можуть використовуватися взаємозамінно.

Завдання на встановлення відповідності (ЗВВ): Це завдання, де потрібно встановити відповідність між елементами.

Завдання на встановлення правильної послідовності (ЗВПП): Це включає завдання на систематизацію або угруповання, де потрібно визначити правильний порядок.

Сукупність загальних вимог до тестових завдань включає:

Кожне завдання повинно мати свій порядковий номер, який може змінюватися після об'єктивної оцінки трудності завдання та вибору стратегії його представлення.

Кожне завдання повинно мати еталон правильної відповіді.

Усі елементи завдання розташовуються на фіксованих місцях, в межах вибраної форми.

Завдання однієї форми супроводжуються стандартною інструкцією, яка передуює формулюванню завдань в тесті.

Для кожного завдання розробляється правило виставлення дихотомічної або політомічної оцінки і т.д. Після обробки тестових даних і інтерпретації результатів передтестові завдання набувають статус тестових, але в цю категорію потрапляють лише ті, які володіють хорошими статистичними характеристиками.

До атрибутів тестових завдань входять:

Рівень складності тестових завдань: Оцінка складності тестових завдань здійснюється різними методами в залежності від обраної теорії обробки емпіричних результатів виконання тесту.

Гомогенність тестових завдань: Під час конструювання тесту важливо чітко уявляти собі зміст завдань, які планується включити до остаточної версії тесту. Зміст завдань повинен відповідати властивості гомогенності, яка вказує на ступінь однорідності тесту з точки зору оцінюваної якості учня. Гомогенність, в цьому випадку, визначає, наскільки однорідним є завдання щодо вимірюваної властивості учня.

У педагогічних підсумкових тестах особлива увага приділяється рівню підготовки студента з даного предмету.

1.2 Вимоги до діагностичних тестів

Існує п'ять загальних критеріїв для контролю якості тестів, а саме:

Валідність: Це оцінка адекватності тесту та його ефективності. Валідність тесту вимірюється узгодженням його результатів із іншими критеріями, такими як успішність у відповідній діяльності. Цей показник відображає точність вимірювання досліджуваної властивості та його адекватність дослідженій проблемі. Виділяють такі види валідності, як очевидна, змістовна, конструктивна, критеріальна та конкурентна (порівняльна).

Визначеність (загальнодоступність): Ця вимога полягає в необхідності зрозумілості тесту для кожного учня, який має виконати завдання. Це важливо не лише для забезпечення розуміння кожним учнем того, що він повинен зробити, але й для уникнення різниці у правильних відповідях, які відрізняються від еталону.

Простота: Ця вимога передбачає, що всі завдання тесту повинні мати приблизно однаковий рівень складності, і тест не повинен включати завдань різного рівня складності.

Однозначність: Це визначення подібності оцінки якості тесту різними експертами. Для виконання цієї вимоги тест повинен мати чіткий еталон, який служить основою для однозначної оцінки.

Надійність: Цей аспект вказує на те, наскільки стійким є тест до впливу зовнішніх або випадкових факторів. Тест вважається надійним, якщо він дає подібні результати для кожного випробовуваного при повторному тестуванні, за умови, що рівень підготовки випробовуваних не змінився, а їх мотивація залишилася незмінною.

На практиці використовуються три основних методи оцінки надійності ретестування:

Повторне тестування: Цей метод включає проведення тестування тим самим тестом через два тижні після першого тестування.

Паралельне тестування: У цьому методі надаються дві паралельні форми тесту, які вимірюють однакову якість. Обидві форми тесту є еквівалентними і мають однаковий рівень складності.

Розщеплення: Цей метод включає розбиття завдань на парні і непарні, оскільки деякі форми завдань всередині тесту є паралельними. Це призводить до створення двох різних тестів, які можуть використовуватися для оцінки надійності тестування.

1.3 Основні форми та види тестів

Слід розрізняти два основні підходи до розробки тестів для конкурсного (професійного відбору) і для атестації учнів: нормативно-орієнтований і критеріально-орієнтований. Перший підхід дозволяє порівнювати учбові досягнення (рівень знань і умінь) окремих випробовуваних один з одним на основі розподілу балів. А критеріально-орієнтований підхід дозволяє оцінювати, в якому ступені випробовуваний оволодів необхідним для професійної діяльності учбовим матеріалом. Обидва ці підходи в рівній мірі

необхідні для створення діагностичних тестів в інтелектуальних повчальних системах.

Між нормативно-орієнтованими і критеріально-орієнтованими тестами існує ряд відмінностей, що полягають не в самих тестових завданнях, а в інтерпретації індивідуальних балів. Перша відмінність - цілі створення тесту. Нормативно-орієнтовані тести дозволяють оцінити відповідність знань і умінь випробовуваного деякій нормі: підходить - не підходить. Критеріально-орієнтовані тести дають можливість оцінки рівня того, хто навчається і ефективності програми навчання.

Друга відмінність – рівень деталізації області змісту. Для критеріально-орієнтованих тестів найчастіше потрібна велика деталізація. Третя відмінність – статистична обробка. Оброблені бали по результатах нормативно-орієнтованого тестування базуються на статистичних даних нормативної групи, тобто специфічної достатньо великій вибірці випробовуваних, для чого застосовуються спеціальні нормативні шкали. Крім того, існує і ряд інших відмінностей.

В даний час найбільш поширеними стали такі форми тестових завдань:

Закрита форма, що передбачає вибір одного або кількох правильних варіантів відповідей з числа запропонованих.

Форма на встановлення відповідності між двома запропонованими множинами.

Відкрита форма з обмеженнями на відповідь, де можна ввести як відповідь одне, так і кілька чисел, слів або формул.

Форма на встановлення правильної послідовності.

Впровадження завдань із багатоваріантними відповідями стимулює необхідність знаходження різних шляхів для вирішення задачі, що є важливим для досягнення основної мети навчання – уміння самостійно обирати спосіб виконання поставленого завдання.

Використання комп'ютерів розширює можливості застосування різних форм тестів, зокрема, дозволяє автоматизувати перевірку текстових відповідей.

1.4 Складання тестів та визнання їх області змісту

При вивченні будь-якої навчальної дисципліни, особливо важливими є теми, без яких неможливе засвоєння більш складного матеріалу під час навчання або які є необхідними для подальшої професійної діяльності. Заочування важливості певних розділів курсу можна врахувати, збільшивши частку питань, пов'язаних з цими розділами, в загальній кількості тестових завдань. Проте найбільш важливі розділи не завжди містять найбільше кількість матеріалу.

При створенні тестових завдань слід керуватися кількома правилами, які необхідні для створення надійного та збалансованого інструменту оцінювання знань. По-перше, слід аналізувати зміст завдань, забезпечуючи рівномірне представлення в тесті різних навчальних тем, понять і т. д. Тест не повинен бути перенасиченим другорядними термінами чи неважливими деталями, покладаючи акцент на механічну пам'ять. Формулювання завдань тесту повинно бути чітким, лаконічним і однозначним, щоб кожен учень розумів суть поставленого питання. Важливо виключити можливість використання одного завдання тесту для вгадування відповіді на інше.

Варіанти відповідей до кожного завдання повинні бути обрані так, щоб уникнути простої здогадки чи відкидання очевидно неправильної відповіді. Важливо вибрати найбільш прийнятну форму відповідей, враховуючи, що питання має бути сформульоване коротко, бажано стисло і однозначно. Наприклад, зручною може бути альтернативна форма відповідей, де учень підкреслює один із перерахованих варіантів: "так - ні", "вірно - невірно".

Завдання для тестів мають бути інформативними, охоплювати одне або декілька понять, визначень і т.д. При цьому важливо уникати надмірної

складності або занадто простоти тестових завдань. Кількість варіантів відповідей повинна бути, за можливість, не менше п'яти, при цьому невірні відповіді бажано відображати найбільш типові помилки.

Для атестації студентів можна використовувати критеріально-орієнтовані тести. При цьому необхідно вирішити завдання вимірювання рівня засвоєння матеріалу студентами великої області знань, навичок і умінь, з урахуванням важливості та обсягу вивченого матеріалу в різних розділах курсу. Для досягнення цієї мети слід:

1. Визначити сферу, зміст і мету тестування, провести аналіз навчальної дисципліни і відібрати матеріал для тесту.
2. Задати обмеження і вибрати підходи до процесу розробки, створити план тесту та його специфікацію.
3. Створити завдання і піддати їх аналізу експертами для оцінки відповідності області змісту та цілям тестування.
4. Провести пробне тестування та проаналізувати його результати.
5. Вибрати стандарти оцінювання.
6. Оцінити надійність і валідність тесту.
7. Завершити остаточне доопрацювання тесту та його паралельних форм.

1.5 Принципи побудови автоматизованих систем контролю знань

Визначимо основні принципи створення автоматизованих систем контролю знань (АСКЗ) нового покоління, що базуються на актуальних методах і моделях, розвиваючись в рамках теорії інтелектуальних обчислень та інженерії знань. Ці принципи визначають концепцію інтелектуального тестування, яке відповідає уявленню викладача про належну організацію процесу контролю та оцінювання знань, одночасно забезпечуючи реалізацію раніше неформалізованих педагогічних прийомів і методик:

1. Перехід від визначення істинності пропонованих варіантів відповідей у бінарній логіці ("правильно - неправильно") до більш

універсальної схеми оцінювання відповідей за допомогою функцій переваги, які визначені у термінах нечіткої логіки. Важливо зауважити, що такий перехід не виключає традиційного підходу, оскільки за сучасними уявленнями, бінарна логіка може розглядатися як окремий (точніше, дегенерований) випадок нечіткої логіки.

2. Перехід від індивідуального складання тесту до колективної експертної підготовки усіх етапів тестування, що підвищить довіру кінцевих користувачів до АСКЗ і поліпшить валідність результатів тестування.

3. Кількісна оцінка складності і важливості кожного тестового завдання за допомогою пропорційної цифрової шкали, що дозволить об'єктивно оцінювати продемонстровані знання.

4. Розподіл безлічі тестових завдань на тематичні підгрупи, елементи яких семантично корелюють один з одним, із обов'язковим ранжуванням як внутрішніх, так і загальних тематичних груп. Впровадження цього принципу створить об'єктивну основу для формалізації ряду використовуваних в наш час "ручних" методик контролю знань, таких як оцінка широти або глибини знань, тести з підвищеною або зниженою складністю та інші.

5. Перехід від звичайного використання програмно реалізованих алгоритмів прямого тестування (де вибір наступного завдання практично не залежить від відповідей на попередні питання) до їх модульного складання при підготовці тесту, а також до створення адаптивних алгоритмів тестування, де вибір i -го завдання зумовлений відповідями на попередні $(i-1)$ -е, $(i-2)$ -е та інші етапи тесту. Впровадження цього принципу дозволить формалізувати широко використовувані в педагогічній практиці методики додаткових, настановчих і уточнюючих питань.

6. Побудова, стандартизований опис і уніфікована реалізація різноманітних алгоритмів тестування в межах однієї АСКЗ, що втілюють різні методики контролю знань, а також можливість вибору в конкретних ситуаціях

тих методик, які відповідають нормативним вимогам або визначаються власними перевагами організатора тестування.

7. Створення інструментарію для створення, налаштування та зміни різних шкал підсумкового оцінювання знань, включаючи можливість регулювання кількості і ширини оціночних інтервалів, а також визначення та варіювання зон невизначеності оцінок. Це надає можливість проводити параметричний аналіз валідності проміжних і підсумкових результатів тестування.

8. Автоматизація найбільш трудомісткого етапу підготовки до тестування, пов'язаного з формуванням великої кількості тестових завдань і варіантів відповідей на них. Цю процедуру можна ґрунтувати на формалізованій моделі знань з даної дисципліни у вигляді структурованої семантичної мережі, використовуючи відомі технології фреймів з інженерії знань.

Застосування цих принципів дозволяє говорити про виникнення нового класу систем тестування - Інтелектуальних автоматизованих систем контролю знань (ІАСКЗ).

Інтеграція інформаційних технологій у навчальний процес сприяє впровадженню нових організаційних форм та методів навчання, що має ряд переваг, включаючи:

- Збереження часу завдяки автоматизації операцій обчислювального характеру.
- Підвищення наочності та полегшення сприйняття навчальної інформації через компактне та чітке її представлення.
- Інтенсифікація навчання через алгоритмізацію процесу розв'язування завдань.
- Розширення та поглиблення змісту навчання за рахунок організації експериментально-дослідницької діяльності студента на основі моделювання процесів і явищ.

- Забезпечення оперативного контролю за результативністю навчання.

Інформаційні технології вводять в навчальний процес нові засоби управління самостійною роботою студентів, такі як:

- Засоби надання навчальної інформації, такі як довідкові системи, електронні енциклопедії та словники, комп'ютерні навчальні курси.
- Засоби підтримки предметної та професійної діяльності.
- Засоби опрацювання, оформлення та презентації результатів самостійної роботи.
- Засоби автоматизованого контролю знань.
- Засоби подання навчально-методичного забезпечення дисципліни.
- Автоматизовані засоби реєстрації та рейтингового оцінювання поточних результатів навчальної діяльності.

Ці інструменти дозволяють студентам виконувати різноманітні завдання, розпочинаючи від опрацювання лекційного матеріалу і вивчення нового матеріалу до виконання творчих завдань, таких як підготовка доповідей, наукових повідомлень і виконання курсових та дипломних робіт.

До засобів підтримки предметної та професійної діяльності варто віднести програмні середовища, віртуальні лабораторії, імітаційні системи та інші засоби, які спрямовані на залучення студентів до експериментального дослідження. Застосування цих інструментів в навчанні ознайомлює студентів із професійними задачами, які можна вирішити за допомогою цих технологій, та розвиває їхні навички використання комп'ютера як інструменту у професійній діяльності. Використання засобів підтримки предметної та професійної діяльності в самостійній роботі дозволяє реалізувати різні види завдань, включаючи репродуктивні (осмислення лекційного матеріалу), частково-пошукові (опрацювання теоретичного матеріалу на динамічних моделях або закріплення знань за заданим алгоритмом тощо) та творчі роботи (планування та проведення комп'ютерного експерименту для вивчення

певного явища чи об'єкта, розробка оригінальної власної моделі тощо). Важливо відзначити, що самостійні роботи, орієнтовані на комп'ютерний експеримент, перетворюють студента не лише на спостерігача, але й на активного учасника навчання. Процес засвоєння знань стає результатом його особистої діяльності.

До засобів опрацювання, оформлення та презентації результатів самостійної роботи можна віднести програми статистичної обробки даних, табличні процесори, текстові редактори, системи підготовки електронних презентацій та інші інструменти. Ці засоби є необхідною складовою навчального процесу, оскільки вони ефективно економлять час, ресурси та зусилля при виконанні різних завдань, таких як розрахунки чи графічне відображення. Оформлення різноманітної документації, оброблення та подання результатів лабораторних досліджень, підготовка рефератів, курсових та дипломних проєктів у електронному форматі стає необхідним елементом навчальної діяльності, і сприяє уніфікації та полегшенню цих процесів для студентів.

Засоби автоматизованого контролю знань використовують ідеалізовану модель очікуваних знань і вмінь студента з метою перевірки їх відповідності цій моделі. Це досягається за допомогою спеціально розроблених тестових завдань, які структуровані за елементами навчального матеріалу та визначеним рівнем його освоєння. Сучасні засоби контролю автоматично нагромаджують статистичні дані про користувача, щоб оцінити результати його роботи.

Автоматизовані системи контролю знань можна використовувати як у режимі контролю, так і у режимі самоперевірки, наприклад, перед колоквиумом, контрольною або лабораторною роботою. Важливо відзначити, що останній режим, по-перше, навчає студентів самоконтролю та підвищує їхню самостійність, а по-друге, відокремлення контролю від оцінки сприяє пізнавальній активності. Тут метою та мотивом дій є бажання отримати

об'єктивну інформацію про рівень власних знань для свідомого планування подальших навчальних заходів.

Засоби автоматизованого контролю знань є ефективним інструментом для аналізу результативності навчання. З їх допомогою можна здійснювати репродуктивні самостійні роботи студента, такі як відпрацювання математичних навичок або закріплення знань з правопису чи перекладу.

До засобів представлення навчально-методичного забезпечення дисципліни варто віднести електронні каталоги навчальної літератури та дидактичних ресурсів у мережі Інтернет, а також електронні банки індивідуальних завдань. Програму дисципліни та заплановані контрольні заходи можна зручно розміщати на сайті освітнього закладу, де студент має можливість ознайомитися з обсягом самостійної роботи, графіком виконання, вимогами до оформлення та оцінювання результатів своєї діяльності.

Сучасний рівень розвитку інформаційних та телекомунікаційних технологій дозволяє вільний доступ до інформаційних ресурсів, які розміщені в мережі Інтернет. Ця найбільша інформаційна система містить різноманітну інформацію, накопичену людством, представлену у формі електронних джерел, включаючи довідкову, навчальну та наукову інформацію. Вивчення будь-якої дисципліни ускладнюється фактором швидкого застаріння навчального матеріалу через динамічний розвиток сучасних наук.

Створення електронних каталогів дидактичних ресурсів Інтернету є доцільним заходом в контексті зазначених тенденцій.

Автоматизовані засоби реєстрації та рейтингового оцінювання результатів навчальної діяльності дозволяють ефективно обліковувати та зберігати індивідуальні завдання для самостійної роботи студентів. Вони також дозволяють проводити поточний аналіз діяльності студента, враховувати його досягнення і визначати поточний рейтинг.

Використання комп'ютера сприяє якісним змінам в рівні самостійної роботи студентів через посилення їхньої мотивації навчання, розширення

можливостей подачі інформації, активізації навчально-пізнавальної діяльності. Крім того, він дозволяє розширити та ускладнити навчальні завдання та впроваджує об'єктивний контроль за діяльністю студентів, призводячи до скорочення часу контролю та оперативного отримання зворотного зв'язку.

Проведений аналіз можливостей комп'ютерних педагогічних засобів дозволяє визначити рівень управління самостійною роботою студентів при їх застосуванні. Використання засобів автоматизованого контролю знань та програм-тренажерів у навчальному процесі передбачає жорстке (пряме) управління навчальною діяльністю студента з боку комп'ютера. Студент продовжує свій прогрес до кінцевого результату відповідно до траєкторії, визначеної викладачем.

