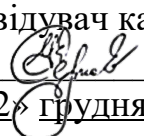


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова
Навчально науковий інститут комп'ютерних наук та управління проектами
Кафедра управління проектами

«Допущений до захисту»
Завідувач кафедри
 Чернов С.К.
«12» грудня 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня вищої освіти «магістр»

на тему:

Удосконалення механізмів управління знаннями в освітніх проектах

Виконав: студент 6171м групи

 Гайченко А.О.

(підпис)

Керівник роботи:

К.Т.Н., ДОЦЕНТ

(посада) науковий ступень вчене звання)

 Чернова Лб.С.

(підпис)

Миколаїв – 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально науковий інститут комп'ютерних наук та управління проєктами

Кафедра управління проєктами
Спеціальність 122 Комп'ютерні науки
Освітня програма «Управління проєктами»

«ЗАТВЕРДЖУЮ»

Гарант освітньої програми
Чернова Л.С.


(підпис)

«29» вересня 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «магістр»

Студенту Гайченко Анастасії Олександрівни

(Прізвище, ім'я, по батькові)

1. Тема роботи: Удосконалення механізмів управління знаннями в освітніх проєктах

Керівник роботи к.т.н., доцент Чернова Любава Сергіївна

Затверджені наказом ректора № 805-уч. від «29» вересня 2022 року

2. Термін подання роботи: 12.12.2022р.

3. Вихідні дані по роботі: удосконалити механізми управління знаннями в освітніх проєктах

4. Перелік питань, що належать до розробки (найменування розділів)

Розділ 1. Теоретичні основи управління проєктами

Розділ 2. Аналіз проєкту освітнього веб-додатку

Розділ 3. Етапи розробки та створення програмного додатку

Розділ 4. Охорона праці

Розділ 5. Охорона навколишнього середовища

5. Перелік презентаційних матеріалів виконаний в програмі Power Point

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ 4. Охорона праці	Гурець Н.В., старший викладач	06.09.2022 <i>Н.Гурець</i>	06.09.2022 <i>Гайченко А.</i>
Розділ 5. Охорона навколишнього середовища	Гурець Н.В., старший викладач	06.09.2022 <i>Н.Гурець</i>	06.09.2022 <i>Гайченко А.</i>

7. Дата видачі завдання 05.09.2022 р.

КАЛЕНДАРНИЙ ПЛАН


№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Вивчення літературних джерел з предмету дослідження	протягом навчання на 5 курсі	виконано
2	Складання розгорнутого плану магістерської роботи та ознайомлення керівника з планом кваліфікаційної роботи	Вересень 2022	виконано
3	Написання розділу 1	Вересень 2022	виконано
4	Написання розділу 2	Жовтень 2022	виконано
5	Написання розділу 3	Листопад 2022	виконано
6	Написання розділів з Охорони праці та навколишнього середовища (4,5)	Листопад 2022	виконано
7	Оформлення магістерської роботи	Грудень 2022	виконано
8	Передача магістерської роботи рецензенту для рецензування	Грудень 2022	виконано
9	Передача магістерської роботи науковому керівникові для написання відгуку	Грудень 2022	виконано
10	Попередній захист магістерської роботи	12.12.2022	виконано
11	Захист магістерської роботи	19.12.2022	виконано

Студент

Гайченко А.
(підпис)

Гайченко А.О.

Керівник роботи


(підпис)

Чернова Л.С.

Анотація

Гайченко А.О.; Удосконалення механізмів управління знаннями в освітніх проектах; 122 «Комп'ютерні науки» освітньої програми «Управління проектами»; Навчально-науковий інститут комп'ютерних наук та управління проектами, Кафедра Управління проектами; к.т.н., доцент Чернова Л. С.; місто Миколаїв; 2022 рік; 86 сторінок.

Предметом розробки є система обліку успіхів учнів на основі бально-рейтингової системи.

Метою роботи є аналіз структурних ефектів та властивостей на основі розробки власної програми, які можуть бути отримані під час виконання роботи.

У даній роботі описано результати проектування та програмної реалізації універсальної системи обліку успішності, яка дозволяє на основі узагальненої уніфікованої математичної моделі сформувати бально-рейтингову систему, яка може задовольнити потреби всіх учасників навчального процесу.

Ключові слова: оцінка успішності навчання, бальна система оцінки, комп'ютерно-навчальні програми, електронний навчальний посібник, освітній простір.

Annotation

A. Haichenko; Improvement of knowledge management mechanisms in educational projects; 122 "Computer Science" of the educational program "Project Management"; Educational and Scientific Institute of Computer Sciences and Project Management, Department of Project Management; Ph.D., associate professor L. Chernova; Mykolaiv; 2022; 86 pages.

The purpose of the work is the analysis of structural effects and properties based on the development of one's own program, which can be obtained during the execution of the work.

This paper describes the results of the design and software implementation of a universal success accounting system, which allows to form a point-rating system based on a generalized unified mathematical model that can meet the needs of all participants in the educational process.

Key words: assessment of learning success, scoring system, computer-learning programs, electronic study guide, educational space.

Зміст

ВСТУП.....	6
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ УПРАВЛІННЯ ПРОЕКТАМИ	9
1.1. ОПИС ПРОБЛЕМИ.....	9
1.2. АНАЛІЗ МІЖНАРОДНОГО ТА ВІТЧИЗНЯНОГО ДОСВІДУ ВИРІШЕННЯ ПРОБЛЕМИ	20
1.3. АНАЛІЗ ОБ’ЄКТУ ДОСЛІДЖЕННЯ.....	26
ВИСНОВКИ ДО РОЗДІЛУ	29
РОЗДІЛ 2. АНАЛІЗ ПРОЕКТУ ОСВІТНЬОГО ВЕБ-ДОДАТКУ	32
2.1. ХАРАКТЕРИСТИКА ПРОЕКТУ	32
2.2. АНАЛІЗ ЗДІЙСНИМОСТІ ПРОЕКТУ	37
2.3. ПЛАНУВАННЯ ТА МОНИТОРИНГ ПРОЕКТУ.....	44
ВИСНОВКИ ДО РОЗДІЛУ	55
РОЗДІЛ 3. ЕТАПИ РОЗРОБКИ ТА СТВОРЕННЯ ПРОГРАМНОГО ДОДАТКУ	57
3.1. ОПИС АЛГОРИТМУ ТА СЕРЕДОВИЩА ДЛЯ РОЗРОБКИ	57
3.2. РОЗРОБКА ТА ОПИС ФУНКЦІОНАЛУ ВЛАСНОГО ПРОДУКТУ	70
3.3. ТЕСТУВАННЯ.....	79
ВИСНОВКИ ДО РОЗДІЛУ	81
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	83
4.1. АНАЛІЗ ПОТЕНЦІЙНОЇ НЕБЕЗПЕКИ	83
4.2. ПРОМИСЛОВА ГІГІЄНА ТА ЗАХОДИ ГІГІЄНИ ПРАЦІ	87
4.3. ЗАХОДИ ПОЖЕЖНОЇ БЕЗПЕКИ.....	93
РОЗДІЛ 5. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	96
5.1. ПОНЯТТЯ ОХОРОНИ НАВКОЛИШНЬОГО ПРИРОДНОГО СЕРЕДОВИЩА, ОБ’ЄКТИ, ОСНОВНІ ПРИНЦИПИ ТА ЗАВДАННЯ	96
5.2. ПРАВА ТА ОБОВ’ЯЗКИ ГРОМАДЯН ТА ОРГАНІВ ДЕРЖАВНОЇ ВЛАДИ ЩОДО ОХОРОНИ НАВКОЛИШНЬОГО ПРИРОДНОГО СЕРЕДОВИЩА.....	98
5.3. ОСНОВНІ ПРИНЦИПИ ОХОРОНИ НАВКОЛИШНЬОГО СЕРЕДОВИЩА.....	102
ВИСНОВКИ.....	104
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	106
ДОДАТКИ.....	109
ДОДАТОК А(DB.CS).....	109
ДОДАТОК Б(FORM1.CS).....	110

ВСТУП

Актуальність обраної теми полягає у необхідності осмислення сучасної освіти, оскільки будь-який ресурс, опублікований у Всесвітній мережі, від глобального інформаційного порталу до невеликої домашньої сторінки, безпосередньо пов'язаний із інженерно-дизайнерським рішенням.

В освітній практиці України поняття проект було започатковано лише в останнє десятиліття і ще продовжує набирати оберти та досліджуватися. Ідейним натхненником для написання цієї роботи стала реалізація освітнього проекту "Педагогічний експеримент" (автор Л. Ващенко), де автор започатковував науково-організаційний супровід інноваційних педагогічних програм; створив систему підготовки освітян-практиків до інноваційної педагогічної діяльності; розширював мережу навчальних закладів інноваційного типу діяльності та взаємозв'язку освіти та візуалізації.