Використання електронних енциклопедій, словників, довідкових систем, комп'ютерних навчальних курсів та інструментальних засобів для створення моделей не обмежує сферу діяльності студента. Ці засоби дозволяють студентам реалізовувати себе та підтверджувати свою компетентність. Взаємодія в системі "студент–комп'ютер" базується на принципах самоуправління, де студент визначає цілі своєї роботи, обирає методи та способи їх досягнення. Таким чином, відбувається збагачення, нарощування та трансформація власного досвіду, що визначає індивідуальний розвиток студента та підвищення рівня його самостійності. Комп'ютер виступає лише як зручний та потужний інструмент для досягнення конкретного результату.

Робота з імітаційними моделями та предметно-орієнтованими середовищами включає в себе створення викладачем початкової моделі дослідження та формулювання проблемних завдань для подальшого пошуку рішень. Студент взаємодіє з цими завданнями, сприймає проблему, планує етапи дослідження, ґрунтуючись на поставлених питаннях, та реалізує хід дослідження. Водночас, якщо студент має обмежені навички самостійної

роботи, комп'ютер надає необхідну підтримку. В такому випадку йдеться про спільне управління: студент самостійно освоює навчальний матеріал, але за потреби може отримати конкретні вказівки, контекстні поради або рекомендації від системи підтримки, яка представлена комп'ютерним програмним засобом або викладачем.

1.6 Системи контролю знань в умовах болонського процесу

Ключовим етапом впровадження Болонської конвенції є створення системи контролю знань, при чому одним із методів є тестування.

Участь в Болонському процесі має на меті збільшення мобільності студентів і викладачів, надаючи можливість змінювати країну і університет під час навчання для розширення своїх знань. Українська вища освіта користується перевагою фундаментальності освітніх програм. Щоб використовувати цю перевагу та залучати студентів з інших країн, необхідно приєднатися до побудови загальноєвропейської системи. Для визнання наших дипломів ми повинні перейти на дворівневу систему освіти (бакалаврат і магістрат), відмовитися від системи заліків годин та впровадити єдину систему залікових одиниць або кредитів, аналогічну європейській практиці. Це також передбачає введення системи об'єктивної оцінки знань учнів, що стає можливим завдяки розвитку тестових технологій і впровадженню обов'язкового тестування.

Основні мети Болонського процесу за період з 1998/99 по 2023 рік включають:

I. Утворення системи чітких та порівняльних кваліфікацій, зокрема через впровадження додатків до дипломів (Diploma Supplement), що призначені для:

- підвищення рівня зайнятості європейських громадян;
- підвищення міжнародної конкурентоспроможності європейської освіти.

II. Реформування систем вищої освіти через:

- перехід до дворівневої системи підготовки, що включає бакалавр і магістр;
- умову успішного завершення першого циклу для доступу до другого (протягом не менше 3 років).

III. Впровадження системи кредитних одиниць, аналогічної європейській системі залікового перекладу ECTS, для підтримки широкомасштабної мобільності студентів.

IV. Подолання перешкод для академічної мобільності:

- для студентів;
- для викладачів;
- для наукових співробітників і адміністративного персоналу з метою участі в загальноєвропейських дослідженнях.

V. Забезпечення європейської співпраці в гарантуванні якості вищої освіти.

Незважаючи на досягнення освіти в рамках нової соціополітичної системи України, важливо визнати, що вона поки не забезпечує необхідної якості. Значна частина випускників вищих навчальних закладів не досягає відповідного рівня конкурентоздатності на європейському ринку праці. Це вимагає глибшого аналізу тенденцій у європейській та світовій освіті.

Необхідність трансформації освітньої системи України, її удосконалення та підвищення якості стали визначальною соціокультурною проблемою, що в значній мірі обумовлюється впливом процесів глобалізації та потребою створення благоприятних умов для індивідуального розвитку, соціалізації та самореалізації особистості у сучасному світі.

Ці процеси передбачають необхідність перегляду, гармонізації та визначення нормативно-правового забезпечення сфери освіти, враховуючи стандарти та сертифікаційні вимоги міжнародних та європейських систем.

Україна, аналогічно розвиненим країнам, визнає вищу освіту однією з ключових галузей суспільного розвитку. Стратегічні напрямки її розвитку визначені в Конституції, Законах "Про освіту", "Про вищу освіту", Національній доктрині розвитку освіти, а також у вказівках Президента та рішеннях Кабінету Міністрів.

Основною метою державної політики у сфері освіти є створення умов для особистісного розвитку та творчої самореалізації громадян України, оновлення змісту освіти та організації навчально-виховного процесу відповідно до демократичних цінностей, принципів ринкової економіки та досягнень сучасної науки і техніки.

Основними напрямками державної стратегії стосовно розвитку вищої освіти є:

- Зорієнтованість на особистісний розвиток у вищій освіті.
- Формування національних і загальнолюдських цінностей через освітній процес.
- Забезпечення рівних можливостей для громадян у доступі до вищої освіти.
- Постійне підвищення якості освіти, її оновлення у відповідності до потреб сучасного суспільства.
- Впровадження освітніх інновацій та інформаційних технологій.
- Створення ефективних механізмів залучення та використання позабюджетних коштів у системі освіти.
- Підвищення соціального статусу та професіоналізму працівників освіти, забезпечення їхньої державної і суспільної підтримки.
- Розвиток освіти як відкритої державно-суспільної системи.
- Інтеграція вітчизняної вищої освіти в європейський та світовий освітні простори.

Сучасні тенденції у загальній цивілізації суттєво впливають на реформування системи освіти в Україні, що включає:

- Перехід до динамічної ступеневої системи підготовки фахівців, яка дозволить задовольняти потреби особистості в отриманні освіти та кваліфікаційного рівня за бажаною спеціалізацією та забезпечить мобільність на ринку праці.

- Створення мережі вищих навчальних закладів, що відповідає інтересам та потребам кожної людини і держави за різними критеріями, такими як програми, терміни навчання та джерела фінансування.

- Підвищення рівня освіти та культурного рівня суспільства, створення умов для навчання протягом всього життя.

- Підвищення рівня вищої освіти України до міжнародних стандартів та її інтеграція у світове науково-освітнє співтовариство.

Стратегічні завдання перетворення системи вищої освіти в Україні полягають у переході від кількісних показників освітніх послуг до їх якісного покращення. Цей процес трансформації має ґрунтуватися на таких принципах:

- Збереження і зміцнення національних освітніх традицій, де вища освіта визначається як національна ідея, спрямована на формування громадянина України і гармонійно розвиненої особистості. Це передбачає, що потреба у фундаментальних знаннях та підвищенні загальноосвітнього і професійного рівня пов'язана із зміцненням держави.

- Підпорядкування розвитку вищої освіти законам ринкової економіки, таким як закон розподілу праці, закон змінності праці та закон конкуренції. Врахування цих законів є важливим для формування логіки суспільного розвитку через розвиток економічної сфери.

- Урахування важливих соціальних, політичних, духовних, культурних та морально-психологічних чинників, оскільки ці аспекти грають важливу роль у суспільному розвитку. Багато існуючих проблем у системі вищої освіти пов'язані з нерівноважністю цього комплексу суспільних факторів.

- Розвиток вищої освіти повинен розглядатися в контексті тенденцій у світових освітніх системах, включаючи європейські. Зокрема, необхідно

привести законодавчу та нормативно-правову базу вищої освіти України у відповідність із світовими стандартами, систематизувати структуру вищої освіти та її компоненти, розглянути та оновити перелік спеціальностей, переглянути зміст освіти; забезпечити інтеграцію інформаційних технологій у навчальний процес та забезпечити доступ до міжнародних інформаційних систем. Вищій освіті важливо не лише орієнтуватися на попит на ринку праці, але й збагачувати освітній зміст новітніми матеріалами, впроваджувати передові технології навчання з високим рівнем комп'ютеризації, розвивати творчі та бізнес-зв'язки з потенційними роботодавцями.

1.7 Висновок до розділу 1

Комп'ютеризація освіти створює передумови для впровадження інформаційних технологій у навчання, заснованих на використанні різноманітних програмних засобів педагогічного призначення. Наявність цих програмних засобів дозволяє:

- Збільшити обсяг та продуктивність самостійної роботи в навчальному процесі, підвищити вимоги до рівня її результатів за рахунок використання програм для самонавчання, самоконтролю та самокорекції, а також для підтримки предметної та професійної діяльності, обробки та представлення результатів роботи;
- Збільшити мотивацію студентів до самостійної роботи, наблизивши її до реальних умов майбутньої професійної діяльності;
- Отримати гнучкість та персоніфікацію управління самостійною роботою студентів, базовану на нагромадженні та аналізі статистичних даних про хід і результативність навчальної діяльності кожного студента;
- Поліпшити управлінську діяльність викладача за рахунок автоматизації інформаційно-інструктивної, реєстраційно-облікової та контрольно-оцінювальної складових.

РОЗДІЛ II

ОПИС ІСНУЮЧОЇ СИСТЕМИ КОМП'ЮТЕРНОГО ТЕСТУВАННЯ ЗНАНЬ СТУДЕНТІВ

2.1 Огляд програми для створення тестів

Традиційно використовується паперове (бланкове) тестування. Проте, недавно були розроблені різноманітні програми для тестування, такі як системи електронного тестування (СЕТ), що дозволяють проводити тестування за допомогою комп'ютера. Можливості цих систем дозволяють їх використовувати як на заняттях, так і в поза заняттями. Зазвичай сучасні системи складаються з декількох функціональних модулів, які можуть бути об'єднані в єдине ціле й інсталюватись на комп'ютери або існувати окремо у вигляді виконавчих файлів. Стандартна СЕТ, як правило, включає в себе такі компоненти:

- Редактор тестів (модуль для створення тестів);
- Модуль тестування;
- Модуль для обробки результатів тестування;
- Довідкова система;
- Модуль для мережевого тестування.

Система електронного тестування повинна мати можливість створювати запитання та відповіді, які можуть включати, крім тексту, також формули та рисунки. В іншому випадку ефективність використання такої системи буде дуже низькою, і її можна буде використовувати тільки для перевірки знань, означень та формулювань теорем, понять і т.д. Бажано, щоб СЕТ підтримувала якнайбільше різновидів завдань.

Однією з ключових характеристик таких систем, враховуючи варіативний рівень комп'ютерної грамотності як серед учнів, так і серед

вчителів різних предметів, є збалансованість між простотою використання та функціональними можливостями MyTest.

MyTest - це система для підготовки та проведення тестування з метою визначення рівня навчальних досягнень студентів. Вона включає в себе три основні модулі:

- Редактор тестів (MyTestEditor);
- Модуль тестування (MyTestStudent);
- Журнал тестів (MyTestServer).

Редактор тестів використовується для створення тестів. MyTestEditor обладнаний вбудованим текстовим редактором, який дозволяє створювати запитання та відповіді будь-якої складності. Також можна використовувати інші текстові редактори (наприклад, Microsoft Office Word), а потім шляхом копіювання вставляти фрагменти запитань та відповідей до редактора тестів. Таким чином, якщо існує електронний варіант тесту, створений за допомогою текстового редактора, його можна легко перетворити у тест за допомогою системи MyTest.

Програма MyTest взаємодіє з дев'ятьма видами завдань:

- окремий вибір;
- множинний вибір;
- встановлення порядку слідування;
- встановлення відповідності;
- визначення істинності чи хибності тверджень;
- ручне введення числа;
- ручне введення тексту;
- вибір місця на зображенні;
- перестановка букв.

Створений тест зберігається в єдиному стиснутому та зашифрованому файлі, який включає завдання, параметри тестування та зображення, пов'язані

з кожним конкретним завданням. Це також є перевагою цієї системи тестування.

При наявності комп'ютерної мережі можна організувати централізований збір і обробку результатів тестування через модуль журналу MyTest. Результати завдань виводяться студентові та надсилаються викладачеві для подальшої оцінки або аналізу. Також можливо розповсюдження тестів студентам через мережу, що усуває потребу в постійному копіюванні файлів тестів на всі комп'ютери. Загалом, програма MyTest легка у вивченні та зручна у використанні, маючи простий інтуїтивно зрозумілий інтерфейс і доступну довідкову систему. Для роботи з MyTest вистачає базових навичок роботи з комп'ютером.

KTCNet 3. Програмний комплекс KTCNet призначений для оцінки знань у будь-якій сфері та проведення психологічного тестування з використанням комп'ютера. Головною особливістю цього продукту є його орієнтація на розподілену мережеву взаємодію між користувачами та адміністраторами системи. Такий підхід дозволяє ефективно управляти системою та отримувати інформацію про результати оперативно.

Основні можливості продукту включають:

- можливість форматування тексту питань і відповідей у практично будь-якому стилі (з вставкою таблиць, формул, верхніх і нижніх індексів, графіків, списків, різних шрифтів тощо);
- створення тестів з будь-якою кількістю питань і відповідей до кожного;
- можливість прикріплення рисунків, відео, OLE-графіки та звукового супроводу до питань і відповідей;
- підтримка п'яти видів питань: вибір 1-го варіанту, вибір декількох варіантів, розташування відповідей по порядку, введення з клавіатури, розставлення відповідностей;

- можливість створення самостійних exe-файлів, які можуть працювати на інших комп'ютерах без участі KTCNet;
- простий користувацький інтерфейс та інші зручності.

Test-W2. Цей програмний продукт розроблено для моніторингу знань учнів, перевірки відповідності їх знань вимогам навчальних програм та визначення рівня навчальних досягнень з будь-якого предмета за допомогою комп'ютера.

Використання цієї системи є досить простим, що становить одну з її основних переваг. На жаль, варто відзначити, що вона обмежена підтримкою лише одного типу завдань, що є значущим недоліком.

Testing. Програмний продукт розроблено для проведення тестування на локальних комп'ютерах або у мережі. Комплекс програм для тестування "Testing" володіє значущим функціональним потенціалом. У порівнянні з раніше розглянутою системою CET, система "Testing" вважається трошки складнішою у використанні.

Цей комплекс програм для тестування складається з шести програм:

- CreateBQ.exe – для створення бази запитань;
- CreateTest.exe – для створення тесту;
- Tsetting.exe – для налаштування параметрів тестера;
- Tester.exe – процес тестування;
- TServer.exe – для тестування в локальній мережі;
- AnalyzRes.exe – для аналізу результатів.

Можливості цього комплексу програм для тестування включають:

- підтримку різних типів запитань: вибіркового (одинарний), вибіркового (множинний), встановлення відповідності (встановлення послідовності), введення відповіді;
- визначення кількості варіантів відповідей до 5;

- призначення ваги (складності) питання від 1 до 4. Залежно від визначеної ваги, за кожне питання на тестуванні буде нараховуватись різна кількість балів, а також буде відведено різний час на його відповідь;
- обмеження за часом може бути встановлене для всього тесту, дозволяючи визначити кількість хвилин для його проходження. Крім того, для окремих питань можна встановити обмеження в секундах. Також передбачена можливість збільшення часу на відповідь в залежності від складності питання, при цьому вибране обмеження служить основою для питань вагою 1;
- налаштування включають можливість конфігурування шкали оцінювання. Система передбачає дві стандартні шкали результатів: п'ятибальну та ECTS (європейську кредитно-модульну систему), а також можливість створення іншої, наприклад, дванадцятибальної;
- додаткові параметри включають в себе вибір загальної кількості запитань з бази, налаштування порядку виведення запитань та варіантів відповідей (однаковий для всіх чи різний).
- система також дозволяє проводити тестування по локальній мережі, а також виконувати аналіз відповідей та інші функції.

2.2 Опис програмного забезпечення MyTestX

Для розгляду слід обрати програму, яка відповідає всім стандартам тестування. Програма MyTest розробляється з 2003 року, і за цей період вийшло безліч різних версій. Кожна нова версія удосконалювала попередню та вводила нові можливості. Починаючи з простих, але зручних тестових оболонок, сучасна версія MyTestX не лише програма, а й потужний комплекс для підготовки та проведення комп'ютерного тестування.

MyTestX - це система програм, яка включає програму для тестування студентів, редактор тестів та журнал результатів. Вона призначена для створення та проведення комп'ютерного тестування, а також для збору та

аналізу результатів, виставлення оцінок за визначеною шкалою, вказаною в тесті.

Програма MyTestX є легкою та зручною у використанні. Вона працює з десятима типами завдань: одиночний вибір, множинний вибір, встановлення порядку проходження, встановлення відповідності, вказівка істинності чи хибності тверджень, ручне введення числа, ручне введення тексту, вибір місця на зображенні, перестановка букв та заповнення пропусків (MyTestXPro). У тесті можна використовувати будь-яку кількість та будь-які типи завдань, від одного до всіх одночасно. Зокрема, у завданнях із вибором відповіді можна використовувати до 10 (включно) варіантів відповіді.

Програма складається з трьох основних модулів: Модуль тестування (MyTestStudent), Редактор тестів (MyTestEditor) і Журнал тестування (MyTestServer).

Для створення тестів доступний дуже зручний редактор з користувацьким та привітним інтерфейсом. Навіть викладачі, які мають обмежений досвід роботи з комп'ютерами, можуть легко створювати свої власні тести для програми MyTestX та використовувати їх на своїх заняттях.

У програмі існують широкі можливості форматування тексту питань та варіантів відповіді. Ви маєте змогу налаштовувати шрифт, колір символів та фону, використовувати верхні та нижні індекси, розбивати текст на абзаци та застосовувати до них розширене форматування, використовувати списки, вставляти малюнки та формули і т.д. Для спрощення використання доступний власний текстовий редактор.

До кожного завдання можна вказати його складність (кількість балів за вірну відповідь), додати підказку (за яку може бути накладено штраф у балах) та надати пояснення правильної відповіді (яке виводиться в разі помилки у режимі навчання), а також налаштувати інші параметри.

Є можливість використовувати різні форми питань у завданнях, зручно створювати вибірку завдань для студентів, перемішувати завдання та варіанти

Програма включає кілька незалежних режимів: навчальний, штрафний, вільний і монопольний. У навчальному режимі студент отримує повідомлення про свої помилки, а також може отримати пояснення до завдання. У штрафному режимі за невірні відповіді віднімаються бали, і студент може пропустити завдання (бали не надаються і не віднімаються). У вільному режимі студент може відповідати на питання у будь-якому порядку, переходити (повертатися) до будь-якого питання самостійно. У монопольному режимі вікно програми займає весь екран і його неможливо згорнути.