В наш час тренди дизайну тісно пов'язані з не тільки візуальною складовою, що представляє бренд, але й досвідом користування користувачів ресурсу, їх задоволеності зрозумілим та лаконічним дизайном, інтуїтивними налаштуваннями. Все це робить роботу програміста важливою, оскільки вимагає врахування поєднання візуальної ідентифікації та програмного забезпечення, використовуючи та проектуючи стильову модель для кожного проекту окремо.

Мета роботи – аналіз структурних ефектів та властивостей на основі розробки власної програми, які можуть бути отримані під час виконання роботи.

Для реалізації мети дослідження було поставлено такі *завдання*:

- аналіз актуальної теоретичної бази на тему розробка віртуального навчального посібника;
- визначення основних етапів розробки програми;

- оцінка стильових моделей програм;
- розробка власного дизайну з урахуванням функціональності та стильового наповнення.

Об'єктом дослідження є процес розробки програми.

Предметом дослідження є процес проектування стильових образів та елементів продукту.

Серед *основних методів дослідження* у цій роботі були використані метод аналізу, синтезу та моделювання. Метод аналізу був використаний для наявної інформації та попередніх досліджень. Крім того, інформація зібрана та проаналізована, надалі була синтезована у загальну структуру розуміння дизайну як стильової моделі. Моделювання свого застосування дозволило підтвердити твердження про використання стильової моделі як засобу візуальної ідентифікації.

Наукова новизна одержаних результатів полягає в тому, що в результаті проведення дослідження підтверджено залежність структури додатка, стильової моделі як засобу візуальної ідентифікації ресурсу та позитивного досвіду користування цільовою аудиторією.

Практичне значення одержаних результатів у тому, що у межах цієї роботи розроблено структури докладання, відмінні між собою. Крім того, проаналізовано та систематизовано теоретичну інформацію про існуючі підходи до розуміння додатків. Отримані результати властивостей додатків допоможуть прогнозувати попит на додатки на етапі його проектування та враховувати подальший досвід використання ресурсів.

Ключові слова: оцінка успішності навчання, бальна система оцінки, комп'ютерно-навчальні програми, електронний навчальний посібник, освітній простір.

Перелік умовних скорочень: КНП - комп'ютерно-навчальні програми; СДН - система дистанційного навчання; ВНЗ- вищий навчальний заклад, DOS- дискова операційна система.

Магістерська робота складається із вступу, п'яти розділів із висновками, загальних висновків та списку використаної літератури, а також додатків. Основна частина роботи, викладена на 65 сторінках друкованого тексту, містить 15 малюнків. Список використаної літератури із 31 найменування представлений на 3 сторінках. Повний обсяг роботи складає 89 сторінки.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ УПРАВЛІННЯ ПРОЕКТАМИ

1.1. Опис проблеми

Поняття "проект" в останні роки активно увійшло у практику освіти. Воно виступає як в ролі ефективного методу навчання учнів, так і в якості форми організації інноваційно-розвивального освітнього середовища навчального закладу чи всієї регіональної системи освіти. [1]

До найважливіших відмінностей роботи над проектом від інших видів діяльності належить обмеженість у часі, унікальність, необхідність заздалегідь координувати багатопланові дії.

Обмеженість у часі означає, що будь-який проект має момент його початку та його завершення. Завершується проект після досягнення намічених цілей. У регулярно відновлюваної діяльності немає заздалегідь обумовленого терміну завершення. Унікальність та неповторність результату проекту означає, що в умовах, у складі, у порядку необхідних робіт є щось нове. Отже, неможливо гарантовано отримати те, що потрібно шляхом простого копіювання раніше виконаної роботи. Регулярно відновлювана діяльність ефективна, якщо потрібно відтворити неодноразово отриманий в аналогічних умовах результат.

На початку кожного проекту через його унікальність доводиться наново визначати склад майбутніх робіт, їх порядок та виконавців. Так як у кожному проекті є щось нове, то на етапі планування необхідно приділяти увагу узгодженню навіть, начебто, звичних процедур. Діяльність, що стала регулярною, реалізується за вже відпрацьованим планом.

Результативне виконання проекту протягом усього часу його існування забезпечується за допомогою процесів управління проектом, які можуть бути згруповані у наступні п'ять етапів управління проектом: ініціація, планування, виконання, моніторинг та контроль, завершення проекту.

Головний продукт ініціації – викладений у формі Статуту параметричний опис концепції проекту, обґрунтування необхідності та реалістичності проекту та формальна авторизація проекту.

Ініціація проекту починається із визначення проектної ідеї – основного задуму проекту. З урахуванням наміченої ідеї на укрупненому вигляді розробляється приблизний план її реалізації. За його допомогою проводиться попередня оцінка необхідних ресурсів. Завершенням цього етапу стає оцінка життєздатності проекту та ухвалення рішення про те, чи варто працювати над ним далі.

Планування проекту спрямоване на розробку плану проекту, в якому визначено та документовано всі дії, необхідні для досягнення цілей проекту. План проекту необхідний координації діяльності всіх учасників проекту. Цілісний план визначає порядок виконання всіх робіт. Він описує, що, хто, як і коли робитиме. Початком цього етапу стає визначення системи логічно пов'язаних цілей. Далі визначаються переліки та взаємозв'язки необхідних робіт. Потім кожної роботи визначаються необхідні ресурси. У результаті планування розробляється організаційна структура проекту, описується порядок взаємодії його. Наприкінці успішно пройденого етапу планування створюються робочі документи. Вони містяться прийняті до виконання мети і завдання проекту, логічно ув'язана система майбутніх робіт, календарні плани-графіки робіт, бюджет, опис системи управління проектом.

Виконання проекту спрямовано безпосереднє створення товару (послуги), заради якого реалізується даний проект. Спочатку набирається команда проекту. Проводиться інструктаж команди, запуск її роботи за ухваленим планом. У ході виконання робіт, запланованих у плані проекту, йде налагодження командної взаємодії, реалізуються заходи щодо усунення неприпустимого відхилення від планових показників. Головним у виконанні проекту стає ефективне досягнення намічених цілей. Це означає, що

практичні кроки, згідно із затвердженим планом, мають забезпечити досягнення у встановлені терміни якісних результатів за рахунок використання виділених ресурсів.

Моніторинг та контроль націлені на забезпечення виконання проекту відповідно до його плану, відстеження певних параметрів проекту, які мають зберігатися в заданих межах, а також на своєчасне виявлення можливих проблем та відхилень, прийняття рішень у відповідь на зміни, що виникають у проекті, контроль якості. У ході контролю проводиться перевірка результату на ступінь його відповідності до планових показників. За підсумками такої перевірки може бути прийняте рішення або доведення якості продукту або послуги до необхідного рівня, або про завершення проекту.

Завершення проекту включає процедури щодо складання підсумкових звітів, проведення остаточних взаєморозрахунків між учасниками проекту. На цьому етапі можуть укладатися договори про додаткові послуги. Окремою частиною завершального етапу може стати узагальнення та опис накопиченого досвіду. Важливою складовою успішного завершення проекту є передача виробленого продукту (послуги) тому користувачеві, якому він і призначався.

Успішність проекту по-різному оцінюється в різних методиках. Успішність може по-різному оцінюватися різними учасниками проекту.

Групи оцінок успішності:

Орієнтовані на контракт із жорсткою фіксацією вимог та мінімізацією змін у ході проекту, наприклад, традиційні методології, у тому числі РМВОК: «проект успішний, якщо виконаний відповідно до затверджених критеріїв: обсягу, терміну, якості». Тобто проект успішний, якщо виконано та закрито договір між Замовником та Виконавцем (незалежно від того, чи був він юридичним документом у разі зовнішніх проектів чи визначався

якось інакше у разі внутрішніх проектів). При цьому оцінка успішності єдина як для замовника, так і для виконавця.

Орієнтовані на задоволеність замовника з гнучким керуванням вимогами, наприклад, гнучкі методології SCRUM: «проект успішний, якщо замовник задоволений»

Орієнтовані на тривалу взаємодію із Замовником: керування програмами, спрямоване на тривалу взаємодію, а не на один проект/контракт. Тут наголошується на продовженні співпраці Виконавця із Замовником у рамках подальших проектів та іншої взаємодії.