При правильному відборі матеріалу для контролю, зміст тесту можна використовувати не лише для оцінки, але й для самостійного навчання. Це дозволяє студентів виявляти прогалини у своїх знаннях і вживати заходів для їх заповнення. У таких випадках можна говорити про значний навчальний потенціал тестових завдань, використання якого стане ефективним напрямком практичної реалізації принципу єдності та взаємозв'язку навчання і контролю.

Кожен тест має оптимальний час для проходження, і зменшення або перевищення його може погіршити якість тесту. Тому в налаштуваннях тесту передбачено обмеження часу як для всього тесту, так і для відповіді на конкретне завдання (для різних завдань можна встановити різний час).

Оберемо описану програму для детального розгляду та розглянемо створення об'єкта для тестування, щоб вивчити пропонований набір програм.

2.3 Проєктування моделі тестування

Як було вказано раніше, цей комплект включає три програми: MyTestServer, MyTestEditor і MyTestStudent. Далі розглянемо застосування журналу тестування в MyTestXPro.

Припустимо, що комп'ютерне тестування проводиться на кількох комп'ютерах. На кожен з них необхідно скопіювати тест, а потім зібрати результати з кожного. Якщо комп'ютери підключені до мережі, то хоча загальні каталоги полегшують завдання, але не на повністю. Для більш

ефективного вирішення цієї задачі можна скористатися Журналом тестування (MyTestServer).

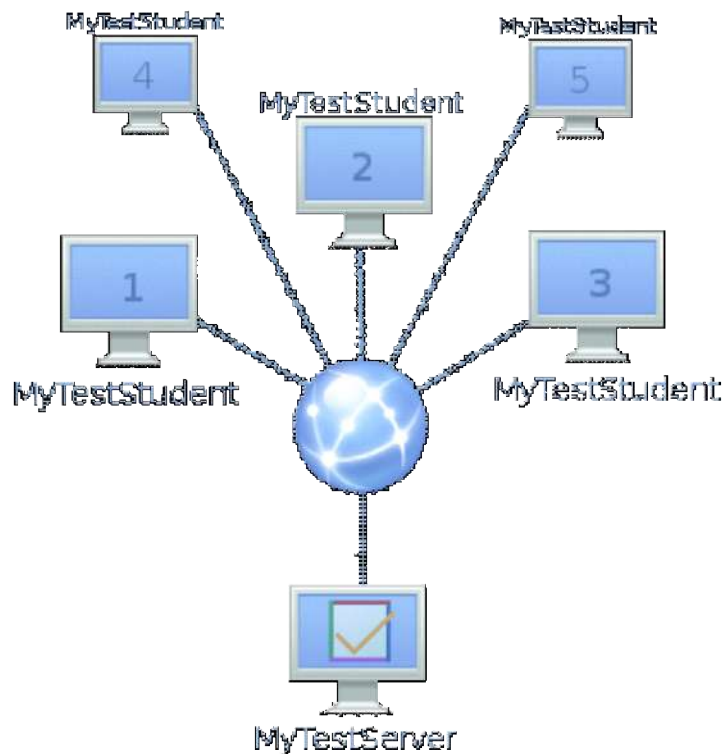


Рис. 2.2 – Схема раціонального налаштування програм в мережі

Достатньо встановити MyTestXPro на комп'ютерах студентів у мінімальному варіанті, включаючи лише модуль тестування (MyTestStudent). Однак на одному з комп'ютерів потрібно встановити повний комплект, включаючи Журнал тестування (MyTestServer). Ймовірно, це буде комп'ютер викладача, але це не обов'язково — може бути будь-який з комп'ютерів. (рис. 2.2).

2.4 Налаштування модуля тестування

Після запуску програми (модуль тестування) слід відкрити вікно налаштувань (розділ "Налаштування" - "Параметри") і перейти на вкладку "Мережа". За замовчуванням мережеві можливості вимкнуті. (рис. 2.3).

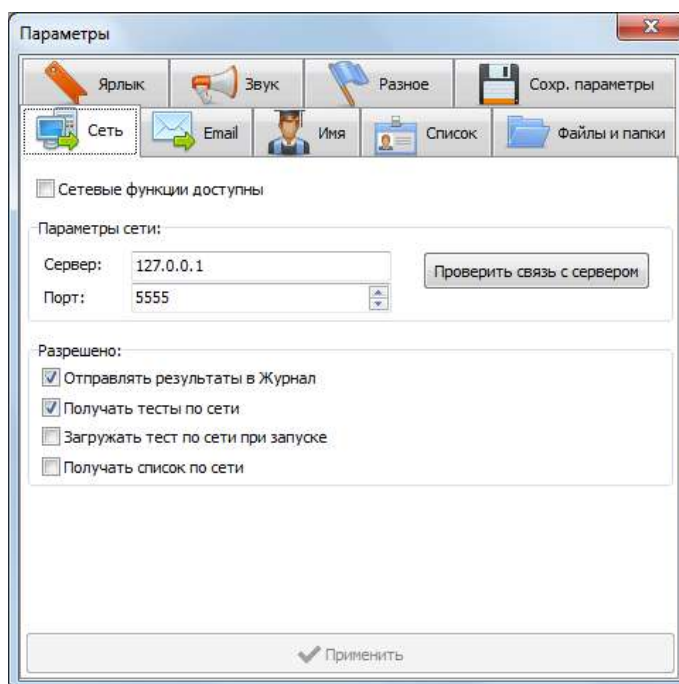


Рис. 2.3 – Вікно параметрів

Щоб студент отримував тести з мережі та Журнал тестування відправляв результати викладачеві, слід налаштувати модуль тестування. У реальних умовах, в поле "Сервер" необхідно вказати ім'я або IP-адресу комп'ютера, на якому працюватиме Журнал тестування.

Початково для роботи з програмою не потрібно виконувати жодних додаткових налаштувань Журналу тестування; можливо, пізніше буде необхідно налаштувати різні опції. Якщо брандмауер операційної системи видає запит щодо дозволу чи блокування мережевих можливостей програми, слід надати дозвіл, інакше операційна система буде їх блокувати.

Після встановлення та налаштування програми можна випробувати її мережеві можливості.

2.5 Роздача тестів

При користуванні Журналом тестування не потрібно копіювати файли тестів на кожен комп'ютер або надавати до них доступ через загальний

каталог. Таким чином, ми не лише спрощуємо свою роботу, але й додатково захищаємо тести від несанкціонованого доступу.

Розглянемо, як роздати тести. Запустіть Журнал тестування, перейдіть на вкладку "Роздати тест". У таблиці відображаються тести, доступні для роздачі, з вказаними обмеженнями по даті, часу або IP-адресі. Якщо ви ще не роздавали тести, то таблиця буде порожня. (рис. 2.4).

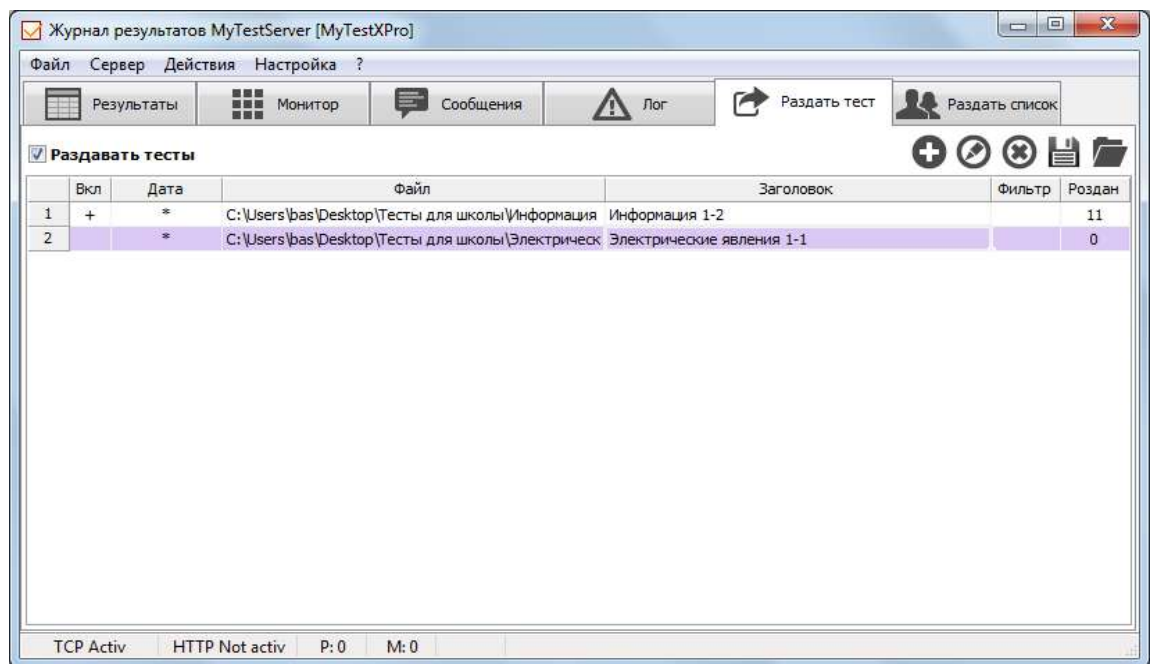


Рис. 2.4 – Журнал результатів

Тепер студенти можуть отримати необхідні тести з мережі, натиснувши кнопку "Отримати тест" (рис. 2.5) в Журналі. Після запуску програми, яка була налаштована, як описано вище, студент може вибрати "Отримати по мережі" в меню Файл або використовувати кнопку "Отримати по мережі". Модуль тестування встановлює зв'язок з Журналом, отримує тест і автоматично відкриває його. Файл тесту не зберігається на комп'ютері студента. Після цього студенти проходять тест так, наче він відкритий з файлу.

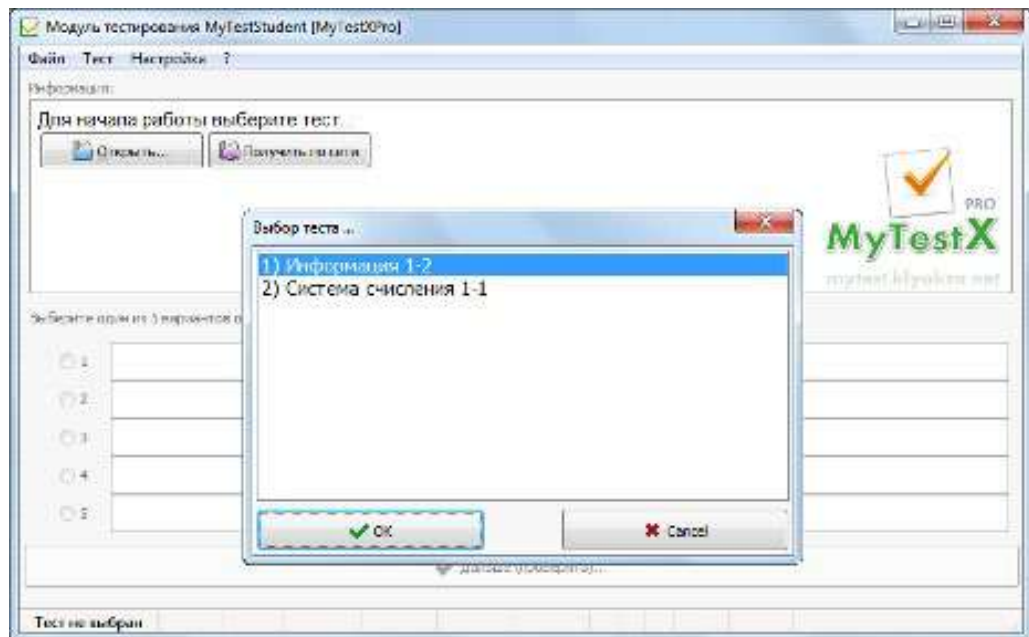


Рис. 2.5 – Початкова сторінка MyTestStudent

2.6 Централізоване отримання результатів

MyTestXPro надає викладачеві (вчителю, інструктору) не лише загальну оцінку за тест, але й деталізовану інформацію про проходження тестування. Зокрема, Журнал тестування здатний автоматично збирати результати тестування (рис. 2.6).

Для того щоб Журнал тестування міг отримувати результати, необхідно:

- Налаштувати програму, як описано вище;
- Увімкнути цю можливість в параметрах тесту.

Таким чином, викладач може контролювати, чи можуть результати тестування надсилатися в Журнал тестування для подальшого аналізу. Для цього потрібно відкрити в Редакторі тестів потрібний тест і переконатися, що встановлені параметри "Відправляти результати по мережі в журнал" і "Відображати результат на моніторі" (рис. 2.7).

Вкладка "Монітор" аналогічна вкладці "Результати", але вона відображає не кінцеві результати тестування, а поточні. Тут можна

переглядати інформацію про те, який тест виконує кожен студент, скільки завдань він вже виконав і наскільки успішно.

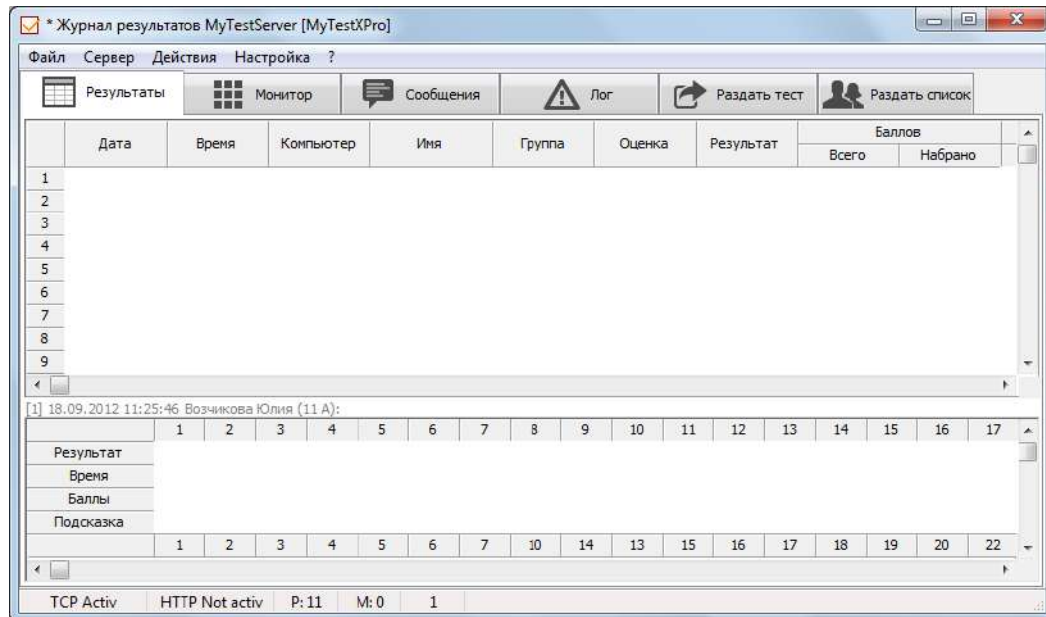


Рис. 2.6 – Журнал результатів в MyTestServer

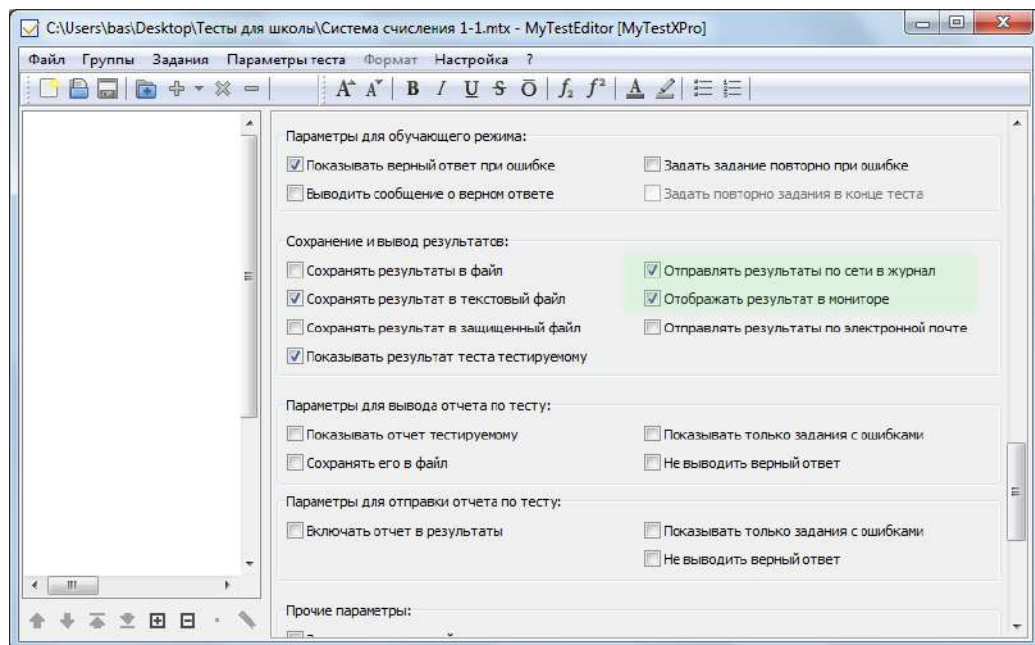


Рис. 2.7 – Налаштування журналу тестування

Після завершення тестування запис для даного студента зникає з монітора, і в таблиці результатів з'являється інформація про кінцеві результати тестування.

Записи, які давно не оновлювалися, що може вказувати на перерваний зв'язок по мережі або примусове закриття програми, позначаються відповідним кольором. Перенесення записів з монітора до результатів має сенс, особливо для таких "завислих" записів.

Під час отримання результатів може бути відтворений звуковий сигнал, і користувач може побачити спливаюче повідомлення. Отримані результати можна зберегти і/або проаналізувати.

2.7 Збереження результатів

Зібрані результати тестування можна зберегти у файл для подальшого аналізу у вигідний час.

Щоб зберегти результати, виберіть опцію Файл → Зберегти... (або Зберегти як...), вкажіть ім'я файлу і каталог для збереження.