Збалансовані, наприклад PRINCE2: «проект успішний при збалансованості принаймні за трьома категоріями — бізнесу, орієнтації на користувача та технологічної зрілості». Тут наголошується на фінансовій успішності проекту, задоволеності користувачів та розвитку технологій. Оцінка успішності може різнитися з погляду бізнесу, користувача та виконавця. Такі методики оцінки найчастіше використовуються для внутрішніх проектів, коли замовник та виконавець знаходяться в одній організації.

Учасники проекту (Project Stakeholders) – фізичні особи та організації, які безпосередньо залучені до проекту, чи інтереси яких можуть бути порушені при виконанні проекту.

Завдання команди управління проектом полягає в тому, щоб визначити учасників проекту та їх вимоги та, наскільки це можливо, керувати цими вимогами, щоб забезпечити успішне завершення проекту.

Учасники проекту мають різні рівні відповідальності та повноважень за участю у проекті, причому відповідальність та повноваження можуть змінюватись на різних етапах життєвого циклу проекту.

Учасники можуть позитивно чи негативно впливати на проект. Позитивно впливають учасники – це ті, кому вигідно успішне завершення

проекту, тоді як учасникам, що негативно впливають, успішне завершення проекту є небажаним.

Основні ролі учасників проекту:

Ініціатор – автор головної ідеї проекту, його попереднього обґрунтування та пропозицій щодо здійснення проекту. Як ініціатор може виступати практично будь-хто з майбутніх учасників проекту.

Замовник – особа, зацікавлена у здійсненні проекту та досягненні його цілей. Майбутній власник результатів проекту. Замовник визначає основні вимоги до проекту.

Клієнт, користувач – особа, яка використовуватиме результат проекту.

Виконуюча організація – підприємство, співробітники якого безпосередньо беруть участь у виконанні проекту.

Спонсор – особа, яка забезпечує фінансові, матеріальні, людські та інші ресурси для виконання проекту.

Інвестор – особа, яка вкладає інвестиції у проект, наприклад, у вигляді кредитів. Ціль інвесторів – максимізація прибутку від своїх інвестицій при реалізації проекту.

Менеджер проекту – фізична особа, якій делегуються повноваження з керівництва усіма роботами проекту. Він є відповідальним за успішне виконання проекту.

Команда проекту – учасники проекту, об'єднані за цільовим принципом реалізації проекту. Створюється на період проекту.

Основи планування та підготовка до складання плану в ms Project

Мета роботи:

План проекту призначений для того, щоб визначити, за допомогою яких робіт досягатиметься результат проекту, які люди та обладнання потрібні для виконання цих робіт, у який час ці люди та обладнання будуть зайняті роботою над проектом. Тому проектний план містить три основні елементи: завдання (task), ресурси (resource) та призначення (assignment).

Завданням називається робота, що здійснюється в рамках проекту для досягнення певного результату. Оскільки зазвичай проект містить багато завдань, то зручності відстеження плану їх об'єднують у групи, чи фази. Сукупність фаз проекту називається його життєвим циклом.

Фаза проекту складається з одного або кількох завдань, в результаті виконання яких досягається один або декілька основних результатів проекту. Таким чином, результати, досягнуті завдяки виконанню кожної із завдань, що входять у фазу, формують її результат.

Якщо досягнення результатів завдання досить виконати лише її, то досягнення результату фази потрібно виконати групу інших завдань. І це полягає відмінність фази від завдання: її результат підсумовує результати інших завдань. Саме тому MS Project фази називають сумарними завданнями (summary task).

Проект розбивається на фази та для зручності відстеження ходу робіт. Після завершення проектної фази зазвичай здійснюється аналіз як отриманих результатів, щоб з мінімальними витратами визначити та виправити помилки, так і загального ходу виконання проекту, щоб визначити, чи варто переходити до його наступної фази.

Розбиття проекту на фази дозволяє подати його у вигляді списку основних результатів та дат, до яких ці результати мають бути отримані. Керівник проекту здійснює безпосередній контроль за виконанням кожного завдання всередині проекту, повідомляючи вищому менеджеру тільки про досягнення фазових результатів. Цьому менеджеру, у свою чергу, для контролю за виконанням проекту цілком достатньо таких даних.

Щоб покращити оволодіння новими видами діяльності, знаннями та навичками, щороку в освіті з'являються сотні комп'ютерних навчальних програм (КНП), які успішно використовуються як навчальні засоби та допомога у викладанні та вивченні конкретних дисциплін. Це досягається

швидкістю обчислень, візуальним відображенням і збереженням результатів, швидкістю введення і видачі інформації, активізацією зорового мислення і зорової пам'яті учнів, можливістю нетрадиційного постановки завдань і своєчасного їх виконання. Комп'ютери можуть бути успішно використані на всіх етапах навчальної програми: вони суттєво впливають на контрольні-оцінювальні функції, надають їй цікавих характеристик, сприяють активізації навчально-пізнавальної діяльності учнів.

Комп'ютерні технології дозволяють вивести поданий матеріал на якісно вищий рівень наочності, значно розширити можливості для включення різноманітних вправ у навчальний процес, зробити навчання більш активним за допомогою безперервного зворотного зв'язку, підкріпленого продуманим стимулюючим процесом навчання, що сприяє збільшенню його динамічності і в кінцевому підсумку призводить до неохочих досягнень, які не є основною метою власне процесуального аспекту навчання – формування позитивного ставлення студента до матеріалу, що вивчається, інтересу до нього, задоволення ним у результатах кожного локального етапу навчання.

Сучасне комп'ютерне навчання внесло в його зміст нові елементи, не змінивши традиційної структури навчально-пізнавальної діяльності. Вчителі сприймають комп'ютер як потужний засіб навчання і намагаються використовувати його в навчальному процесі. Але у розробників КНП було багато проблем: від неможливості використовувати певні методи подання інформації до складності програмування навчальних завдань. [2] Тому, мріючи створити універсальний інструмент, вони часто обмежуються поетапним впровадженням програмних продуктів у навчальний процес. Зазвичай це пов'язано з відсутністю необхідної інформації про можливості КНП.

Електронний навчальний посібник — програмно-методичний навчальний комплекс, який відповідає типовому навчальному плану та дає

можливість студенту самостійно або за допомогою викладача опанувати навчальний курс або його розділ. Цей продукт створено із вбудованою структурою, словниками, можливостями пошуку тощо.[2]

Електронний навчальний посібник може бути призначений для самостійного вивчення навчального матеріалу з певної дисципліни або підтримки курсу лекцій з метою його поглиблення. вивчення. набраний ним (або навіть студентами з метою відтворення гарного реферату у великій кількості екземплярів за мінімальних витрат) і розміщений на студентському сервері або іншому загальнодоступному електронному сайті.

Основними ознаками та перевагами електронного підручника є:

1. Можливість побудувати простий та зручний механізм навігації в електронному підручнику.
2. Розробка пошукової системи в межах електронного підручника, зокрема, при використанні гіпертекстового формату видання.
3. Можливість вбудованого автоматизованого контролю рівня знань студента.
4. Можливість особливого варіанта структурування матеріалу.
5. Можливість адаптації вивченого матеріалу підручника до рівня знань учня, що призводить до різкого підвищення рівня мотивації учня.
6. Здатність адаптувати і оптимізувати інтерфейс користувача під індивідуальні потреби студента.

До додаткових можливостей електронного підручника, порівняно з друкованим текстом, можна віднести:

1. Можливість включати спеціальні фрагменти, що імітують перебіг багатьох фізико-технологічних процесів.
2. Можливість включати аудіо-файли до підручника, зокрема наближати процес роботи з підручником та слухати лекції викладача.
3. Вміння включати до підручника фрагменти відеофільмів для ілюстрації окремих положень підручника.

4. Включення інтерактивних фрагментів до підручника задля забезпечення оперативного діалогу з учнем.

5. Повномасштабне мультимедійне оформлення підручника, що включає діалог природною мовою, організацію відео-конференції з авторами та консультантами за бажанням студента тощо.

Таким чином, крім різних носіїв, електронний підручник має низку принципових відмінностей від підручника, виготовленого друкарським методом:

- можливість мультимедіа;
- забезпечення віртуальної реальності;
- високий ступінь інтерактивності;
- можливість індивідуального підходу до учня.

Введення мультимедійних елементів у структуру електронного посібника дозволяє одночасно передавати різні види інформації. Зазвичай це означає поєднання тексту, звуку, графіки, анімації та відео. Засоби наочної демонстрації дозволяють покращити сприйняття нового матеріалу, включити у процес запам'ятовування як слухові, а й зорові центри. Багато процесів і об'єктів в електронному підручнику можна у динаміці їх розвитку, і навіть як 2-D чи 3-D моделей, що створює в користувача ілюзію реальності зображуваних об'єктів. Інтерактивність дозволяє встановити зворотний зв'язок від користувача інформації (учня) до її джерела (викладача). Інтерактивна взаємодія характеризується негайною відповіддю та візуально підтвердженою реакцією на дію чи повідомлення.