Збережені результати можна відкрити пізніше за допомогою опції Файл → Відкрити.... Також є можливість об'єднати результати в один файл або імпортувати їх з інших файлів (Файл → Імпорт).

В Журналі тестування існує можливість автоматичного збереження прийнятих результатів. Для більшості користувачів можна залишити тільки одну опцію в автоматичному збереженні – Зберігати результати в текстовий файл. Ця операція виконується швидко і дозволяє відновити оцінки (результати) тестування у випадку переривання чи відключення комп'ютера. Результати також можна зберігати вручну (Файл → Зберегти...) за потреби.

2.8 Створення відгуку

Якщо ви хочете переглянути прикріплений до результату звіт, скористайтеся командою Дії → Результати → Звіти → Показати звіт або натисніть клавішу F6.

Не обов'язково включати звіт до результатів тестування, його можна створити, маючи файл з тестом, для якого вже отримані результати.

Давайте розглянемо приклад, введемо необхідні параметри для одного учня, наприклад, Володимира Чижа, учня 11А класу, який пройшов тест на тему «Системи числення».

У звіті буде вказано його результат, час виконання, отримана оцінка і його особисті дані, які будуть взяті з інформації, яку ми внесли при складанні тесту.

Для створення звіту виберіть Дії → Результати → Звіти → Створити звіт у вікні в меню. Необхідно вказати ім'я файлу тесту та параметри звіту (показувати правильні відповіді, показувати всі або тільки завдання з помилками). Якщо утримувати клавішу Shift при виборі команди, звіт буде не тільки створений і показаний, а й прикріплений до результату. (рис. 2.8).

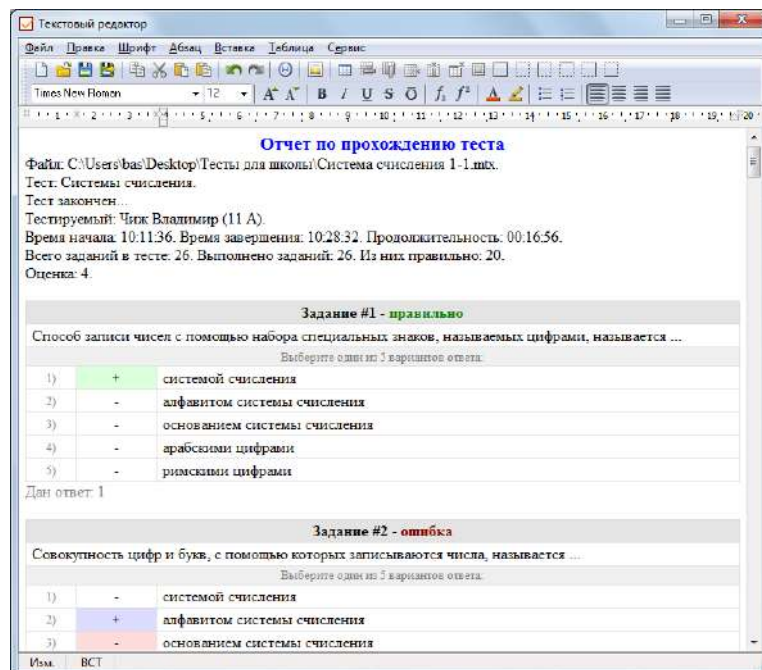


Рис. 2.8 – Створення звіту

Таким чином, було розглянуто процес створення програми для тестування в рамках пакету програм MyTestXPro.

2.9 Висновок до розділу 2

У розділі детально розглянуто методику використання автоматизованої системи тестування студентів MyTestXPro. Також проведено аналіз історії тестування, яка визначає контроль над знаннями студентів, і встановлено, що з цієї точки зору тестування є ефективним інструментом для перевірки рівня підготовки та оцінки кваліфікації особи. Важливо відзначити, що проведення тестування сприяє розвитку у людини або групи людей здатності приймати обґрунтовані рішення в різних ситуаціях.

Під час аналізу програми тестування MyTestXPro та вирішення поставленої задачі, виявлено певний недолік у цьому додатку — його обмежена локалізація лише для платформи Windows. Це обмеження ставить певні обмеження на користування програмою, особливо з урахуванням росту популярності онлайн систем. Особливо важливо розглядати можливість розробки онлайн системи екзаменування, яка дозволить користувачам заходити в систему з будь-якого пристрою з встановленим браузером..

РОЗДІЛ III

РОЗРОБКА ВЕБ СИСТЕМИ ЕКЗАМЕНУВАННЯ

3.1 Загальний опис та аналіз архітектури

Варто відзначити, що на сучасний момент тестування є одним з найпоширеніших методів оцінки не лише осіб, а й інших живих організмів, а також використовується для перевірки функціональності різних класів пристроїв, таких як електронні обчислювальні машини і інші.

Аналізуючи поставлену задачу та методи її вирішення, було вирішено розробляти програмний комплекс на основі веб-технологій. Основною перевагою веб-застосунку перед іншими альтернативами є його універсальність і можливість використання на будь-яких пристроях без необхідності портування під конкретну операційну систему (браузер і його віртуальна машина діють як цільова універсальна операційна система та комп'ютер).

Сучасні веб-додатки використовують клієнт-серверну архітектуру, яка стала одним із основних методів організації розподілених програмних систем протягом останнього десятиліття. За такої організації, клієнтською частиною є веб-браузер (Mozilla Firefox, Google Chrome, Opera, Internet Explorer, Apple Safari і інші), а серверною — веб-сервер (Apache HTTP Server, Microsoft Internet Information Services і інші).

Web-браузер має завдання відображення веб-документів (сторінок HTML) та вбудованих мультимедійних об'єктів, що їх створюють. У свою чергу, Web-сервер функціонує як обробник запитів від віддалених веб-клієнтів. Ця обробка може включати в себе відправку статичної веб-сторінки безпосередньо з HTML-файлу, генерацію динамічного вмісту за допомогою спеціалізованих або універсальних мов програмування (таких як PHP, Python, Java та інші), передачу даних від користувача на веб-сервер, включаючи

файли, а також оновлення конкретного фрагмента вже відображеної веб-сторінки (використовуючи так звану технологію Ajax).

У подібних випадках для обміну інформацією використовується протокол передачі гіпертексту (HTTP - HyperText Transfer Protocol). Під гіпертекстом мають на увазі документи в мережі Інтернет, які пов'язані за допомогою спеціальної адресації (URL). Основні характеристики протоколу HTTP включають наступне.

1. Обмін даними здійснюється у форматі "запит-відповідь", тобто кожна транзакцію обміну даними ініціює веб-клієнт, а веб-сервер генерує відповідь.
2. Протокол HTTP відноситься до верхнього (прикладного) рівня моделі відкритих систем OSI, і тому його синтаксис найбільш наближений до природної мови.
3. HTTP не зберігає історії попередніх транзакцій, тому збереження стану реалізується іншими засобами, зокрема, на основі змінних cookie.
4. Кожен ресурс, доступ до якого здійснюється за допомогою HTTP, ідентифікується через URL (узагальнений локатор ресурсів) - загальноприйнятий метод адресації в Інтернеті.
5. Запит HTTP складається із рядка запиту ("метод" "адреса ресурсу" "версія HTTP"), заголовків клієнта ("назва": "значення") і, в опціональному вигляді, тіла запиту.
6. Відповідь HTTP включає статусний рядок ("версія HTTP", "код статусу", "опис коду статусу"), заголовки сервера ("назва": "значення") і, опціонально, тіло відповіді. Обмін інформацією здійснюється за допомогою методів HTTP: GET, POST, PUT, HEAD, TRACE, OPTIONS, CONNECT, DELETE. Заголовки HTTP уточнюють запит або відповідь і виконують службову роль при інтерпретації пакетів. Заголовки HTTP класифікуються так:

- Заголовки клієнта (конфігурація клієнта, серверна адреса, тип, кодування, мова вмісту ресурсу, параметри кешування тощо);
- Заголовки сервера (конфігурація сервера, перенаправлення, встановлення Cookie, параметри аутентифікації тощо);
- Заголовки вмісту повідомлення (MIME-тип повідомлення, його розмір, параметри мови, кодування, ущільнення повідомлення тощо);
- Загальні заголовки (дата відправлення/отримання повідомлення, сторінка, з якої виконувався перехід, управління кешуванням та з'єднанням тощо).

Протокол HTTP, з своєї природи, не передбачає збереження попередніх станів обміну повідомленнями між клієнтом і сервером. Таким чином, для цього протоколу було введено два спеціальні заголовки (Set-Cookie та Cookie). Вони дозволяють серверу встановлювати («зберігати») на клієнті невеликі об'єми даних (до 4096 байтів), які клієнт повертає з кожним запитом. Це дозволяє серверу однозначно та унікально ідентифікувати кожного клієнта. Фізично ці дані зберігаються у спеціальному конфігураційному файлі веб-браузера, а сервер встановлює час актуальності цих даних за допомогою спеціального синтаксису заголовка Set-cookie.

Збереження стану є характерною рисою для різних веб-додатків, таких як клієнти електронної пошти, інтернет-магазини, інформаційні портали і т. д. Один із важливих аспектів протоколу HTTP для розробників веб-додатків - це коди статусу обробки запиту клієнта (наприклад, 200 OK, 301 Moved Permanently, 403 Forbidden, 404 Not Found і т. д.).

Отже, за допомогою унікальних властивостей протоколу HTTP, таких як правила адресації, формат пакету, заголовки, методи та коди статусу, втілюються різноманітні методи взаємодії між веб-клієнтом та веб-сервером за допомогою спеціалізованих мов програмування та розмітки (PHP, Python, HTML, XML і т. д.).

3.2 Мова програмування PHP

Мова програмування PHP представляє собою скриптову мову, призначену для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найбільш поширених мов, які використовуються у сфері веб-розробок, разом з Java, .NET, Perl, Python та Ruby. Більшість хостинг-провайдерів підтримують мову PHP, і вона є проєктом відкритого програмного забезпечення.

PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. У відміну від скриптової мови JavaScript, PHP-код не відображається користувачеві, оскільки браузер отримує готовий HTML-код. Це сприяє безпеці, але може погіршити інтерактивність сторінок [13].

Мова PHP має вбудовані бібліотеки для роботи з різними системами управління базами даних, такими як MySQL, PostgreSQL, mSQL, Oracle, dbm, Hyperware, Informix, InterBase, Sybase. Благодаря стандартному інтерфейсу для зв'язку з базами даних, можна здійснювати підключення до будь-яких баз даних, які підтримуються відповідними драйверами.

Популярність PHP у сфері розробки веб-сайтів обумовлена наявністю обширного спектру вбудованих інструментів для створення веб-додатків [14]. До основних з них належать:

- автоматичне конвертування POST і GET-параметрів, а також змінних оточення веб-сервера в масиви для подальшої обробки;
- взаємодія з різноманітними системами управління базами даних, такими як MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape і Apache Derby, Informix, Ovrmos SQL, Lotus Notes, DB++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird/InterBase, Paradox File Access, MaxDB, інтерфейс PDO;
- автоматизована генерація і відправка HTTP-заголовків;
- обробка HTTP-авторизації;

- робота з cookies і сесіями;
- опрацювання локальних і віддалених файлів, а також взаємодія з сокетами;
- обробка завантажених на сервер файлів;
- використання XForms для роботи з веб-формами.

Механізм PHP розпочинає виконання коду відразу після знаходження першої екрануючої послідовності (<?) і продовжує виконання до того моменту, коли зустрічає парну екрануючу послідовність (?>).

Мова PHP буде знайомою програмістам, що працюють у різних галузях. Багато конструкцій мови є запозиченими з мов програмування C і Perl. Код PHP схожий на той, який можна знайти в типових програмах, написаних мовами C або Pascal. Це значно спрощує початковий етап вивчення PHP. Програмування на мові PHP поєднує переваги Perl та C, і спеціалізовано орієнтоване на веб-розробку, має універсальний і зрозумілий синтаксис [15]. Незважаючи на те, що PHP є досить молодого мовою програмування, вона набула великої популярності серед веб-програмістів і на сьогоднішній день є найбільш використовуваною мовою для створення веб-застосунків та скриптів.

Широкий спектр функцій у PHP уникає необхідності в розробці складних і довгих користувацьких функцій на C або Pascal. Мова PHP володіє універсальним і зрозумілим синтаксисом, спеціально орієнтованим на роботу в Інтернеті. Важливо зазначити, що вона є відкритою для перегляду та безкоштовною. Значущою перевагою є те, що PHP відноситься до інтерпретованих мов, що дозволяє обробляти сценарії з високою швидкістю [16].

Однією з прикладних областей PHP є розробка парсера, або синтаксичного аналізатора, який є програмою або її частиною і виконує синтаксичний аналіз. Задача цього парсера полягає в тому, щоб перейти за вказаними URL-посиланнями, завантажити HTML-сторінку вказаного сайту і

потім провести розбір та обробку отриманих даних відповідно до визначеної задачі [17].

Ефективність виявляється важливим аспектом у програмуванні для оточень, спрямованих на багато користувачів, серед яких входить і веб. Суттєвою перевагою PHP є те, що ця мова відноситься до інтерпретованих. Це дозволяє обробляти сценарії з високою швидкістю. За деякими оцінками, більшість PHP-сценаріїв, особливо невеликих розмірів, обробляються швидше, ніж аналогічні програми, написані на Perl. Зате незалежно від того, що роблять розробники PHP, виконавчі файли, отримані через компіляцію, працюють значно швидше — вдесятки, іноді навіть сотні разів. Проте продуктивність PHP вистачає для створення повноцінних веб-застосунків.

3.3 Опис веб-додатку

Інтернет-система екзаменування, побудована на PHP, JS, HTML та CSS. Були використані бібліотеки JQuery та Bootstrap. У створеному застосуванні усі дані про тести і користувачів зберігаються у базі даних, що дозволило знизити об'єм написаного коду, тим самим полегшити процес створення проєкту.

Для виконання проєктування структури бази даних обрано інструмент phpMyAdmin. База даних складається з 6 таблиць. Кожна з таблиць має первинний ключ - поле таблиці, що однозначно ідентифікують кожен рядок. (рис. 3.1).

База даних складається з наступних таблиць:

- options – таблиця варіантів;
- rank – таблиця рейтингів;
- history – таблиця журналу виконань тестів;
- feedback – таблиця відгуків;
- quiz – таблиця тестів;
- questions – таблиця питань;

- answer – таблиця відповідей;
- user – таблиця користувачів;
- user_answer – таблиця відповідей користувачів;
- admin – таблиця користувачів адміністраторів.

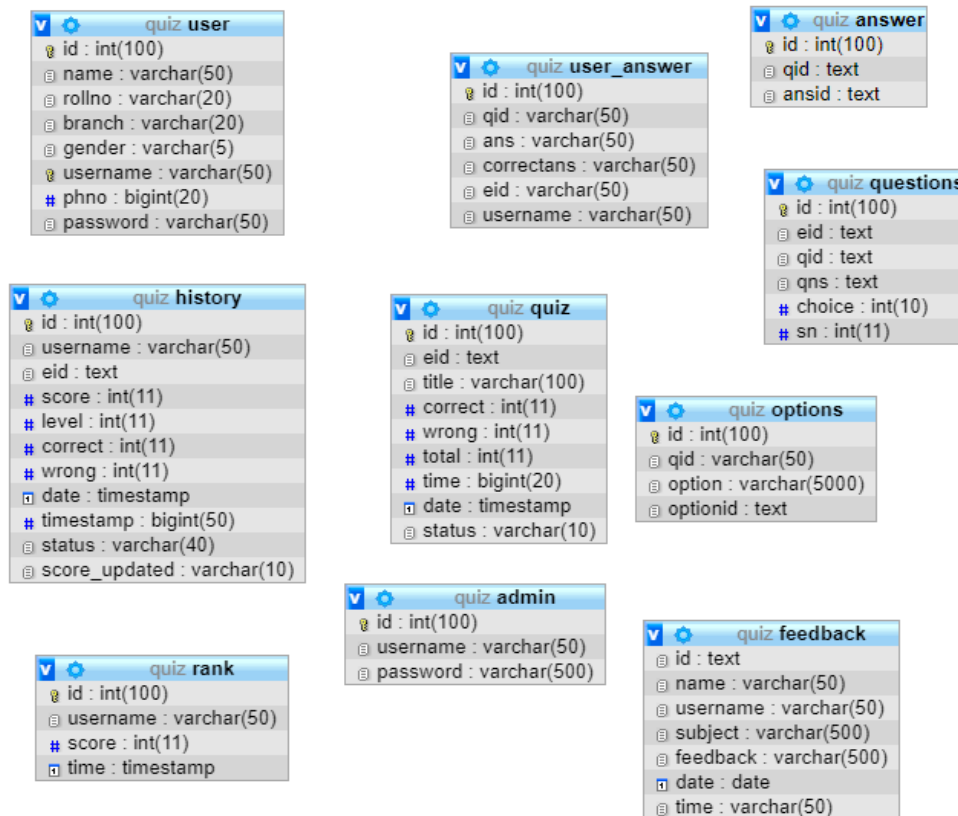


Рис. 3.1 Схема таблиц бази даних

Базовий функціонал системи:

- Підтримка таймера.
- Керування тестами на панелі адміністратора.
- Навігація серед усіх питань тесту (під час тесту) та закінчення тесту, коли користувач захоче.
- Керування процесом виконання тест користувачем в будь-який час і продовження тесту, навіть якщо сталася помилка або якщо було перевищено час очікування сеансу.

- Контроль зберігання відповідей на запитання та показу детального аналізу результатів тестів.
- Фіксація та обробка відгуків
- Покращений графічний інтерфейс панелі тесту.

Панель адміністратора використовується для налаштування тестів. Тест не буде ввімкнений, якщо адміністратор не натисне кнопку "Увімкнути".

Оцінки оновлюються в режимі реального часу на сервері, однак таблиця лідерів оновлюється лише тоді, коли користувач закінчить тест, або вийде час відведення на тест, або адміністратор закінчить тест, натиснувши кнопку «Вимкнути».