Таким чином, електронні посібники мають велику практичну цінність. З їхньою допомогою можна не лише повідомити фактичну інформацію, надану ілюстративним матеріалом, а й наочно продемонструвати деякі процеси, які неможливо показати за допомогою стандартних методів навчання. Крім того, учень може самостійно, без допомоги викладача чи

наукового керівника користуватися електронним посібником, знаходячи відповіді на свої запитання.

Також важливою важливістю електронних посібників є те, що викладач може швидко доповнювати та змінювати текстовий чи ілюстративний матеріал, коли виникає потреба, що дуже важливо для динамічно-змінних дисциплін.

У сучасному освітньому просторі часто використовується так звана система дистанційного навчання (СДН). Таке навчання грає важливу роль модернізації освіти.[3]

Існує безліч визначень цього поняття. Основні сучасні визначення такі: дистанційне навчання є самостійною формою навчання, а провідним інструментом у ньому стають інформаційні технології. Система дистанційного навчання, таким чином, є сукупністю організаційних, телекомунікаційних, педагогічних та наукових ресурсів, що реалізують освітні програми за допомогою дистанційних технологій.[4] Цей програмний комплекс призначений для розробки електронних курсів та організації електронного навчання; сьогодні є складовою віртуального освітнього середовища багатьох навчальних закладів.

У положеннях про електронне навчання та використання технологій дистанційного навчання у вузах СДН визначається таким чином:

«СДН – це електронне інформаційно-освітнє середовище у вигляді системно організованого комплексу інформаційно-комунікаційних засобів і технологій, процесів організаційно-методичного забезпечення, діяльності педагогічного колективу, орієнтованого впровадження системи підтримки навчального процесу з метою задовольнити освітні потреби учнів».[5]

Система дистанційного навчання (СДН) – це взаємодія викладача та студентів один з одним на відстані, що відображає всі складові, властиві навчальному процесу (цілі, зміст, методи, організаційні форми, засоби

навчання) та реалізована специфічними засобами навчального процесу. Інтернет-технології або інші засоби, що забезпечують інтерактивність [6].

Систему дистанційного навчання можна охарактеризувати як комплексну, що охоплює широкий спектр програмно-апаратних комунікацій, методичних вказівок та заходів щодо забезпечення студентів навчальною інформацією. Це також включає проведення необхідної перевірки набутих знань. Для цього використовуються комп'ютерні мережі [5].

Основними принципами СДН є налагодження інтерактивного спілкування між викладачем та студентом без забезпечення безпосередньої зустрічі, а також самостійний розвиток певного обсягу знань та навичок за певним курсом та його програмою. Важливо відзначити, що СДН принципово відрізняється від традиційної освіти тим, що створює нове освітнє інформаційне середовище, в яке приходить студент, який точно розуміє, які знання та вміння йому потрібні. Відмінною особливістю СДН є надання студентам можливості самостійно отримувати необхідні знання, використовуючи розроблені інформаційні ресурси (бази даних, бази знань, комп'ютерні системи, включаючи мультимедіа, електронні бібліотеки, а також традиційні підручники та навчальні посібники).

Серед відмінностей СДН від традиційного навчання є низка психолого-педагогічних проблем, які доводиться вирішувати викладачеві та студентам у СДН:

- проблеми у встановленні міжособистісних контактів;
- проблеми формування ефективно працюючих малих навчальних груп під час навчання.[6]

Таким чином, СДН є невід'ємною частиною як денної, так і дистанційної форми навчання, а також може виступати як самостійна форма навчання.

Із вищесказаного, задача полягає наступним чином:

За допомогою C# створити програму розробити форму і структуру форми для проектування інформаційної системи. Програма має бути адаптивною під користувача. Необхідно передбачити перевірку правильності даних. Також необхідно реалізувати та передбачити всілякі поля для реалізації форми, вказавши необхідні поля для заповнення форми. Додати допоміжні розділи в додатку створивши тим самим структуру (наприклад, вітальна сторінка, сторінка з формою реєстрації, сторінка, що відкривається при вдалому заповненні форми, сторінка контакти), і зв'язати ці сторінки між собою за допомогою посилань або кнопок.

1.2. Аналіз міжнародного та вітчизняного досвіду вирішення проблеми

Теоретико-прикладні підходи до проектної діяльності досліджуються в роботах вітчизняних та зарубіжних учених (Єрмаков І., Підласий І., Воропаєв В., Поздняков В., Шапіро В. та ін.), проте управління освітніми проектами залишається малодослідженою проблемою, оскільки проектний менеджмент був прерогативою тих галузей наукового і практичного знання, що належали до системи матеріального виробництва.

Одним із завдань навчання є забезпечення керованості результатів навчання. Більшість експертів колишнього пострадянського простору вказують на необхідність використання тестів для контролю за засвоєнням знань учнями.

При цьому деякі з них все ще сприймають тестування як неякісний та упереджений спосіб перевірки знань.

Ці твердження ґрунтуються на певних причинах. Згідно з опитуванням, чверть викладачів ВНЗ колишнього пострадянського простору віддають перевагу традиційним розмовам зі студентами. Це можна пояснити, наприклад, тим, що середній вік вчителів перевищує 50 років. Інша причина

пов'язана з відмінностями в тому, як викладачі та експерти створюють системи контролю знань.

Контроль засвоєння знань за допомогою комп'ютерних програм дозволяє:

- визначити рівень учнів та засвоєння ними навчальних матеріалів;
- визначити свій рівень теоретичних знань і вчителі – отримують інформацію щодо досягнення цілей навчання, удосконалення викладання із суміжних предметів тощо.

Важливою перевагою тесту є те, що він знімає суб'єктивність тестувальника, дає можливість своєчасно отримати об'єктивну інформацію тощо.

Застосування активних методів навчання вимагає великої початкової розумової роботи вчителя з підготовки засобів навчання, формування завдань, контрольних запитань і відповідей на них. Очевидно, що неможливо пройти тест на введення запитань, це потрібно зробити через варіанти відповідей.[7]

Найпоширеніший тест, із закритими відповідями. Їхні недоліки добре відомі.

Щодо факультативного типу відповіді, то В. І. Карлащук, автор книги «Підручник», стверджує, що правильна відповідь може бути дана тоді, коли учень впізнає його серед інших, тобто він може не знати відповіді на поставлене питання, але впізнав її. Очевидно, що студенти продемонстрували меншу розумову активність у випадку впізнавання, ніж у випадку пригадування, не кажучи вже про міркування. Також можна випадковим чином вгадати правильну відповідь. Через цей та ряд інших недоліків деякі вчителі часто негативно ставляться до комп'ютерного контролю вибіркового відповідей. Проте, з точки зору В. І. Карлащука, ці

недоліки можна мінімізувати шляхом вмілого написання керуючих програм.
[8]

Відповіді у відкритій (проектній) формі тепер вітаються для проходження тесту, а коли студент вводить відповідь з клавіатури, програма порівнює її з довідковими даними в базі даних і виставляє відповідну оцінку.

Для авторів тестів ця форма мало чим відрізняється від попередньої, тому така система використовується рідко. Начебто інтерактивні розмови зі студентами шляхом ускладнення процедур. Водночас виконання роботи потребує від учителя більших зусиль, обмірковування всіх можливих відповідей тощо. Тому неможливо переконатися, що відповіді кандидатів, визначених програмою, дійсно є неправильними.

Необхідно знайти компромісне вирішення цієї проблеми.

Однією з найважливіших завдань розробки ресурсу є реалізація його функції, як комунікативність. Дизайн програми повинен бути таким, щоб користувач, який опинився тут за своїм запитом, не тільки не залишив його, але захотів би більш детально ознайомитися зі змістом ресурсу та іншою пропонованою ним інформацією [9].

Додаток не зможе довго та успішно функціонувати, якщо на кожному етапі розробки його дизайн не гармоніює з його змістом. Розробка дизайну програми - це створення прямого та рівного шляху між відвідувачем сайту та його метою, якою є дані, що розшукуються. Простий та наочний доступ до інформації – це основна мета грамотного дизайну, в якому, звичайно, можуть використовуватися і яскраві кольорові та графічні рішення, якщо вони не заважають сприйняттю змісту програми.

Після запровадження карантину з проблемою зіштовхнулися усі навчальні заклади. Хоча кожен заклад і мав онлайн майданчик для ведення пар, але в більшості випадків вони мали не найкращий функціонал і не зручну структуру. Хорошим прикладом цієї проблеми є корпоративна

платформа дистанційного навчання Київського національного торговельно-економічного університету (рисунок 1.1).

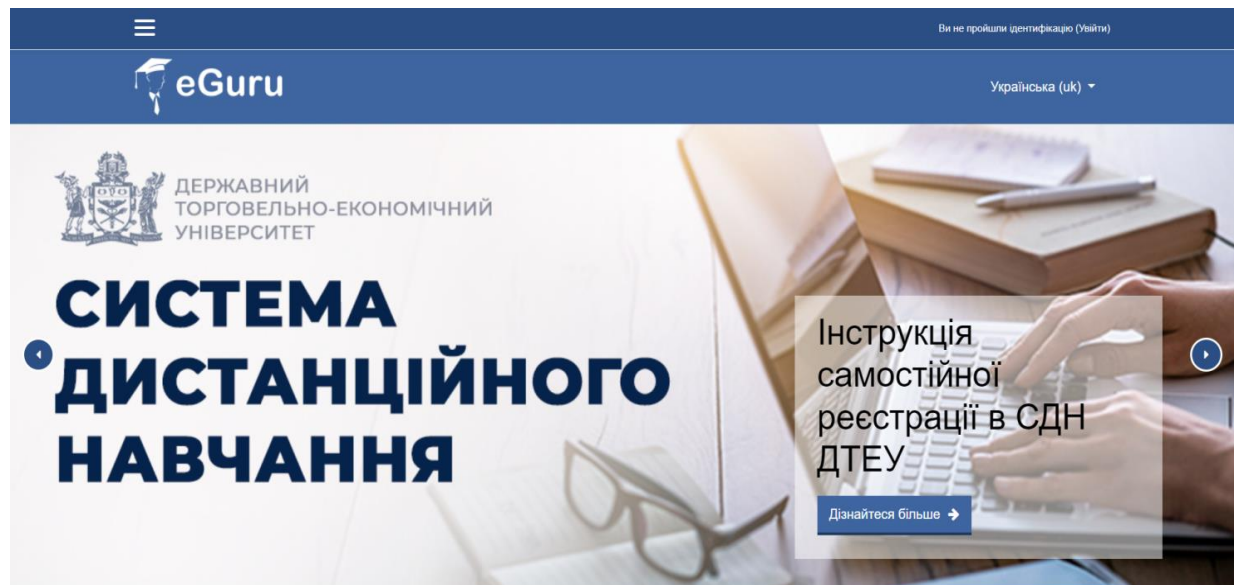


Рисунок 1.1. Вид корпоративної платформи дистанційного навчання Київського національного торговельно-економічного університету

Основною проблемою даної платформи стала нестабільність роботи і не зручна структура. Через проблему нестабільної роботи цей сайт часто не працював чи працював із проблемами. Через незручну структуру даної платформи студенти не могли знайти потрібні курси або лекційний матеріал. Вирішенням цієї проблеми стало використання інших навчально-методичних платформ.

Доступними альтернативами корпоративної платформи дистанційного навчання Київського національного торговельно-економічного університету стали такі платформи Google Клас та Microsoft Teams. Чи не кращий функціонал і не зручна структура. Гарним прикладом є Google Клас.

Google Клас – безкоштовний веб-сервіс, розроблений Google для шкіл, який покликаний спростити створення, розповсюдження та оцінку завдань безпаперовим способом (рисунок 1.2). Основна мета Google Клас –

спростити процес обміну файлами між вчителями та учнями. Основною перевагою для викладачів над конкурентами стала можливість зручності створення класів, зручне проведення контрольних/іспитів, нагляд за виконанням домашніх завдань та автоматичний розрахунок балів. Для студентів перевагами стали зручна по проста структура сайту, доступ до завдань у будь-який час та стабільність роботи ресурсу.

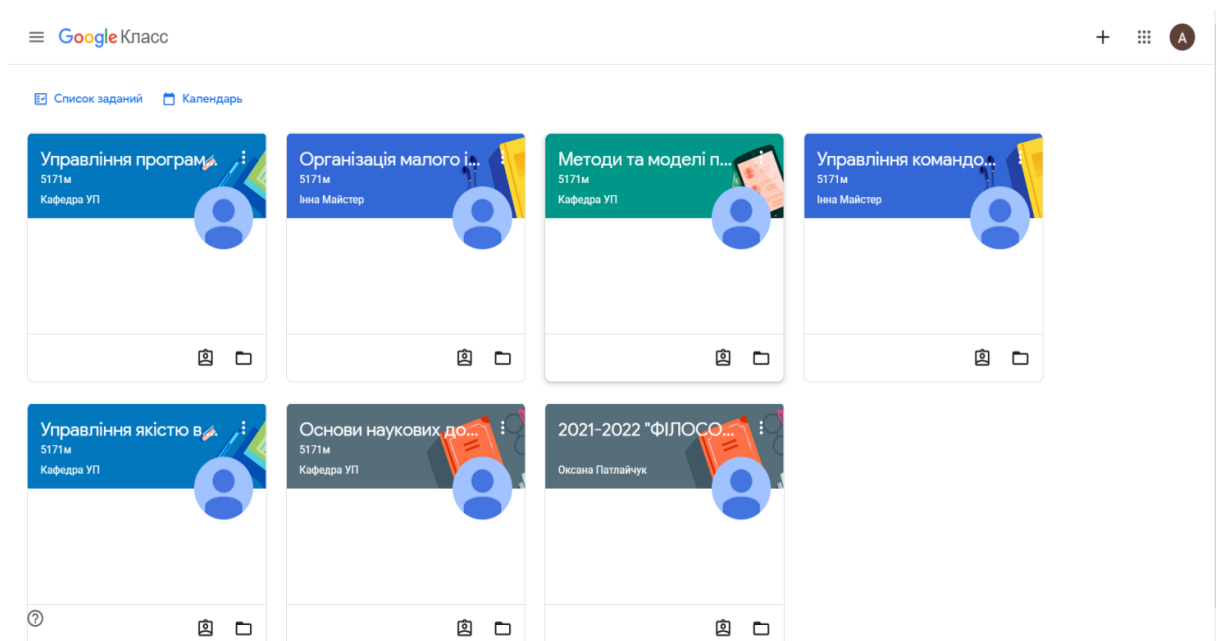


Рисунок 1.2. Вигляд корпоративно-навчальної платформи Google Клас

Microsoft Teams – це корпоративна платформа, яка об'єднує в робочому просторі чат, зустрічі, нотатки та вкладення (Рисунок 1.3). Переваги платформи є такими ж, як у Google Клас, але через перебої в роботі сайту багато викладачів від нього відмовлялися.

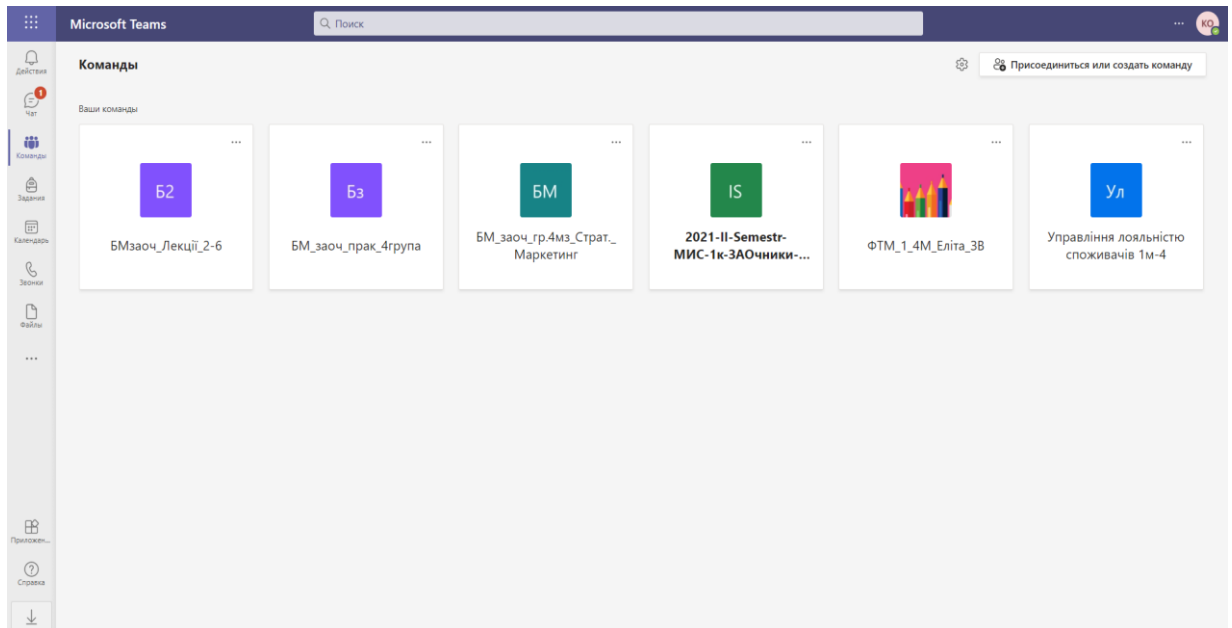


Рисунок 1.3. Вид корпоративно-навчальної платформи Microsoft Teams

Головною причиною переходу викладачів на платформи Google Клас та Microsoft Teams стала універсальність їхнього «Веб-дизайну» та архітектури. Універсальний дизайн є економічно ефективним підходом, оскільки задовольняє потреби всіх користувачів на початковому етапі розробки та проектування.