Після того, як адміністратор натисне кнопку відключення, тест закінчується для всіх користувачів, які виконують тест, незалежно від їх активного чи неактивного стану (незалежно від того, увійшли чи залишили тест під час виконання). Рейтингова таблиця буде оновлена або тоді, коли користувач "закінчить" свій тест, і коли адміністратор "відключить" тест.

Після вимкнення тест стає недоступною. Якщо тест буде ввімкнений пізніше, його можуть виконати лише ті користувачі, що виконали його.

Рекомендується ввімкнути тест, коли всі користувачі готові, і відключити тест, коли всі користувачі виконали тест.

Рис. 3.2 Головна сторінка з формами реєстрації та авторизації

Головну сторінку користувача зображено на рис. 3.2 та написано згідно стандартів Bootstrap додатків. Це початкова сторінка веб-додатку, що дозволяє зареєструватися, увійти в додаток під правами користувача або адміністратора. Фрагмент коду оновлення головної сторінки користувача веб-системою має вид:

```
if ($status == "enabled") {
    echo '<tr><td style="vertical-
align:middle">' . $c++ . '</td><td style="vertical-
align:middle">' . $title . '</td><td style="vertical-
align:middle">' . $total . '</td><td style="vertical-
align:middle">' . $correct * $total . '</td><td style="vertical-
align:middle">' . $time . '&nbsp;min</td><td style="vertical-
align:middle">Enabled</td>
    <td style="vertical-
align:middle"><b><a href="update.php?deidquiz=' . $eid . '" clas
s="btn logb" style="color:#FFFFFF;background:#ff0000;font-
size:12px;padding:5px;">&nbsp;<span><b>Disable</b></span></a></b>
</td></tr>';
} else {
    echo '<tr><td style="vertical-
align:middle">' . $c++ . '</td><td style="vertical-
align:middle">' . $title . '</td><td style="vertical-
align:middle">' . $total . '</td><td style="vertical-
align:middle">' . $correct * $total . '</td><td style="vertical-
align:middle">' . $time . '&nbsp;min</td><td style="vertical-
align:middle">Disabled</td>
    <td style="vertical-
align:middle"><b><a href="update.php?eeidquiz=' . $eid . '" clas
s="btn logb" style="color:#FFFFFF;background:darkgreen;font-
size:12px;padding:5px;">&nbsp;<span><b>Enable</b></span></a></b>
</td></tr>';
}
```

Головний інтерфейс, має короткий опис, який включає : назву тестів, кількість питань, кількість правильних та неправильних відповідей, загальну

кількість балів, регламент часу , дії та загальну інструкцію проходження тесту (рис. 3.3).

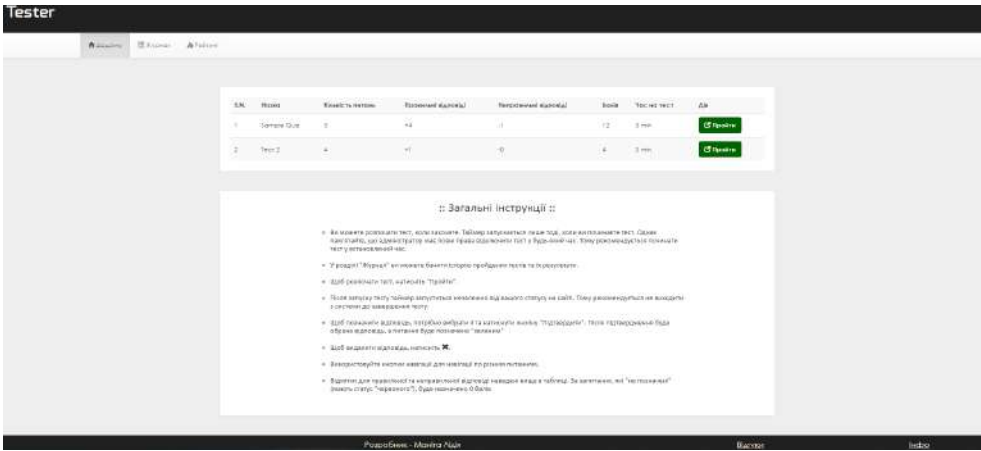


Рис. 3.3 Головний інтерфейс користувача (роль:студент)

Інтерфейс сторінки рейтингу дозволяє оцінити стан проходження тестів відносно інших учасників та додає змагальності у процесі їх виконання (рис. 3.4).



Рис. 3.4 Рейтинг користувачів

Форма для відгуків містить необхідні поля для заповнення реквізитів інформаційного відгуку про проблеми з виконанням тестів та їх належну якість (рис. 3.5).

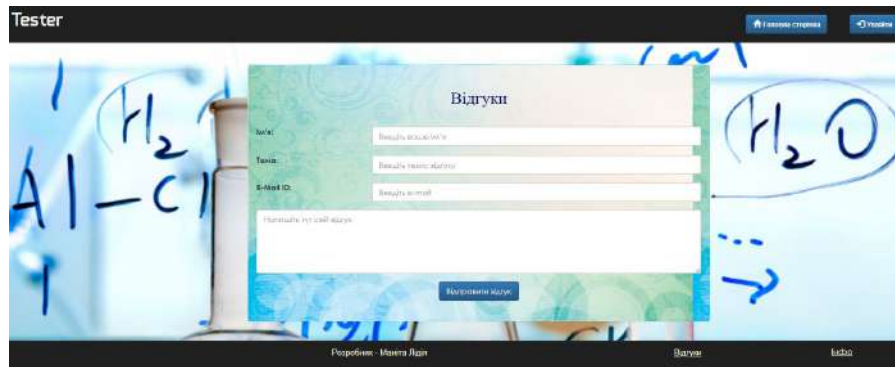


Рис. 3.5 Форма для відгуків

Лістинг обробки для бази даних форми для відгуків наведено нижче:

```
<?php
include_once 'dbConnection.php';
$ref=@$_GET['q'];
$name = $_POST['name'];
$email = $_POST['email'];
$subject = $_POST['subject'];
$id=uniqid();
$date=date("Y-m-d");
$time=date("h:i:sa");
$feedback = $_POST['feedback'];
$q=mysqli_query($con,"INSERT INTO feedback VALUES ('$id' , '$name', '
$email' , '$subject', '$feedback' , '$date' , '$time')")or die ("Error
");
header("location:$ref?q=Дякуємо за ваш відгук");
?>
```

Для ведення коректної роботи веб-додатку, управлінням контентом та користувачами було створено форму адміністрування з користувачами зі статусом адміністратора. Через головний інтерфейс користувача (рис. 3.6) надається можливість вимикати та вмикати тести, редагувати користувачів (рис. 3.7), редагувати та аналізувати відгуки, додавати та видаляти тести із системи.

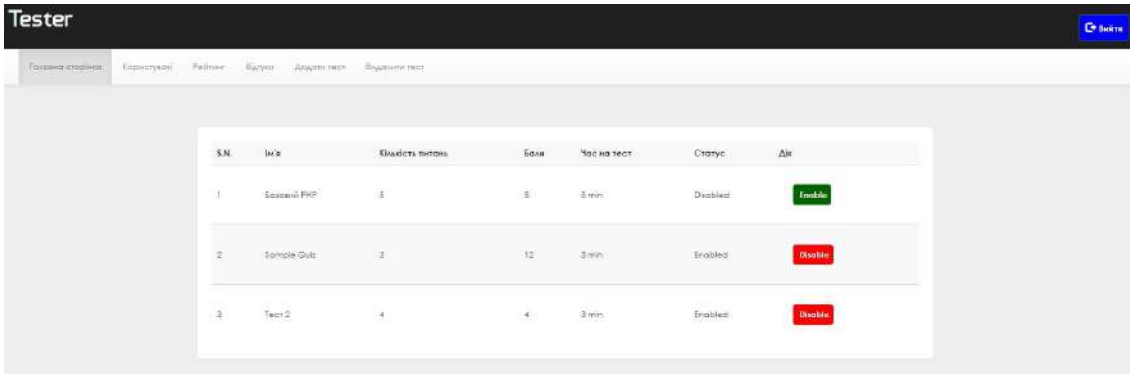


Рис. 3.6 Головний інтерфейс користувача (роль:адміністратор)

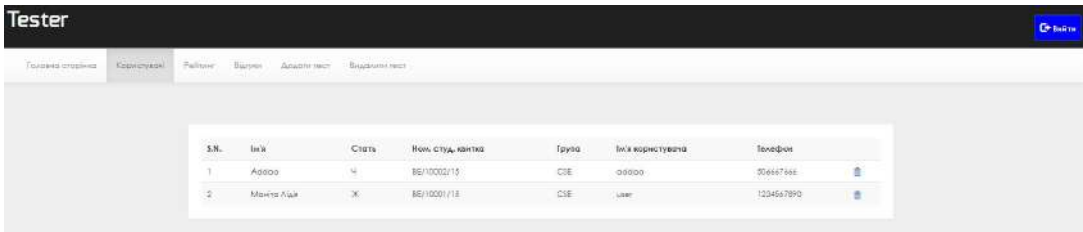


Рис. 3.7 Форма списку користувачів

У формі створення тестів надається можливість власноруч створити та налаштувати тести для студентів та користувачів веб-додатку. Можливо відредагувати кількість питань, кількість балів та штрафів, а також встановити обмеження по часу (рис. 3.8)

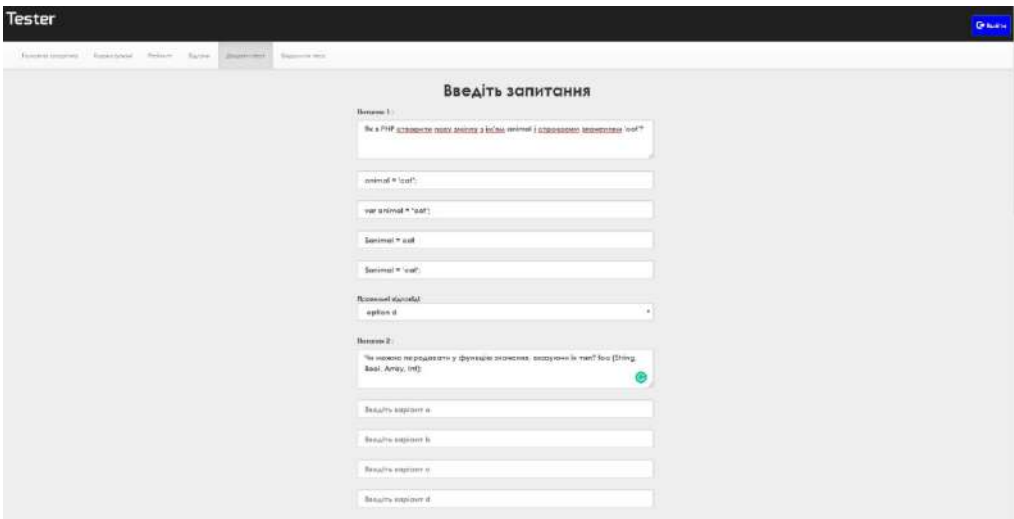


Рис. 3.8 Форма створення тестів

Фрагмент коду форми створення тестів наведено далі:

```
if (@$_GET['q'] == 4 && !(@$_GET['step'])) {
    echo '
<div class="row">
<span class="title1" style="margin-left:40%;font-size:30px;"><b></b></span><br /><br />
    <div class="col-md-3"></div><div class="col-md-6">
        <form class="form-horizontal title1" name="form" action="update.php?q=addquiz" method="POST">
        <fieldset>
        <div class="form-group">
            <label class="col-md-12 control-label" for="name"></label>
            <div class="col-md-12">
                <input id="name" name="name" placeholder="Назва тесту" class="form-control input-md" type="text">

            </div>
        </div>
        <div class="form-group">
            <label class="col-md-12 control-label" for="total"></label>
            <div class="col-md-12">
                <input id="total" name="total" placeholder="Введіть загальну кількість питань" class="form-control input-md" type="number">

            </div>
        </div>
        <div class="form-group">
            <label class="col-md-12 control-label" for="right"></label>
            <div class="col-md-12">
                <input id="right" name="right" placeholder="Введіть кількість балів за правильну відповідь" class="form-control input-md" min="0" type="number">

            </div>
        </div>
        <div class="form-group">
            <label class="col-md-12 control-label" for="wrong"></label>
            <div class="col-md-12">
                <input id="wrong" name="wrong" placeholder="Введіть кількість балів за неправильну відповідь" class="form-control input-md" min="0" type="number">

            </div>
        </div>
    </div>
}
```

```

<div class="form-group">
  <label class="col-md-12 control-label" for="time"></label>
  <div class="col-md-12">
    <input id="time" name="time" placeholder="Введіть обмеження часу для
    тесту в хвилину" class="form-control input-
    md" min="1" type="number">

  </div>
</div>

<div class="form-group">
  <label class="col-md-12 control-label" for=""></label>
  <div class="col-md-12">
    <input type="submit" style="margin-left:45%" class="btn btn-
    primary" value="Ок" class="btn btn-primary"/>
  </div>
</div>

</fieldset>
</form></div>';

}

```

Для видалення тестів використовується спеціально розроблена форма (рис.3.9).

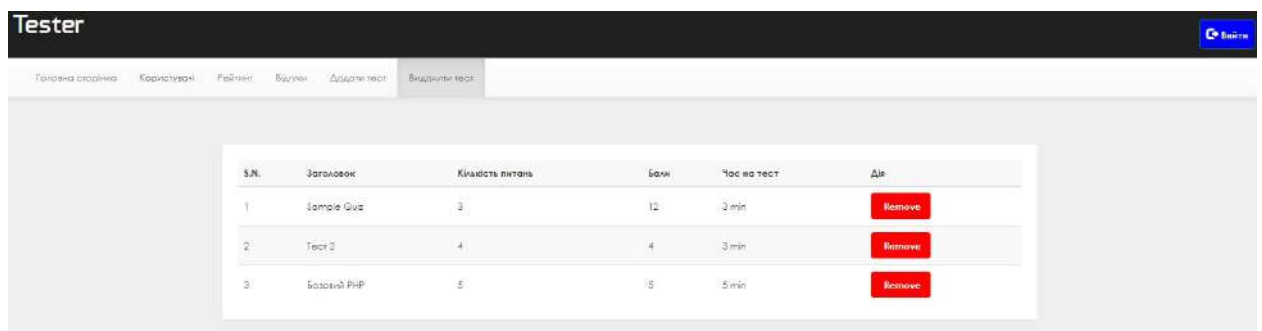


Рис. 3.9 Форма для видалення тестів

3.4 Висновки до розділу 3

В процесі дослідження було створено програмне забезпечення для організації та проведення тестів і опитувань у освітньому процесі. Це програмне забезпечення автоматизує процеси тестування і опитування, що дозволяє здійснювати їх швидко. Обрано розробку програмної системи на основі веб-технологій та з використанням модульної архітектури. Це сприяє підвищенню гнучкості та зручності системи як під час розробки і супроводу, так і в процесі використання. Результати виконання тестових завдань підтверджують коректність отриманих результатів, тому система відповідає поставленим вимогам. Користувачами системи можуть бути адміністратори, які редагують інформацію, створюють і керують тестами, а також студенти, які проходять тести. Програмне забезпечення може функціонувати на будь-якій операційній системі з встановленим браузером, який підтримує останні веб-стандарти, і з постійним доступом до Інтернету..

ВИСНОВКИ

Протягом останніх років у всьому світі в системах освіти відбулися суттєві структурні трансформації, що виникли внаслідок розвитку Інтернету та його зростаючого впливу на всі сфери суспільства. Вдосконаленню системи освіти найбільше сприятимуть нові інформаційні технології, зокрема, засоби дистанційного навчання. На сьогодні в світі вже накопичено значний досвід впровадження систем дистанційного навчання, які використовують комп'ютерні мережі, системи телевізійного мовлення та сучасні телекомунікаційні технології.

Надзвичайно важливо забезпечити постійну освіту і, що не менш важливо, ефективний контроль рівня знань. Тому в останні роки приділяється підвищена увага методикам дистанційного навчання та контролю знань. Комп'ютерні системи контролю знань є досить ефективними, дозволяючи не лише забезпечити державну перевірку якості знань, але й стати основою для самостійного навчання. Таким чином, детальне вивчення принципів, логіки, технологій та валідності дистанційного навчання і контролю знань, аналіз ефективності систем такого навчання та контролю, є актуальною і важливою проблемою, яка потребує вирішення.

Отже, комп'ютерне тестування розширює можливості контролю та оцінювання рівня навчальних досягнень учнів і представляє собою альтернативу традиційним методам перевірки. Воно може проводитися з врахуванням різних видів (поточного, тематичного, семестрового, річного) та форм (індивідуального або колективного) контролю як інструменту оперативного керівництва. Такий метод оцінювання швидко, об'єктивно та ефективно діагностує результати навчальної діяльності учнів, що робить його ефективним і перспективним засобом контролю.

У ході магістерського дослідження був проведений аналіз наявного програмного забезпечення для організації та проведення тестів і опитувань у

навчальному процесі, вивчено системи, що призначені для вирішення цих завдань. Відбувся огляд методів і засобів розробки програмної системи. Розглянуто обґрунтування вибору створення програмної системи, заснованої на веб-технологіях, і побудованої за модульною архітектурою. Це призвело до підвищення гнучкості та зручності системи як при розробці та супроводі, так і при використанні.

Результати тестових завдань підтвердили коректність отриманих результатів, вказуючи, що система відповідає встановленим вимогам. Користувачами системи можуть бути адміністратори, які редагують інформацію, створюють та керують тестами, а також студенти, які проходять тести. Програмне забезпечення може функціонувати на будь-якій операційній системі, де встановлено браузер, що підтримує найновіші веб-стандарти, і забезпечує постійний доступ до Інтернету.