При розробці електронних освітніх ресурсів подання навчального матеріалу має бути з урахуванням низки чинників: психологічні засади засвоєння навчального матеріалу, його дидактичні особливості, естетика візуальних форм тощо [10].

Основна мета корпоративно-навчальних платформ полягає у сприянні більш ефективному та якісному засвоєнню навчального матеріалу при його екранному поданні не завжди враховуються засобами мультимедіа.

Зважаючи на високу конкуренцію в новому електронному середовищі та зростаючий інтерес електронного бізнесу до розробки веб-сайтів, що максимально задовольняють користувачів, це дослідження пропонує низку керівних принципів, щоб запропонувати адекватну структуру та дизайн веб-сайтів. У цьому рядку огляд літератури дозволяє констатувати важливість

веб-дизайну, щоб отримати позитивні результати в контексті електронної комерції.

1.3. Аналіз об'єкту дослідження

Перед початком розробки КНП(комп'ютерно-навчальна програма) необхідно ознайомитися з продуктами, що вже є на ринку, і оцінити їх придатність до використання при вивченні тієї чи іншої навчальної дисципліни. Аналіз існуючого КНП дозволить більш ясно уявити, які дидактичні завдання необхідно реалізувати, на якому програмно-технічному рівні та за допомогою якої технології розробки необхідно реалізувати керівництво.

Проведено аналіз Інтернет-джерел для виявлення подібних електронних лабораторних практикумів та КНП на тему. На даний момент в Інтернеті відсутні електронні лабораторні майстерні та КНП на тему «Прилади та автомати для контролю точності та якості», які є у продажу або у відкритому доступі[11]. Усі матеріали, доступні на існуючих сайтах (електронних освітніх ресурсах), присвячені лише окремим питанням зазначеної тематики. Пропонується виключно стандартний формат надання інформації: текстово-графічні дані, подані у вигляді підручника або методичні рекомендації, які є електронною копією паперової версії такого інформаційного продукту.

Вважаю, що створення ЄС «Пристрої та автоматичні пристрої для контролю точності та якості» є доцільним, оскільки дозволить викладати матеріал у тісному зв'язку з теорією та практикою, одночасно ввівши елемент гіпертексту, збільшити обсяг графічних даних. та підвищити їхню якість, дозволить запровадити блок самоконтролю, необхідний для навчання студентів дистанційного навчання.

Таким чином, створення та використання електронних підручників та посібників є одним із головних завдань підвищення рівня освіти. КНП може виступати як допоміжний засіб для вчителя при розробці та закріпленні понятійного апарату, може виступати як елемент дидактичного комплексу для організації самостійної роботи учнів.

При створенні автоматизованої системи до неї висувалися такі вимоги: забезпечити інформаційну безпеку та розділення доступу до системи для викладачів і студентів.

Щоб обмежити несанкціоноване використання програми, необхідно розділити права користувача. Користувачами програмних продуктів є викладачі та студенти.

Викладачі-користувачі мають право переглядати вміст бази результатів тестування, створювати, редагувати та видаляти вміст бази завдань і бази відповідей.

Студенти-користувачі мають право використовувати програму лише в режимі контролю знань і навчання.

Тестові дані зберігаються в базі даних результатів тестування.

Інтуїтивно зрозумілий графічний інтерфейс, заснований на ергономіці та психології.

Інтерфейс КНП – включає спілкування студентів з програмою. Організація інтерфейсу є дуже важливим споживчим атрибутом програмних продуктів, що використовуються в навчальних цілях. Передбачається, що програмою користуються недосвідчені користувачі, і кількість сеансів роботи з програмою зазвичай відносно невелика, тому швидкість і легкість навчання керуванню програмою є особливо важливими. Тому необхідно надати можливість зосередитися на досліджуваному предметі і якомога менше думати про спосіб спілкування з комп'ютером. Крім того, інтерфейс повинен відображати специфіку дисципліни, що вивчається, особливо мову,

яку використовує дисципліна, щоб вона була повністю відображена та сприйнята.

Характеристики якісного інтерфейсу включають такі властивості:

- легкий для перегляду, логічно організований текст на екрані;
- виводити на екран тільки необхідну інформацію;
- використовувати графічну інформацію для пояснення усних виступів;
- надихає на чергування стилів дизайну.

Після завантаження ПК він знаходиться в реальному режимі, де адресація обмежена 1 МБ і немає захисту вашого коду. Це просто у режимі дискової операційної системи (DOS). Щоб завантажувати та запускати великі програми та мати доступ до більшої пам'яті, потрібно навчитися переходити в захищений режим за допомогою інструкцій на мові асемблера. Однак усі виклики базової системи вводу/виводу (BIOS) доступні лише у реальному режимі. Якщо ви хочете використовувати BIOS для введення-виводу, необхідно мати механізми перемикавання із захищеного режиму на реальний для виконання викликів BIOS[12]. Таким чином, в програма на C# або якась керуюча програма повинні виконувати це перемикавання, прозоре для користувача. Якщо ви плануєте використовувати програмні переривання замість переривань BIOS, вам потрібно написати власний код на складання для вирішення всіх операцій вводу-виводу. Ми використовували як BIOS, так і програмні переривання і написали перемикач захищеного режиму в режим реального для роботи в обох режимах. Це нетривіальна річ, яку потрібно зробити під час збирання, яка вимагає від вас ознайомлення з документом специфікації архітектури Intel, який доступний на їхньому веб-сайті. Цей документ є корисним ресурсом для розуміння та реалізації переривань, схем адресації, засобів завдань, пасток та винятків.

У звичайну програму C# ми включаємо файли *.h, які є бібліотеками для ОС, які будуть включені у файл, що виконується. Якщо ми використовуємо будь-які конструктори C#, вони автоматично включатимуть ці бібліотеки, які не мають сенсу в середовищі без ОС[13]. Таким чином, середовище компіляції має використовувати пакетні файли для компіляції та зв'язування необхідних модулів. Як програміст програм на рівні мови C#, ми розробили файл інтерфейсу, що містить усі інтерфейси, необхідні для цих програм. Цей файл має API, включаючи введення/виведення, завдання, пам'ять, таймер, загальну пам'ять, блокування і т.д. Для створення виконуваних модулів ми використовуємо компілятор Visual Studio C # (пакетний режим), асемблер MASM 6.11 і компілятор Turbo. Ми написали всі пакетні файли для компіляції та завантаження, початкових та прикладних програм. Ці пакетні файли прості та легкі для розуміння та використання. API – це виклик C#, який, своєю чергою, викликає виклик C, а той, своєю чергою, викликає виклик складання. Програміст C# бачить лише API. Якщо інтерфейс можна створити лише на рівні C#, то код буде реалізовано лише на рівні C#. Ми уникаємо використання будь-яких прямих дзвінків зборки в програмі C#; однак такий виклик дозволено компілятором.

Висновки до розділу

У цьому розділі розглядалися основні поняття та концепції створення програми та опис переваг електронного посібника. Актуальність розробки даної системи, та існуючі шляхи вирішення поставленої задачі.

Зі сказаного вище, можна зробити такі висновки: розробка програми йде поетапно.

Основні ознаки електронного посібника:

1. можливість побудувати простий та зручний механізм навігації в електронному підручнику;

2. розробка пошукової системи в межах електронного підручника, зокрема, при використанні гіпертекстового формату видання;
3. можливість вбудованого автоматизованого контролю рівня знань студента;
4. можливість особливого варіанта структурування матеріалу;
5. можливість адаптації вивченого матеріалу підручника до рівня знань учня, що призводить до різкого підвищення рівня мотивації учня;
6. здатність адаптувати і оптимізувати інтерфейс користувача під індивідуальні потреби студента.