Розроблений проєкт має актуальність у сучасній освіті, надаючи студентам та викладачам можливість швидко оцінити рівень засвоєних знань. Безкоштовність програмного забезпечення є його значною перевагою, оскільки освітні установи можуть використовувати його без додаткових витрат на ліцензію. Таким чином, цю систему комп'ютерного онлайн-тестування можна впроваджувати в навчальний процес для самоперевірки студентів, контролю знань та закріплення матеріалу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Адаменко О.В., Духовна М.М., Панченко Л.Ф. та ін. Тестові завдання для контролю знань в курсі "Обчислювальна техніка і технічні засоби навчання": Навч.- метод, посібник / За ред. Т.О. Козлакової. К.: ВІПОЛ, 1996. 84 с.
2. Алексейчук І.С. Про технологію створення системи тестування. Нові технології навчання: Науково-методичний збірник. К.: НМЦВД, 2000. С.43–92.
3. Алексюк А.М. Педагогіка вищої освіти України. К.:Либіть, 1998. 557 с.
4. Антонюк М.С. «Інформаційна технологія побудови автоматизованої системи управління навчальним процесом», автореферат на здобуття наукового ступеня кандидата технічних наук. 20 с.
5. Балл Г.О. Теория учебных задач: психолого-педагогический аспект. М.: Педагогика, 1990. 184 с.
6. Белоус Н.В., Войтович И.В., Пархоменко С.А. Модель навчання на основі тестових завдань довільних форм // Під ред. В.А. Гребенюка і В.В. Семенца. Харків; Ялта: УАДО, ХНУРЭ, 2003. С. 286 – 288.
7. Беркович В.Н. Самостоятельная работа заочников в условиях информатизации учебного процесса. *Информатика и образование*. 2007. №6. С. 30 – 32.
8. Беспалько В.П. Педагогика и прогрессивные технологии обучения. М.: Просвещение, 1995. 208 с.
9. Биков В.Ю., Лапінський В.В. Методологічні та методичні основи створення і використання електронних засобів навчального призначення. *Комп'ютер у школі та сім'ї*. №3 . 2012. С. 3– 6.
10. Болье А. — Learning SQL. (2005). URL: <http://shop.oreilly.com/product/9780596007270.do>.

11. Бондар В.І. Дидактика: ефективні технології навчання студентів. К.: Вересень, 1996. 129 с.
12. Бондар В.І. Теоретичні основи і технології педагогічного аналізу: управлінський аспект: Навчальний посібник. К., 1996. 67 с.
13. Бугко А.І. Крилов А.О., Сергєєва Т.П. Організація автоматизованого навчання. Інститут комп'ютерних технологій. К: УМК ВО, 1984. 274 с.
14. Булах І.С. Комп'ютерна діагностик навчальної успішності. К.: ЦКМ МОЗ України УДМУ, 1995. 221 с.
15. Васильева, Е. Ю. Подходы к оценке качества деятельности преподавателя вуза. *Университетское управление: практика и анализ*. 2006. № 2 (11). С. 74 – 78.
16. Вимірювання навчальних досягнень школярів і студентів: гуманістичні, методологічні, методичні, технологічні аспекти: І Міжнародна науково-методична конференція. Тези доповідей. Харків: ОВС, 2003. 112с.
17. Волков Н.И., Алексеев А.Н., Алексеева Н.А. «Тестовый контроль знаний: Учебное пособие» Сумы: ИТД «Университетская книга», 2004. С.108
18. Волков Н.И., Алексеев А.Н., Кочевский А.Н. Інформаційні технології в освіті, науці і техніці: Матеріали IV Усеукраїнської конференції молодих науковців ІТОНТ-2004. Черкаси: ЧНУ, 2004. С. 84– 87.
19. Журавльова О.Б. Дистанційне навчання: концепція, зміст, керування. Новосибірськ: Вид-во Сиб. гос. університету телекомунікацій й інформатики, 2001. 96с.
20. Іванов В.Ф., Мелещенко О.К. Сучасні комп'ютерні технології й засоби масової комунікації: аспекти застосування. Київ: ІЗМН, 1996. 243с.
21. Кейт Джонс. DOM Scripting: Web Design with JavaScript and the Document Object Model. Перше, 2005. 368 с.
22. Колисниченко Д. Н. Самоучитель PHP 5. СПб.: Наука и Техника, 2007. 640 с.

23. Краснова Г.А., Беляев М.И., Соловов А.В. Технологии создания электронных обучающих средств . М.:МГИУ, 2001. 224 с.
24. Кузнецов Максим, Симдянов Игорь MySQL на примерах. Спб.: «БХВ-Петербург», 2008. 952 с.
25. Кузнецов Максим, Симдянов Игорь PHP 5/6. Спб.: «БХВ-Петербург», 2009. 1024 с.
26. Кузнецов Максим, Симдянов Игорь Объектно-ориентированное программирование на PHP. Спб.: «БХВ-Петербург», 2007. 608 с.
27. Майоров А. Н. Теория и практика создания тестов для системы образования. М.: Народное образование, 2000. 352 с.
28. Машбиц Е.А. Психолого-педагогические проблемы компьютеризации обучения. М.: Педагогика, 1988. 191 с.
29. Млинців Ф.М. Ефективність навчання. М.: Педагогіка, 1976. 189 с.
30. Молочков В.П. Информационные технологии обучения. *Компьютерные учебные программы и инновации*. 2004. №1. С. 65-68.
31. Никандров Н.Д. Педагогіка вищої школи. Л.: ЛГПИ ім. А.И. Герцена, 1974. 116 с.
32. Педагогіка вищої школи: Учбово-методичний посібник / Під ред. Н.М. Пейсахова. Казань: Изд-во Казанського університету, 1985. 191 с
33. Роберт Шелдон, Джоффри Мойе MySQL: базовый курс Beginning MySQL. М.: «Диалектика» 2007. 880 с.
34. Скотт Хокінс. Адміністрування веб-сервера Apache і керівництво по електронній комерції. М.: «Вільямс», 2001. 336 с.
35. Тетьякова Н. В. Оценка качества работы преподавателя на основе методики многомерного анализа его деятельности. Научно-теоретический журнал "Ученые записки университета имени П. Ф. Лесгафта".2011. №11(81).С. 151-155.
36. Anthony Gore — Full-Stack Vue.js 2 and Laravel 5 . (2015). URL: <https://bit.ly/2OEODzR>.

37. Markus Egger — MVVM Survival Guide for Enterprise Architectures in Silverlight and WPF. (2012). URL: <https://www.packtpub.com/application-development/mvvm-survival-guide-enterprise-architectures-silverlight-and-wpf>.
38. Martin Fowler — GUI Architectures. Часть 1. (2009). URL: <https://bit.ly/2CvCk1e>.
39. MySQL. Довідник. MySQL AB. М: «Вільямс», 2006 521 с.
40. Vue.js Material Component Framework — Vuetify.js. (2016). URL: <https://vuetifyjs.com>.
41. What is REST? URL: <http://www.restapitutorial.com/lessons/whatisrest.html>.


```

<div class="container">
<div class="row">
<div class="col-md-12">
<?php
if (@$_GET['q'] == 1) {

    $result = mysqli_query($con, "SELECT * FROM quiz WHERE status = 'enabled' ORDER BY date DESC") or die('Error');
    echo '<div class="panel"><table class="table table-striped title1" style="vertical-align:middle">
<tr><td style="vertical-align:middle"><b>S.N.</b></td><td style="vertical-align:middle"><b>Назва</b></td><td style="vertical-align:middle"><b>Кількість питань</b></td><td style="vertical-align:middle"><b>Правильні відповіді</b></td><td style="vertical-align:middle"><b>Неправильні відповіді</b></td><td style="vertical-align:middle"><b>Балів</b></td><td style="vertical-align:middle"><b>Час на тест</b></td><td style="vertical-align:middle"><b>Дія</b></td></tr>';
    $c = 1;
    while ($row = mysqli_fetch_array($result)) {
        $title = $row['title'];
        $total = $row['total'];
        $correct = $row['correct'];
        $wrong = $row['wrong'];
        $time = $row['time'];
        $eid = $row['eid'];
        $q12 = mysqli_query($con, "SELECT score FROM history WHERE eid='$eid' AND username='$username'") or die('Error98');
        $rowcount = mysqli_num_rows($q12);
        if ($rowcount == 0) {
            echo '<tr><td style="vertical-align:middle">' . $c++ . '</td><td style="vertical-align:middle">' . $title . '</td><td style="vertical-align:middle">' . $total . '</td><td style="vertical-align:middle">' . $correct . '</td><td style="vertical-align:middle">' . $wrong . '</td><td style="vertical-align:middle">' . $correct * $total . '</td><td style="vertical-align:middle">' . $time . '&nbsp;&nbsp;&nbsp;</td><td style="vertical-align:middle"><b><a href="account.php?q=quiz&step=2&eid=' . $eid . '&n=1&t=' . $total . '&start=start" class="btn" style="color:#FFFFFF;background:darkgreen;font-size:12px;padding:7px;padding-left:10px;padding-right:10px"><span class="glyphicon glyphicon-new-window" aria-hidden="true"></span>&nbsp;&nbsp;<span><b>Пройти</b></span></a></b></td></tr>';
        } else {
            $q = mysqli_query($con, "SELECT * FROM history WHERE username = '$_SESSION[username]' AND eid='$eid' ") or die('Error197');
            while ($row = mysqli_fetch_array($q)) {
                $timec = $row['timestamp'];
                $status = $row['status'];
            }
        }
    }
}

```



```

        $q = mysqli_query($con, "SELECT * FROM quiz WHERE eid='$eid'
") or die('Error197');
        while ($row = mysqli_fetch_array($q)) {
            $ttimec = $row['time'];
            $qstatus = $row['status'];
        }
        $remaining = (($ttimec * 60) - ((time() - $timec)));
        if ($remaining > 0 && $qstatus == "enabled" && $status == "on
going") {
            echo '<tr style="color:darkgreen"><td style="vertical-
align:middle">' . $c++ . '</td><td style="vertical-
align:middle">' . $title . '&nbsp;<span title="This quiz is already solve
by you" class="glyphicon glyphicon-ok" aria-
hidden="true"></span></td><td style="vertical-
align:middle">' . $total . '</td><td style="vertical-
align:middle">+ ' . $correct . '</td><td style="vertical-align:middle">-
' . $wrong . '</td><td style="vertical-
align:middle">' . $correct * $total . '</td><td style="vertical-
align:middle">' . $time . '&nbsp;<min</td>
<td style="vertical-
align:middle"><b><a href="account.php?q=quiz&step=2&eid=' . $eid . '&n=1&
t=' . $total . '&start=start" class="btn" style="margin:0px;background:da
rkorange;color:white">&nbsp;<span class="title1"><b>Continue</b></span></a></b></td></tr>';
        } else {
            echo '<tr style="color:darkgreen"><td style="vertical-
align:middle">' . $c++ . '</td><td style="vertical-
align:middle">' . $title . '&nbsp;<span title="This quiz is already solve
by you" class="glyphicon glyphicon-ok" aria-
hidden="true"></span></td><td style="vertical-
align:middle">' . $total . '</td><td style="vertical-
align:middle">+ ' . $correct . '</td><td style="vertical-align:middle">-
' . $wrong . '</td><td style="vertical-
align:middle">' . $correct * $total . '</td><td style="vertical-
align:middle">' . $time . '&nbsp;<min</td>
<td style="vertical-
align:middle"><b><a href="account.php?q=result&eid=' . $eid . '" class="b
tn" style="margin:0px;background:darkred;color:white">&nbsp;<span class="
title1"><b>Результати</b></span></a></b></td></tr>';
        }
    }
    $c = 0;
    echo '</table></div><div class="panel" style="padding-
top:1px;padding-left:15%;padding-right:15%;word-wrap:break-
word"><h3 align="center" style="font-
family:calibri">:: Загальні інструкції ::</h3><br /><ul type="circle"><fo
nt style="font-size:14px;font-family:calibri">';
    $file = fopen("instructions.txt", "r");
    while (!feof($file)) {
        echo '<li>';
        $string = fgets($file);
    }

```

```

$num    = strlen($string) - 1;
$c      = str_split($string);
for ($i = 0; $i < $num; $i++) {
    $last = $c[$i - 1];
    if ($c[$i] == ' ' && $last == ' ') {
        echo '&nbsp;';
    } else {
        echo $c[$i];
    }
}
echo "</li><br />";
}

fclose($file);
echo '</font></ul></div>';

}
?>
<?php
if (@$_GET['q'] == 'quiz' && @$_GET['step'] == 2 && isset($_SESSION['6e447159425d2d']) && $_SESSION['6e447159425d2d'] == "6e447159425d2d" && $_GET['endquiz'] == 'end') {
    unset($_SESSION['6e447159425d2d']);
    $q = mysqli_query($con, "UPDATE history SET status='finished' WHERE username='$_SESSION[username]' AND eid='$_GET[eid]' ") or die('Error197');
    $q = mysqli_query($con, "SELECT * FROM history WHERE eid='$_GET[eid]' AND username='$_SESSION[username]'") or die('Error156');
    while ($row = mysqli_fetch_array($q)) {
        $s = $row['score'];
        $scorestatus = $row['score_updated'];
    }
    if($scorestatus=="false"){
        $q = mysqli_query($con, "UPDATE history SET score_updated='true' WHERE username='$_SESSION[username]' AND eid='$_GET[eid]' ") or die('Error197');
        $q = mysqli_query($con, "SELECT * FROM rank WHERE username='$_SESSION[username]'") or die('Error161');
        $rowcount = mysqli_num_rows($q);
        if ($rowcount == 0) {
            $q2 = mysqli_query($con, "INSERT INTO rank VALUES (NULL,'$_SESSION[username]','$s',NOW())") or die('Error165');
        } else {
            while ($row = mysqli_fetch_array($q)) {
                $sun = $row['score'];
            }

            $sun = $s + $sun;
            $q = mysqli_query($con, "UPDATE `rank` SET `score`=$sun ,time=NOW() WHERE username= '$_SESSION[username]'") or die('Error174');
        }
    }
    header('location:account.php?q=result&eid=' . $_GET[eid]);
}

```



```

}

if (@$_GET['q'] == 'quiz' && @$_GET['step'] == 2 && isset($_GET['start'])
    && $_GET['start'] == "start" && (!isset($_SESSION['6e447159425d2d']))) {
    $q = mysqli_query($con, "SELECT * FROM history WHERE username='$usern
ame' AND eid='$_GET[eid]' ") or die('Error197');

    if (mysqli_num_rows($q) > 0) {
        $q = mysqli_query($con, "SELECT * FROM history WHERE username='$_
SESSION[username]' AND eid='$_GET[eid]' ") or die('Error197');
        while ($row = mysqli_fetch_array($q)) {
            $timel = $row['timestamp'];
            $status = $row['status'];
        }
        $q = mysqli_query($con, "SELECT * FROM quiz WHERE eid='$_GET[eid]
' ") or die('Error197');
        while ($row = mysqli_fetch_array($q)) {
            $ttimel = $row['time'];
            $qstatus = $row['status'];
        }
        $remaining = (($ttimel * 60) - ((time() - $timel)));
        if ($status == "ongoing" && $remaining > 0 && $qstatus == "enable
d") {
            $_SESSION['6e447159425d2d'] = "6e447159425d2d";
            header('location:account.php?q=quiz&step=2&eid=' . $_GET[eid]
. '&n=' . $_GET[n] . '&t=' . $_GET[t]);
        } else {
            $q = mysqli_query($con, "UPDATE history SET status='finis
hed' WHERE username='$_SESSION[username]' AND eid='$_GET[eid]' ") or die(
'Error197');
            $q = mysqli_query($con, "SELECT * FROM history WHERE eid='$_GET[e
id]' AND username='$_SESSION[username]'") or die('Error156');
            while ($row = mysqli_fetch_array($q)) {
                $s = $row['score'];
                $scorestatus = $row['score_updated'];
            }
            if($scorestatus=="false"){
                $q = mysqli_query($con, "UPDATE history SET score_upd
ated='true' WHERE username='$_SESSION[username]' AND eid='$_GET[eid]' ")
or die('Error197');
                $q = mysqli_query($con, "SELECT * FROM rank WHERE use
rname='$username'") or die('Error161');
                $rowcount = mysqli_num_rows($q);
                if ($rowcount == 0) {
                    $q2 = mysqli_query($con, "INSERT INTO rank VALUES
(NULL, '$username', '$s', NOW())") or die('Error165');
                } else {
                    while ($row = mysqli_fetch_array($q)) {
                        $sun = $row['score'];
                    }
                }
            }
        }
    }
}

```

```

        $sun = $s + $sun;
        $q = mysqli_query($con, "UPDATE `rank` SET `score`
        `=$sun ,time=NOW() WHERE username= '$username'" ) or die('Error174');
    }
    }
    header('location:account.php?q=result&eid=' . $_GET[eid]);
}

} else {
    $time = time();
    $q = mysqli_query($con, "INSERT INTO history VALUES(NULL, '$userna
me', '$_GET[eid]', '0', '0', '0', '0', NOW(), '$time', 'ongoing', 'false' )" or d
ie('Error137');
    $_SESSION['6e447159425d2d'] = "6e447159425d2d";
    header('location:account.php?q=quiz&step=2&eid=' . $_GET[eid] . '
&n=' . $_GET[n] . '&t=' . $_GET[t]);
}
}

if (@$_GET['q'] == 'quiz' && @$_GET['step'] == 2 && isset($_SESSION['6e44
7159425d2d']) && $_SESSION['6e447159425d2d'] == "6e447159425d2d") {
    $q = mysqli_query($con, "SELECT * FROM history WHERE username='$usern
ame' AND eid='$_GET[eid]' ") or die('Error197');

    if (mysqli_num_rows($q) > 0) {
        $q = mysqli_query($con, "SELECT * FROM history WHERE username='$_
SESSION[username]' AND eid='$_GET[eid]' ") or die('Error197');
        while ($row = mysqli_fetch_array($q)) {
            $time = $row['timestamp'];
            $status = $row['status'];
        }
        $q = mysqli_query($con, "SELECT * FROM quiz WHERE eid='$_GET[eid]
' ") or die('Error197');
        while ($row = mysqli_fetch_array($q)) {
            $ttime = $row['time'];
            $qstatus = $row['status'];
        }
        $remaining = (($ttime * 60) - ((time() - $time)));
        if ($status == "ongoing" && $remaining > 0 && $qstatus == "enable
d") {
            $q = mysqli_query($con, "SELECT * FROM history WHERE username
='$_SESSION[username]' AND eid='$_GET[eid]' ") or die('Error197');
            while ($row = mysqli_fetch_array($q)) {
                $time = $row['timestamp'];
            }
            $q = mysqli_query($con, "SELECT * FROM quiz WHERE eid='$_GET[
eid]' ") or die('Error197');
            while ($row = mysqli_fetch_array($q)) {
                $ttime = $row['time'];
            }
            $remaining = (($ttime * 60) - ((time() - $time)));
            echo '<script>

```



```

var seconds = ' . $remaining . ' ;
function end(){
    data = prompt("Are you sure to end this Quiz? Remember, once finished,
you wont be able to continue anymore and final results will be displayed.
If you want to continue then enter \"yes\" in the textbox below and pr
ess enter");
    if(data=="yes"){
        window.location = "account.php?q=quiz&step=2&eid=' . $_GET[eid] . '&n=
' . $_GET[n] . '&t=' . $_GET[total] . '&endquiz=end";
    }
}
function enable(){
    document.getElementById("sbutton").removeAttribute("disabled");
}
function frmreset(){
    document.getElementById("sbutton").setAttribute("disabled","true");
    document.getElementById("qform").reset();
}
function secondPassed() {
    var minutes = Math.round((seconds - 30)/60);
    var remainingSeconds = seconds % 60;
    if (remainingSeconds < 10) {
        remainingSeconds = "0" + remainingSeconds;
    }
    document.getElementById(\'countdown\').innerHTML = minutes + ":" +
remainingSeconds;
    if (seconds <= 0) {
        clearInterval(countdownTimer);
        document.getElementById(\'countdown\').innerHTML = "Buzz Buzz..."
;
        window.location = "account.php?q=quiz&step=2&eid=' . $_GET[eid] .
'&n=' . $_GET[n] . '&t=' . $_GET[total] . '&endquiz=end";
    } else {
        seconds--;
    }
}
var countdownTimer = setInterval(\'secondPassed()\', 1000);
</script>';
    echo '<font size="3" style="margin-left:100px;font-
family:\'typo\' font-size:20px; font-
weight:bold;color:darkred">Time Left : </font><span class="timer btn btn-
default" style="margin-left:20px;"><font style="font-
family:\'typo\' ;font-size:20px;font-
weight:bold;color:darkblue" id="countdown"></font></span><span class="tim
er btn btn-primary" style="margin-
left:50px" onclick="end()"><span class=" glyphicon glyphicon-
off"></span>&nbsp;&nbsp;&nbsp;<font style="font-size:12px;font-
weight:bold">Finish Quiz</font></span>';
    $eid = @$_GET['eid'];
    $sn = @$_GET['n'];
    $total = @$_GET['t'];

```

```

        $q = mysqli_query($con, "SELECT * FROM questions WHERE ei
d='$eid' AND sn='$sn' ");
        echo '<div class="panel" style="margin-right:5%;margin-
left:5%;margin-top:10px;border-radius:10px">';
        while ($row = mysqli_fetch_array($q)) {
            $qns = stripslashes($row['qns']);
            $qid = $row['qid'];
            echo '<b><pre style="background-
color:white"><div style="font-size:20px;font-weight:bold;font-
family:calibri;margin:10px">' . $sn . ' : ' . $qns . '</div></pre></b>';
        }

        echo '<form id="qform" action="update.php?q=quiz&step=2&eid='
. $eid . '&n=' . $sn . '&t=' . $total . '&qid=' . $qid . '" method="POST
" class="form-horizontal">
<br />';
        $q = mysqli_query($con, "SELECT * FROM user_answer WHERE qid=
'$qid' AND username='$_SESSION[username]' AND eid='$_GET[eid]'") or die("
Error222");
        if (mysqli_num_rows($q) > 0) {
            $row = mysqli_fetch_array($q);
            $ans = $row['ans'];
            $q = mysqli_query($con, "SELECT * FROM options WHERE qid=
'$qid' AND optionid='$ans'") or die("Error222");
            $row = mysqli_fetch_array($q);
            $ans = $row['option'];
        } else {
            $ans = "";
        }
        if (strlen($ans) > 0) {
            echo "<font style=\"color:green;font-size:12px;font-
weight:bold\">Selected answer: </font><font style=\"color:#565252;font-
size:12px;\">" . $ans . "</font>&nbsp;&nbsp;&nbsp;<a href=update.php?q=quiz&ste
p=2&eid=$eid&n=$sn&t=$total&qid=$qid&delanswer=delanswer><span class=\"gl
yphicon glyphicon-remove\" style=\"font-
size:12px;color:darkred\"></span></a><br /><br />";
        }
        echo '<div class="funkyradio">';
        $q = mysqli_query($con, "SELECT * FROM options WHERE qid='$qi
d' ");
        while ($row = mysqli_fetch_array($q)) {
            $option = stripslashes($row['option']);
            $optionid = $row['optionid'];
            echo '<div class="funkyradio-
success"><input type="radio" id="' . $optionid . '" name="ans" value="' .
$optionid . '" onclick="enable()"> <label for="' . $optionid . '" style=
"width:50%"><div style="color:black;font-size:12px;word-wrap:break-
word">&nbsp;&nbsp;&nbsp;' . $option . '</div></label></div>';
        }
        echo '</div>';
        if ($_GET[t] > $_GET[n] && $_GET[n] != 1) {

```



```

        echo '<br /><a href="account.php?q=quiz&step=2&eid=' . $eid . '&n=' . ($sn - 1) . '&t=' . $total . '" class="btn btn-primary" style="height:30px"><span class="glyphicon glyphicon-arrow-left" aria-hidden="true" style="font-size:12px"></span></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<button type="submit" class="btn btn-default" disabled="true" id="sbutton" style="height:30px"><span class="glyphicon glyphicon-lock" style="font-size:12px" aria-hidden="true"></span><font style="font-size:12px;font-weight:bold"> Lock</font></button>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<button type="button" class="btn btn-default" onclick="frmreset()" style="height:30px"></span><font style="font-size:12px;font-weight:bold">Reset</font></button>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="account.php?q=quiz&step=2&eid=' . $eid . '&n=' . ($sn + 1) . '&t=' . $total . '" class="btn btn-primary" style="height:30px"><span class="glyphicon glyphicon-arrow-right" aria-hidden="true" style="font-size:12px"></span></a></form><br><br>';
    } else if ($_GET[t] == $_GET[n]) {
        echo '<br /><a href="account.php?q=quiz&step=2&eid=' . $eid . '&n=' . ($sn - 1) . '&t=' . $total . '" class="btn btn-primary" style="height:30px"><span class="glyphicon glyphicon-arrow-left" aria-hidden="true" style="font-size:12px"></span></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<button type="submit" class="btn btn-default" disabled="true" id="sbutton" style="height:30px"><span class="glyphicon glyphicon-lock" style="font-size:12px" aria-hidden="true"></span><font style="font-size:12px;font-weight:bold"> Lock</font></button>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<button type="button" class="btn btn-default" onclick="frmreset()" style="height:30px"></span><font style="font-size:12px;font-weight:bold">Reset</font></button>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</form><br><br>';
    } else if ($_GET[t] > $_GET[n] && $_GET[n] == 1) {
        echo '<br />&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<button type="submit" class="btn btn-default" disabled="true" id="sbutton" style="height:30px"><span class="glyphicon glyphicon-lock" style="font-size:12px" aria-hidden="true"></span><font style="font-size:12px;font-weight:bold"> Lock</font></button>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<button type="button" class="btn btn-default" onclick="frmreset()" style="height:30px"></span><font style="font-size:12px;font-weight:bold">Reset</font></button>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="account.php?q=quiz&step=2&eid=' . $eid . '&n=' . ($sn + 1) . '&t=' . $total . '" class="btn btn-primary" style="height:30px"><span class="glyphicon glyphicon-arrow-right" aria-hidden="true" style="font-size:12px"></span></a></form><br><br>';
    } else {

```

```

    }
    echo '</div>';
    echo '<div class="panel" style="text-align:center">';
    $q = mysqli_query($con, "SELECT * FROM questions WHERE eid='$_GET[eid]'" ) or die("Error222");
    $i = 1;
    while ($row = mysqli_fetch_array($q)) {
        $ques[$row['qid']] = $i;
        $i++;
    }
    $q = mysqli_query($con, "SELECT * FROM user_answer WHERE eid=
'_GET[eid]' AND username='$_SESSION[username]'" ) or die("Error222a");
    $i = 1;
    while ($row = mysqli_fetch_array($q)) {
        if (isset($ques[$row['qid']])) {
            $quesans[$ques[$row['qid']]] = true;
        }
    }
    for ($i = 1; $i <= $total; $i++) {
        echo '<a href="account.php?q=quiz&step=2&eid=' . $eid . '
&n=' . $i . '&t=' . $total . '" style="margin:5px;padding:5px;background
-color:';

        if ($quesans[$i]) {
            echo "darkgreen";
        } else {
            echo "darkred";
        }

        echo '";color:white;font-size:16px;font-
family:calibri;border-radius:4px">&nbsp;' . $i . '&nbsp;</a>';
    }
    } else {
        unset($_SESSION['6e447159425d2d']);
        $q = mysqli_query($con, "UPDATE history SET status='finished'
WHERE username='$_SESSION[username]' AND eid='$_GET[eid]' " ) or die('Err
or197');
        $q = mysqli_query($con, "SELECT * FROM history WHERE eid='$_GET[e
id]' AND username='$_SESSION[username]'" ) or die('Error156');
        while ($row = mysqli_fetch_array($q)) {
            $s = $row['score'];
            $scorestatus = $row['score_updated'];
        }
        if($scorestatus=="false"){
            $q = mysqli_query($con, "UPDATE history SET score_upd
ated='true' WHERE username='$_SESSION[username]' AND eid='$_GET[eid]' " )
or die('Error197');
            $q = mysqli_query($con, "SELECT * FROM rank WHERE use
rname='$username'" ) or die('Error161');
            $rowcount = mysqli_num_rows($q);
            if ($rowcount == 0) {
                $q2 = mysqli_query($con, "INSERT INTO rank VALUES
(NULL,'$username','$s',NOW())" ) or die('Error165');
            } else {

```



```

        while ($row = mysqli_fetch_array($q)) {
            $sun = $row['score'];
        }

        $sun = $s + $sun;
        $q = mysqli_query($con, "UPDATE `rank` SET `score`
`=$sun ,time=NOW() WHERE username= '$username'" ) or die('Error174');
    }
    }
    header('location:account.php?q=result&eid=' . $_GET['eid']);
}
} else {
    unset($_SESSION['6e447159425d2d']);
    $q = mysqli_query($con, "UPDATE history SET status='finished' WHE
RE username='$_SESSION[username]' AND eid='$_GET[eid]' ") or die('Error19
7');
    $q = mysqli_query($con, "SELECT * FROM history WHERE eid='$_GET[e
id]' AND username='$_SESSION[username]'" ) or die('Error156');
    while ($row = mysqli_fetch_array($q)) {
        $s = $row['score'];
        $scorestatus = $row['score_updated'];
    }
    if($scorestatus=="false"){
        $q = mysqli_query($con, "UPDATE history SET score_upd
ated='true' WHERE username='$_SESSION[username]' AND eid='$_GET[eid]' ")
or die('Error197');
        $q = mysqli_query($con, "SELECT * FROM rank WHERE use
rname='$username'" ) or die('Error161');
        $rowcount = mysqli_num_rows($q);
        if ($rowcount == 0) {
            $q2 = mysqli_query($con, "INSERT INTO rank VALUES
(NULL, '$username', '$s', NOW())" ) or die('Error165');
        } else {
            while ($row = mysqli_fetch_array($q)) {
                $sun = $row['score'];
            }

            $sun = $s + $sun;
            $q = mysqli_query($con, "UPDATE `rank` SET `score`
`=$sun ,time=NOW() WHERE username= '$username'" ) or die('Error174');
        }
    }
    header('location:account.php?q=result&eid=' . $_GET['eid']);
}
}
if (@$_GET['q'] == 'result' && @$_GET['eid']) {
    $eid = @$_GET['eid'];
    $q = mysqli_query($con, "SELECT * FROM quiz WHERE eid='$eid' ") or di
e('Error157');
    while ($row = mysqli_fetch_array($q)) {
        $total = $row['total'];
    }
}

```

```

    $q = mysqli_query($con, "SELECT * FROM history WHERE eid='$eid' AND u
sername='$username' ") or die('Error157');

    while ($row = mysqli_fetch_array($q)) {
        $s      = $row['score'];
        $w      = $row['wrong'];
        $r      = $row['correct'];
        $status = $row['status'];
    }
    if ($status == "finished") {
        echo '<div class="panel">
<center><h1 class="title" style="color:#660033">Результат</h1><center><br
/><table class="table table-striped title1" style="font-size:20px;font-
weight:1000;">;
        echo '<tr style="color:darkblue"><td style="vertical-
align:middle">Кількість питань</td><td style="vertical-
align:middle">' . $total . '</td></tr>
        <tr style="color:darkgreen"><td style="vertical-
align:middle">Правильні відповіді&nbsp;<span class="glyphicon glyphicon-
ok-arrow" aria-hidden="true"></span></td><td style="vertical-
align:middle">' . $r . '</td></tr>
        <tr style="color:red"><td style="vertical-
align:middle">Wrong Answer&nbsp;<span class="glyphicon glyphicon-remove-
arrow" aria-hidden="true"></span></td><td style="vertical-
align:middle">' . $w . '</td></tr>
        <tr style="color:orange"><td style="vertical-
align:middle">Без спроб&nbsp;<span class="glyphicon glyphicon-ban-
arrow" aria-hidden="true"></span></td><td style="vertical-
align:middle">' . ($total - $r - $w) . '</td></tr>
        <tr style="color:darkblue"><td style="vertical-
align:middle">Результат&nbsp;<span class="glyphicon glyphicon-star" aria-
hidden="true"></span></td><td style="vertical-
align:middle">' . $s . '</td></tr>';
        $q = mysqli_query($con, "SELECT * FROM rank WHERE username='$use
rname' ") or die('Error157');
        while ($row = mysqli_fetch_array($q)) {
            $s = $row['score'];
            echo '<tr style="color:#990000"><td style="vertical-
align:middle">Overall Score&nbsp;<span class="glyphicon glyphicon-
stats" aria-hidden="true"></span></td><td style="vertical-
align:middle">' . $s . '</td></tr>';
        }
        echo '<tr></tr></table></div><div class="panel"><br /><h3 align="
center" style="font-
family:calibri">:: Detailed Analysis ::</h3><br /><ol style="font-
size:20px;font-weight:bold;font-family:calibri;margin-top:20px">';
        $q = mysqli_query($con, "SELECT * FROM questions WHERE eid='$_GET
[eid]'" ) or die('Error197');
        while ($row = mysqli_fetch_array($q)) {
            $question = $row['qns'];
            $qid      = $row['qid'];

```



```

        $q2 = mysqli_query($con, "SELECT * FROM user_answer WHERE eid
        = '$_GET[eid]' AND qid='$qid' AND username='$_SESSION[username]'" ) or die(
        'Error197');
        if (mysqli_num_rows($q2) > 0) {
            $row1
                = mysqli_fetch_array($q2);
            $ansid
                = $row1['ans'];
            $correctansid
                = $row1['correctans'];
            $q3 = mysqli_query($con, "SELECT * FROM options WHERE opt
            ionid='$ansid'" ) or die('Error197');
            $q4 = mysqli_query($con, "SELECT * FROM options WHERE opt
            ionid='$correctansid'" ) or die('Error197');
            $row2
                = mysqli_fetch_array($q3);
            $row3
                = mysqli_fetch_array($q4);
            $ans
                = $row2['option'];
            $correctans
                = $row3['option'];
        } else {
            $q3 = mysqli_query($con, "SELECT * FROM answer WHERE qid=
            '$qid'" ) or die('Error197');
            $row1
                = mysqli_fetch_array($q3);
            $correctansid
                = $row1['ansid'];
            $q4 = mysqli_query($con, "SELECT * FROM options WHERE opt
            ionid='$correctansid'" ) or die('Error197');
            $row2
                = mysqli_fetch_array($q4);
            $correctans
                = $row2['option'];
            $ans
                = "Unanswered";
        }
        if ($correctans == $ans && $ans != "Unanswered") {
            echo '<li><div style="font-size:16px;font-
            weight:bold;font-family:calibri;margin-top:20px;background-
            color:lightgreen;padding:10px;word-wrap:break-
            word;border:2px solid darkgreen;border-
            radius:10px;">' . $question . ' <span class="glyphicon glyphicon-
            ok" style="color:darkgreen"></span></div><br />';
            echo '<font style="font-
            size:14px;color:darkgreen"><b>Ваша відповідь: </b></font><font style="fon
            t-size:14px;">' . $ans . '</font><br />';
            echo '<font style="font-
            size:14px;color:darkgreen"><b>Правильні відповіді: </b></font><font style
            ="font-size:14px;">' . $correctans . '</font><br />';
        }
        else if ($ans == "Unanswered") {
            echo '<li><div style="font-size:16px;font-
            weight:bold;font-family:calibri;margin-top:20px;background-
            color:#f7f576;padding:10px;word-wrap:break-
            word;border:2px solid #b75a0e;border-
            radius:10px;">' . $question . ' </div><br />';
            echo '<font style="font-
            size:14px;color:darkgreen"><b>Правильні відповіді: </b></font><font style
            ="font-size:14px;">' . $correctans . '</font><br />';
        }
        else {