До додаткових можливостей електронного підручника, порівняно з друкованим текстом, можна віднести:

1. можливість включати спеціальні фрагменти, що імітують перебіг багатьох фізико-технологічних процесів;
2. можливість включати аудіофайли до підручника, зокрема наближати процес роботи з підручником та слухати лекції викладача;
3. вміння включати до підручника фрагменти відеофільмів для ілюстрації окремих положень підручника;
4. включення інтерактивних фрагментів до підручника задля забезпечення оперативного діалогу з учнем;
5. повномасштабне мультимедійне оформлення підручника, що включає діалог природною мовою, організацію відеоконференції з автором(ами) та консультантами за бажанням студента тощо.

Таким чином, крім різних носіїв, електронний підручник має низку принципових відмінностей від підручника, виготовленого друкарським методом: можливість мультимедіа;

- Забезпечення віртуальної реальності;
- Високий ступінь інтерактивності;
- можливість індивідуального підходу до учня.

Отже, розробка структури програми включає в себе: визначення вихідних даних для додатка; визначення вимог до зовнішнього вигляду та функціональності; формування структури програми – розділів меню.

РОЗДІЛ 2. АНАЛІЗ ПРОЕКТУ ОСВІТНЬОГО ВЕБ-ДОДАТКУ

2.1. Характеристика проекту

Подібно до більшості організаційних заходів, основна мета проекту — задоволення потреб замовника. Крім цієї фундаментальної подібності, характеристики проекту допомагають відрізнити його з інших дій організації. Основні характеристики проекту такі:

- встановлена мета проекту;
- певний час для його реалізації;
- залучення для його реалізації різних відділів та спеціалістів різного профілю;
- новизна: зазвичай виконується щось, що ніколи не робилося раніше;
- особливі вимоги щодо часу, витрат та якості виконання роботи.

Розглянемо ці показники докладніше.

По-перше, проект мають певну мету, чи то будівництво 12-поверхового житлового комплексу до 1 січня наступного року, чи випуск версії 2.0 спеціального пакету програмного забезпечення в якомога більш стислі терміни. Така єдина конкретна мета часто відсутня у повсякденному житті організації, де працівники день у день виконують одні й самі операції.

По-друге, оскільки існує чітко визначена мета, проект має суворо фіксований момент завершення, що зовсім не схоже на обов'язки, що постійно виконуються в традиційних видах робіт. У багатьох випадках працівники переходять від одного проекту до іншого, а не залишаються ви підняти одну й ту саму роботу. Наприклад, після будівництва стадіону в Самарі інженер може отримати завдання взяти участь у будівництві нафтопереробного заводу в Уфі.

По-третє, на відміну більшості видів організаційної роботи, яка ділиться за принципом функціональної спеціалізації, для проектів характерне комбінування і поєднання зусиль різноманітних фахівців. Наприклад, замість того, щоб працювати в окремих кабінетах під керівництвом конкретних менеджерів, учасники проекту, незалежно від того, чи є вони інженерами, фахівцями в галузі фінансів, маркетингу чи постачання, працюють над виконанням проекту під керівництвом керуючого проектом.

По-четверте, проекти ніколи не бувають абсолютно однаковими та завжди включають якісь унікальні елементи. І питання не в тому, чи будуть мати такі унікальні елементи, а в тому, якою буде ступінь унікальності. Очевидно, що здійснення того, що досі ніколи не робилося, як, наприклад, організація систематичного плавання торговельного флоту Північним морським шляхом, потребує вирішення нових проблем та використання нових технологій управління рухом. З іншого боку, навіть базові будівельні проекти, для виконання яких використовуються добре відомі та відпрацьовані методи та прийоми, вимагають тією чи іншою мірою адаптації їх під конкретні умови, що визначаються замовником, що також робить їх унікальними. І, нарешті, проекти пов'язують обмеження та часу, за витратами та особливі вимоги до виконання робіт. Проекти оцінюють з того, що було виконано, скільки це коштувало, і як багато часу було витрачено.

Метою освітнього проекту є реалізація державних та громадських інтересів у галузі освіти, що призводять до економічних, науково-технічних, соціальних результатів.

Проект може охоплювати різні сфери діяльності: від політичної та законотворчої до маркетингової, управлінської та навчальної, а за масштабом: від локального рівня освітньої організації до міжнародного рівня. Освітні проекти, що реалізуються на рівні освітніх закладів вищої освіти, можуть бути об'єднані у «програму освітніх проектів» для

досягнення єдиного результату, або в «портфель освітніх проектів», що дозволить використовувати проектну методологію для більш ефективного управління ними. Як партнери під час реалізації освітніх проектів мають виступати як різні органи влади, і організації різних форм власності. Найбільш ефективне співробітництво спостерігається за участю в освітніх проектах кількох освітніх закладів. Включаючись у проектну діяльність, освітні установи вищої освіти отримують можливість брати участь у наданні освітніх послуг, суттєво підвищуючи ефективність організації та якісні показники діяльності системи освіти загалом.

Життєвий цикл освітніх проектів містить усі основні стадії управління проектом: від розробки концепції проекту з надання освітньої послуги та її обґрунтування до документаційного підтвердження здобуття освітньої послуги. Основними результатами управління освітніми проектами є цілі, терміни, якість та вартість досягнення результатів. Отримати найкращі результати можна, вибираючи відповідні технології управління проектами, склад, характеристики та призначення ресурсів для реалізації освітніх проектів. Для управління ресурсами необхідно забезпечити ефективну організаційну структуру управління проектами, управління комунікаціями, персоналом тощо.

Вивчивши специфіку освітніх проектів, можна виділити основні моменти:

- особливістю реалізації освітніх проектів є наявність ліцензії на надання освітніх послуг в освітніх установах вищої освіти, часткове бюджетне фінансування (за наявності бюджетних місць), висока конкуренція на ринку освітніх послуг;
- освітні проекти, в основі яких лежить надання освітньої послуги, залежно від зовнішніх умов приносять різну прибутковість до освітньої установи вищої освіти: частина освітніх проектів

працює на імідж та розширення асортименту освітніх послуг, частина приносить основний прибуток;

- фінансування освітніх проектів здійснюється з допомогою різних джерел: кошти учнів, кошти підприємств, спонсорів тощо.

Освітнє проектування - процес розробки та реалізації проектів, спрямованих на організацію освітнього процесу та окремих його видів та напрямків.

Освітнє проектування :

- один із видів планування та організації освітнього процесу;
- це процес розробки та реалізації системи управління підвищення якості освіти або окремих елементів, де всі елементи взаємопов'язані та утворюють певну і стійку цілісність;
- практична цілеспрямована спільна діяльність студентів та викладачів;
- освітня технологія, що актуалізує суб'єктну позицію в освітньому процесі;
- метод, що йде від реальних потреб та інтересів, вікових та індивідуальних особливостей, що стимулює самодіяльність та самореалізацію;

Освітнє проектування сприяє актуалізації знань, умінь, навичок студентів, їх практичного застосування у взаємодії з оточуючим; стимулює потребу в самореалізації, самовираженні, у творчій особистісно та суспільно значущій діяльності.

Веб-додаток для тестування створено для підвищення ефективності вчителя.

Ефективність впровадження системи полягає в автоматизації процесу тестування студентів.

Зниження витрат на оплату праці зменшить і фінансові витрати, що призведе до загального зростання продуктивності та економії.

Основною метою даного розділу є визначення вартості науково-дослідних робіт, витрат на визначення економічного ефекту від основних і відповідних результатів, отриманих при розв'язанні технічних задач у підсумковій ідентифікаційній роботі для використання в суспільному виробництві. При оцінці ефективності прийнятих науково-технічних рішень повинні бути враховані всі необхідні витрати, що вимагають проведення необхідних розрахунків за певним рішенням.

Комісійні витрати орієнтуються на цінову політику мережі, як показано в таблиці 2.1.

Таблиця 2.1 – Вартість програмного та апаратного забезпечення

Назва	Кількість	Ціна
Apache HTTP Server	1	0
СУБД MySql + phpMyAdmin	1	0

Таблиця 2.2 – Матеріали

Назва	Кількість	Ціна
USB-флешка KINGSTON DataTraveler SE9 16Gb (DTSE9H)	1	450
ДискDVD-R 4.7GbSlimCaseVS	1	25

Вартість електроенергії базується на тривалості періоду розробки програмного забезпечення, кількості кіловат-годин, витрачених на розробку програмного забезпечення, і вартості електроенергії за 1 кіловат-годину. Митний збір для юридичних осіб в залежності від міста становить приблизно 4,68 грн. за кіловат годину. Витрати відображені в таблиці 2.3.