```

```

        echo '<li><div style="font-size:16px;font-
weight:bold;font-family:calibri;margin-top:20px;background-
color:#f99595;padding:10px;word-wrap:break-
word;border:2px solid darkred;border-
radius:10px;">' . $question . ' <span class="glyphicon glyphicon-
remove" style="color:red"></span></div><br />';
        echo '<font style="font-
size:14px;color:darkgreen"><b>Ваша відповідь: </b></font><font style="font-
size:14px;">' . $ans . ' </font><br />';
        echo '<font style="font-
size:14px;color:red"><b>Правильні відповіді: </b></font><font style="font-
size:14px;">' . $correctans . ' </font><br />';

    }
    echo "<br /></li>";
}
echo '</ol>';
echo "</div>";
} else {
    die("Thats a 404 Error bro. You are trying to access a wrong page
");
}
}
if (@$_GET['q'] == 2) {
    $q = mysqli_query($con, "SELECT * FROM history WHERE username='$usern
ame' AND status='finished' ORDER BY date DESC ") or die('Error197');
    echo '<div class="panel title">
<table class="table table-striped title1" >
<tr><td style="vertical-
align:middle"><b>S.N.</b></td><td style="vertical-
align:middle"><b>Quiz</b></td><td style="vertical-
align:middle"><b>Кількість питань</b></td><td style="vertical-
align:middle"><b>Правильно</b></td><td style="vertical-
align:middle"><b>Неправильно<b></td><td style="vertical-
align:middle"><b>Без спроб<b></td><td style="vertical-
align:middle"><b>Результат</b></td><td style="vertical-
align:middle"><b>Дія<b></td></tr>';
    $c = 0;
    while ($row = mysqli_fetch_array($q)) {
        $eid = $row['eid'];
        $s = $row['score'];
        $w = $row['wrong'];
        $r = $row['correct'];
        $q23 = mysqli_query($con, "SELECT * FROM quiz WHERE eid='$eid' "
) or die('Error208');
        while ($row = mysqli_fetch_array($q23)) {
            $title = $row['title'];
            $total = $row['total'];
        }
        $c++;
        echo '<tr><td style="vertical-
align:middle">' . $c . ' </td><td style="vertical-

```



```

align:middle">' . $title . '</td><td style="vertical-
align:middle">' . $total . '</td><td style="vertical-
align:middle">' . $r . '</td><td style="vertical-
align:middle">' . $w . '</td><td style="vertical-
align:middle">' . ($total - $r - $w) . '</td><td style="vertical-
align:middle">' . $s . '</td><td style="vertical-
align:middle"><b><a href="account.php?q=result&eid=' . $eid . '" class="b
tn" style="margin:0px;background:darkred;color:white">&nbsp;<span class="
title1"><b>Результати</b></td></tr>';
    }
    echo '</table></div>';
}
if (@$_GET['q'] == 3) {
    if(isset($_GET['show'])){
        $show = $_GET['show'];
        $showfrom = (($show-1)*10) + 1;
        $showtill = $showfrom + 9;
    }
    else{
        $show = 1;
        $showfrom = 1;
        $showtill = 10;
    }
    $q = mysqli_query($con, "SELECT * FROM rank") or die('Error223');
    echo '<div class="panel title">
<table class="table table-striped title1" >
<tr><td style="vertical-
align:middle"><b>Місце</b></td><td style="vertical-
align:middle"><b>Ім'я</b></td><td style="vertical-
align:middle"><b>Група</b></td><td style="vertical-
align:middle"><b>Ім'я користувача</b></td><td style="vertical-
align:middle"><b>Результат</b></td></tr>';
    $c = $showfrom-1;
    $total = mysqli_num_rows($q);
    if($total >= $showfrom){
        $q = mysqli_query($con, "SELECT * FROM rank ORDER BY score DESC,
time ASC LIMIT ".$showfrom-1.",10") or die('Error223');
        while ($row = mysqli_fetch_array($q)) {
            $e = $row['username'];
            $s = $row['score'];
            $q12 = mysqli_query($con, "SELECT * FROM user WHERE username=
'$e' ") or die('Error231');
            while ($row = mysqli_fetch_array($q12)) {
                $name = $row['name'];
                $branch = $row['branch'];
                $username = $row['username'];
            }
            $c++;
            echo '<tr><td style="color:#99cc32"><b>' . $c . '</b></td><td
style="vertical-align:middle">' . $name . '</td><td style="vertical-
align:middle">' . $branch . '</td><td style="vertical-

```

```

align:middle">' . $username . '</td><td style="vertical-
align:middle">' . $s . '</td><td style="vertical-align:middle">';
    }
    }
    else{
    }
    echo '</table></div>';
    echo '<div class="panel title"><table class="table table-
striped title1" ><tr>';
    $total = round($total/10) + 1;
    if(isset($_GET['show'])){
        $show = $_GET['show'];
    }
    else{
        $show = 1;
    }
    if($show == 1 && $total==1){
    }
    else if($show == 1 && $total!=1){
        $i = 1;
        while($i<=$total){
            echo '<td style="vertical-align:middle;text-
align:center"><a style="font-size:14px;font-family:typo;font-
weight:bold" href="account.php?q=3&show='.$i.'">&nbsp;'.$i.'&nbsp;</a></t
d>';
            $i++;
        }
        echo '<td style="vertical-align:middle;text-
align:center"><a style="font-size:14px;font-family:typo;font-
weight:bold" href="account.php?q=3&show='.$show.'">&nbsp;>&nbsp;</a></td>';
    }
    else if($show != 1 && $show==$total){
        echo '<td style="vertical-align:middle;text-
align:center"><a style="font-size:14px;font-family:typo;font-
weight:bold" href="account.php?q=3&show='.$show-
1.'">&nbsp;<&nbsp;</a></td>';

        $i = 1;
        while($i<=$total){
            echo '<td style="vertical-align:middle;text-
align:center"><a style="font-size:14px;font-family:typo;font-
weight:bold" href="account.php?q=3&show='.$i.'">&nbsp;'.$i.'&nbsp;</a></t
d>';
            $i++;
        }
    }
    else{
        echo '<td style="vertical-align:middle;text-
align:center"><a style="font-size:14px;font-family:typo;font-
weight:bold" href="account.php?q=3&show='.$show-
1.'">&nbsp;<&nbsp;</a></td>';
    }
}

```

```

        $i = 1;
        while($i<=$total){
            echo '<td style="vertical-align:middle;text-align:center"><a style="font-size:14px;font-family:typo;font-weight:bold" href="account.php?q=3&show='.$i.'">&nbsp;'.$i.'&nbsp;</a></td>';
            $i++;
        }
        echo '<td style="vertical-align:middle;text-align:center"><a style="font-size:14px;font-family:typo;font-weight:bold" href="account.php?q=3&show='.$show+1.'">&nbsp;>&nbsp;</a></td>';
    }
    echo '</tr></table></div>';
}
?>
</div></div></div></div>
<div class="row footer">
    <div class="col-md-2 box"></div>
    <div class="col-md-6 box">
        <span href="#" data-target="#login" style="color:lightyellow">Розробник - Маніта Лідія<br><br></span></div>
        <div class="col-md-2 box">
            <a href="feedback.php" style="color:lightyellow;text-decoration:underline" onmouseover="this.style('color:yellow')" target="new">Відгуки</a></div>
        <div class="col-md-2 box">
            <a href="about.php" s style="color:lightyellow;text-decoration:underline" onmouseover="this.style('color:yellow')" target="new">Інфо</a></div>
    </div>
</body>
</html>

```


Додаток Б. Код index.php

```

<?php
session_start();
if(isset($_SESSION['username']) && (!isset($_SESSION['key']))) {
    header('location:account.php?q=1');
}
else if(isset($_SESSION['username']) && isset($_SESSION['key']) && $_SESSION['key'] == '54585c506829293a2d4c3b68543b316e2e7a2d277858545a36362e5f39') {
    header('location:dash.php?q=0');
}
else {}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<link rel="icon" href="favicon.ico" type="image/icon" sizes="16x16">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1">

<title> Tester </title>

<link rel="stylesheet" href="css/main.css">
<link rel="stylesheet" href="css/font.css">
<script src="js/jquery.js" type="text/javascript"></script>
<link rel="stylesheet" href="css/bootstrap.min.css"/>
<link rel="stylesheet" href="css/bootstrap-theme.min.css"/>
<script src="js/bootstrap.min.js" type="text/javascript"></script>
<link href='http://fonts.googleapis.com/css?family=Roboto:400,700,300'
rel='stylesheet' type='text/css'>
<?php
if (@$_GET['w']) {
    echo '<script>alert("'" . @$_GET['w'] . "');</script>';
}
?>
<script>
function validateForm() {
    var y = document.forms["form"]["name"].value;
    if (y == null || y == "") {
        document.getElementById("errorMsg").innerHTML="Name must be filled out.";
        return false;
    }
    var br = document.forms["form"]["branch"].value;
    if (br == "") {
        document.getElementById("errorMsg").innerHTML="Please select your branch";
        return false;
    }
}

```

```

var rn = (document.forms["form"]["rollno"].value).split("/");
/* if (rn.length != 3) {
    document.getElementById("errmsg").innerHTML="Incorrect Rollno. Please enter in the format (BE/10XXX/YY)";
    return false;
}
if((rn[0].length != 2 && rn[0].length != 3) || (rn[0].match(/[A-Z]/g)).length != rn[0].length){
    document.getElementById("errmsg").innerHTML="Incorrect Rollno "+rn[0]+" . Make sure all letters are capital (Ex. 'BE' in BE/10XXX/YY)";
    return false;
}
if(rn[1].length != 5 || (rn[1].match(/[0-9]/g)).length != rn[1].length){
    document.getElementById("errmsg").innerHTML="Incorrect Rollno "+rn[1];
    return false;
}
if(rn[2] != "12" && rn[2] != "13" && rn[2] != "14" && rn[2] != "15" && rn[2] != "16"){
    document.getElementById("errmsg").innerHTML="Incorrect Rollno "+rn[2];
    return false;
}*/
var g = document.forms["form"]["gender"].value;
if (g=="") {
    document.getElementById("errmsg").innerHTML="Будь-ласка оберіть стать";
    return false;
}
var x = document.forms["form"]["username"].value;
if (x.length == 0) {
    document.getElementById("errmsg").innerHTML="Введіть коректний логін";
    return false;
}
if (x.length < 4) {
    document.getElementById("errmsg").innerHTML="Логін повинен мати щонайменше 4 символи";
    return false;
}
var m = document.forms["form"]["phno"].value;
if (m.length != 10) {
    document.getElementById("errmsg").innerHTML="Телефон повинен мати щонайменше 10 цифр";
    return false;
}
var a = document.forms["form"]["password"].value;
if(a == null || a == ""){
    document.getElementById("errmsg").innerHTML="Пароль повинен бути заповнений";
    return false;
}

```

```

    }
    if(a.length<5 || a.length>15){
        document.getElementById("errorMsg").innerHTML="Пароль повинен мати не
менше 5 символів";
        return false;
    }
    var b = document.forms["form"]["cpassword"].value;
    if (a!=b){
        document.getElementById("errorMsg").innerHTML="Паролі повинні співпад
ати";
        return false;
    }
}
</script>
</head>
<body>
<div class="header">
<div class="row">
<div class="col-lg-6">
<span class="logo">Tester</span></div>
<div class="col-md-2 col-md-offset-4">
<a href="#" class="btn btn-primary logb" data-toggle="modal" data-
target="#myModal"> <span class="glyphicon glyphicon-log-in" aria-
hidden="true"></span>&nbsp;<span class="title1"><b> Увійти </b> </span><
/a></div>
<div class="modal fade" id="myModal">
    <div class="modal-dialog">
        <div class="modal-content title1">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal" aria-
label="Закрити"><span aria-hidden="true">&times;</span></button>
                <h4 class="modal-title title1"><span style="color:darkblue;font-
size:12px;font-weight: bold">Увійти до акаунту</span></h4>
            </div>
            <div class="modal-body">
                <form class="form-
horizontal" action="login.php?q=index.php" method="POST">
<fieldset>
<div class="form-group">
    <label class="col-md-3 control-label" for="username"></label>
    <div class="col-md-6">
        <input id="username" name="username" placeholder="Username" class="form
-control input-md" type="username">

    </div>
</div>
</div>
<div class="form-group">
    <label class="col-md-3 control-label" for="password"></label>
    <div class="col-md-6">
        <input id="password" name="password" placeholder="Введіть пароль" cla
ss="form-control input-md" type="password">

```



```

    </div>
</div>

    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
        <button type="submit" class="btn btn-primary">Log in</button>
    </fieldset>
</form>
    </div>
</div>
</div>
</div>
</div>

<div class="bg1">
<div class="row">

<div class="col-md-7"></div>
<div class="col-md-4 panel">
    <form class="form-
horizontal" name="form" action="sign.php?q=account.php" onSubmit="return
validateForm()" method="POST">
<fieldset>
<div class="form-group">
    <label class="col-md-12 control-label" for="name"></label>
    <div class="col-md-12">
        <h3 align="center">Форма реєстрації</h3>

    </div>
</div>
</div>

<div class="form-group">
    <label class="col-md-12 control-label" for="name"></label>
    <div class="col-md-12">
        <div id="errorMsg" style="font-size:14px;font-family:calibri;font-
weight:normal;color:red"><?php
if (@$_GET['q7']) {
    echo '<p style="color:red;font-size:15px;">' . @$_GET['q7'];
}
?></div>

    </div>
</div>
<div class="form-group">
    <label class="col-md-12 control-label" for="name"></label>
    <div class="col-md-12">
        <input id="name" name="name" placeholder="Введіть ім'я" class="form-
control input-md" type="text" value="">?php
echo $_GET['name'];

```

```

?>">

    </div>
</div>
<div class="form-group">
    <label class="col-md-12 control-label" for="rollno"></label>
    <div class="col-md-12">
        <input id="rollno" name="rollno" placeholder="Введіть номер студентсько
ro квитка" class="form-control input-md" type="text" value="<?php
echo $_GET['rollno'];
?>">

    </div>
</div>
<div class="form-group">
    <label class="col-md-12 control-label" for="gender"></label>
    <div class="col-md-12">
        <select id="gender" name="gender" placeholder="Оберіть стать" class="
form-control input-md" >
            <option value="" <?php
if (!isset($_GET['gender']))
    echo "selected";
?>>Оберіть стать</option>
            <option value="M" <?php
if ($_GET['gender'] == "M")
    echo "selected";
?>>Чоловіча</option>
            <option value="F" <?php
if ($_GET['gender'] == "F")
    echo "selected";
?>>Жіноча</option> </select>
    </div>
</div>
<div class="form-group">
    <label class="col-md-12 control-label" for="branch"></label>
    <div class="col-md-12">
        <select id="branch" name="branch" placeholder="Оберіть групу" class="
form-control input-md" >
            <option value="" <?php
if (!isset($_GET['branch']))
    echo "selected";
?>>Оберіть групу</option>
            <option value="CSE" <?php
if ($_GET['branch'] == "CSE")
    echo "selected";
?>>Computer Science and Engineering</option>
            <option value="ECE" <?php
if ($_GET['branch'] == "ECE")
    echo "selected";
?>>Electronics and Communication Engineering</option>
            <option value="EEE" <?php
if ($_GET['branch'] == "EEE")

```



```

        echo "selected";
?>>Electrical and Electronics Engineering</option>
    <option value="IT" <?php
if ($_GET['branch'] == "IT")
    echo "selected";
?>>Information Technology</option>
    <option value="CHEM" <?php
if ($_GET['branch'] == "CHEM")
    echo "selected";
?>>Chemical Engineering</option>
    <option value="CIVIL" <?php
if ($_GET['branch'] == "CIVIL")
    echo "selected";
?>>Civil Engineering</option>
    <option value="MECH" <?php
if ($_GET['branch'] == "MECH")
    echo "selected";
?>>Mechanical Engineering</option>
    <option value="BIOTECH" <?php
if ($_GET['branch'] == "BIOTECH")
    echo "selected";
?>>Biotechnology</option>
    <option value="IMSC" <?php
if ($_GET['branch'] == "IMSC")
    echo "selected";
?>>Integrated MSc</option> </select>

</div>
</div>
<div class="form-group">
    <label class="col-md-12 control-label title1" for="username"></label>
    <div class="col-md-12">
        <input id="username" name="username" placeholder="Оберіть логін" clas
s="form-control input-md" type="username" value="<?php
echo $_GET['username'];
?>" style="<?php
if (isset($_GET['q7']))
    echo "border-color:red";
?>">

    </div>
</div>
<div class="form-group">
    <label class="col-md-12 control-label" for="phno"></label>
    <div class="col-md-12">
        <input id="phno" name="phno" placeholder="Введіть номер телефону" class
="form-control input-md" type="number" value="<?php
echo $_GET['phno'];
?>">

    </div>
</div>

```

```

<div class="form-group">
  <label class="col-md-12 control-label" for="password"></label>
  <div class="col-md-12">
    <input id="password" name="password" placeholder="Введіть пароль" class="form-control input-md" type="password">

  </div>
</div>

<div class="form-group">
  <label class="col-md-12 control-label" for="cpassword"></label>
  <div class="col-md-12">
    <input id="cpassword" name="cpassword" placeholder="Повтор пароля" class="form-control input-md" type="password">

  </div>
</div>
<div class="form-group">
  <label class="col-md-12 control-label" for=""></label>
  <div class="col-md-12" style="text-align: center">
    <input type="submit" value="Зареєструватись" class="btn btn-primary" style="text-align:center" />
  </div>
</div>

</fieldset>
</form>
</div>
</div>
</div>
<div class="row footer">
<div class="col-md-2 box">
<a href="#" data-toggle="modal" data-target="#login" style="color:lightyellow">Увійти як адміністратор</a></div>
<div class="col-md-6 box">
<span href="#" data-target="#login" style="color:lightyellow">Розробник – Маніта Лідія<br><br></span></div>
<div class="col-md-2 box">
<a href="feedback.php" style="color:lightyellow;" onmouseover="this.style('color:yellow')" target="new">Відгуки</a></div>
<div class="col-md-2 box">
<a href="about.php" style="color:lightyellow;" onmouseover="this.style('color:yellow')" target="new">Інфо</a></div>
</div>
  <div class="modal fade" id="login">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">

```

```

        <button type="button" class="close" data-
dismiss="modal"><span aria-hidden="true">&times;</span><span class="sr-
only">Close</span></button>
        <h4 class="modal-title"><span style="color:darkblue;font-
size:12px;font-weight: bold">Увійти до адмін панелі</span></h4>
    </div>
    <div class="modal-body title1">
<div class="row">
<div class="col-md-3"></div>
<div class="col-md-6">
<form role="form" method="post" action="admin.php?q=index.php">
<div class="form-group">
<input type="text" name="uname" maxlength="20" placeholder="Username" cl
ass="form-control"/>
</div>
<div class="form-group">
<input type="password" name="password" maxlength="30" placeholder="Passwo
rd" class="form-control"/>
</div>
<div class="form-group" align="center">
<input type="submit" name="login" value="Login" class="btn btn-
primary" />
</div>
</form>
</div><div class="col-md-3"></div></div>
    </div>
</div>
</body>
</html>

```