Таблиця 2.3 - Плата за електроенергію

Елементи системи	Встановлена потужність, кВт	Вартість 1 кіловат за годину (грн.)	робочі години	загальне споживання
Aspire E5-532-C5SZ Сірий	0,057	4,68	398	106,18

Амортизація обладнання оприбутковується в процесі його використання, тобто під час впровадження та створення ПЗ.

Грошова оцінка амортизації — це витрати на амортизацію, що входять до поточної собівартості.

2.2. Аналіз здійсності проекту

Проектний аналіз (project analysis) — один із основних видів робіт (метод дослідження), який виконує ініціатор проекту на передінвестиційній стадії проекту з метою визначення соціально-економічної ефективності проекту, комплексу ризиків, доцільності реалізації проекту та організації його фінансування.

Проектний аналіз є також основним методом експертизи проекту фінансовим інститутом (банком, фондом) з метою прийняття ним рішення про участь у проекті (форму участі, інструменти, вартість інструментів фінансування, що використовуються). Власне експертиза інвестиційного

проекту є процес аналізу його відповідності вимогам комерційного банку (або іншого фінансового інституту), відображеним в інвестиційній політиці банку та відповідних нормативних документах банку.

Проектний аналіз є також необхідним інструментом моніторингу інвестиційного проекту, що реалізується, комплексної оцінки його фактичної ефективності.

Таким чином, проектний аналіз повинен використовуватись на всіх стадіях життєвого циклу проекту. Результати проектного аналізу використовуються не тільки для ухвалення рішення про його виконання, а й для забезпечення відповідності фактичної ефективності проекту його розрахункової ефективності, представленої у бізнес-плані проекту.

Основними засадами проектного аналізу є:

- системність, тобто облік усієї системи взаємовідносин між учасниками проекту та їх економічним оточенням, найважливіших факторів, що впливають на витрати та результати кожного учасника, а також внутрішніх, зовнішніх та синергічних ефектів. Для такого обліку опис проекту має містити опис можливого механізму взаємодії всіх учасників проекту (організаційно-економічного механізму реалізації проекту);
- комплексний характер проектного аналізу (та експертизи проекту), що передбачає застосування різних видів аналізу проекту та підготовку комплексного бізнес-плану проекту (або висновків про результати експертизи);
- облік усіх найістотніших наслідків проекту. Оцінюючи ефективності повинні враховуватися всі істотні наслідки реалізації проекту, як безпосередньо економічні, і позаекономічні (соціальні, екологічні, впливом геть безпеку країни). У тих випадках, коли вплив таких наслідків реалізації проекту на ефективність припускає кількісну

оцінку, її слід зробити. В інших випадках облік цього впливу має здійснюватися експертно;

- проведення аналізу проекту протягом усього його життєвого циклу (окремими учасниками – до виходу із проекту);
- застосування сучасних міжнародних стандартів управління проектами та оцінки проектів;
- відповідність стратегії розвитку та інвестиційної політики країни, регіону, компанії (банку);
- наявність позитивного соціально-економічного ефекту проекту;
- висока якість та достатність проектної та іншої документації, необхідної для підготовки бізнес-плану проекту (або проведення експертизи проекту у банку).

Проектний аналіз включає:

1. Технічний аналіз – чи є проект технічно обґрунтованим.
2. Перспективність - чи є попит продукції проекту.
3. Фінансовий аналіз - чи є проект життєздатним у фінансовому відношенні, чи відшкодовуються витрати за рахунок продажу.
4. Екологія – вплив на довкілля.
5. Організаційний розвиток - чи зможе відповідальна за проект організація здійснити проект та керувати його експлуатацією.
6. Прийнятність проекту у соціальному та культурному відношенні – відповідає чи проект місцевим умовам, чи не має негативного значення деякі групи населення.
7. Економічний аналіз - чи переважають позитивні результати проекту витрати на його здійснення та експлуатацію, з якими

ризиками пов'язаний проект (аналіз ризику).

Для аналізу витрат та доходів за проектом необхідно відповісти на наступні питання:

- який бажаний результат проекту;
- хто здійснює витрати на проект;
- хто скористається позитивними результатами проекту;
- коли та як з'являться позитивні результати проекту

Для аналізу проекту часто використовують певні критерії, які допомагають оцінити вигідність проекту та повернення вкладень.

Вирізняють такі критерії цінності проекту:

1. Чиста наведена вартість.
2. Внутрішня ставка доходності.
3. Співвідношення вигод та витрат проекту.
4. Ефективність витрат.
5. Прибуток у перший рік експлуатації.
6. Період окупності капітальних вкладень.
7. Порівняння різних критеріїв цінності.

Інфляція може бути визначена як загальне зростання цін чи загальний спад реальної грошової вартості. Інфляція, зазвичай, впливає потоки готівки проектів. В цілому, вплив виявляється у наступному:

- (а) збільшення виручки;
- (б) збільшення витрат;
- (в) можливе збільшення відсотків та довгострокових зобов'язань.

У період збільшення інфляції кредитори вимагатимуть більшого повернення. Процентні ставки зазвичай складаються з двох компонентів: реальної процентної ставки та поправки на інфляцію.

ОЦІНКА РИЗИКУ ПРОЕКТУ.

При визначенні ризику можуть виникнути такі проблеми:

- (а) всі рішення ґрунтуються на прогнозах
- (б) всім прогнозам по-різному притаманна невизначеність
- (в) труднощі, викликані з відображенням невизначеності у фінансовій оцінці.

Деякі автори так розрізняють поняття ризик та невизначеність:

Ризик -фактор ймовірності, пов'язаний з можливими результатами, що дає підстави прогнозувати результати, які можна розрахувати математично.

Невизначеність – неможливо зробити математичний прогноз майбутнього результату виходячи з наявних даних.

Таким чином, ризик можна визначити кількісно, а фактор невизначеності немає.

Хоча існує чітка різниця між цими двома проблемами, практиці слова ризик та невизначеність взаємозамінні. Ситуації ризику найкраще вирішувати за допомогою імовірнісних розподілів.

Амортизаційні відрахування визначаються виходячи з норми амортизації.

Норма амортизації визначається виходячи з економічно доцільного строку корисного використання і має компенсувати зношеність основних фондів при можливому зносі та створити економічну основу для заміни [14].

Амортизаційні відрахування за кожну годину роботи системи розраховуються за формулою (2.1).

$$A_{ч} = \Phi_{\text{перв}} / F_{д} \quad (2.1)$$

Де $\Phi_{\text{перв}}$ початкова вартість системи або окремих елементів;

F_d - Річний фонд годин роботи (2500 годин)

Таблиця 2.4 - Розрахунок амортизаційних

п/п	Елемент КТС	$\Phi_{\text{перв}}$	F_d	$A_{\text{ч}}$	Кількість годин	Загальна вартість (грн.)
1	Aspire E5-532- C5SZ Сірий	24999	2500	1,9999	398	795,9602
Загалом						795,9602

Підсумовуючи, вартість придбання, обслуговування та експлуатації програмного та апаратного забезпечення, яке ми отримали, становить 26 415,14 грн.

Вартість заробітної плати.

Для розрахунку витрат на заробітну плату потрібно середньо годинну ставку програмістів помножити на трудомісткість роботи на кожному етапі розробки системи труд./год.

На основі отриманих даних можна розрахувати заробітну плату для всіх етапів розвитку, результати наведено в таблиці 3.5.

Таблиця 3.5 - Розрахунок основної та додаткової заробітної плати

п/п	Зміст роботи	Трудомісткість роботи, люд/год	Основна заробітна плата(грн.)
1	Аналіз предметної області	16	2880
2	Постановка задачі	8	1040
3	Розробка технічного завдання	12	2260
4	Проектування бази	32	4160

5	Розробка інтерфейсу програми	24	4120
6	Розробка модулів програми	115	15950
7	Тестування системи	120	15600
8	Документування	11	1550
9	Керівництво роботи	20	5100
10	Загалом	398	52660

Розрахувати вартість доплати.

Додаткова заробітна плата для розробників визначається у відсотках (15%) від загальної суми посадового окладу [8].

$$ЗП_{\text{дод.}} = 52660 * 0,15 = 7899 \text{ (грн.)}.$$

Кошторис – повний розрахунок витрат підприємства на виробництво та реалізацію продукції за певний календарний період (рік, квартал), складений з економічних елементів собівартості. Оцінки витрат розраховуються шляхом прямого підсумовування окремих економічних елементів, а також оцінок комплексних витрат або кошторисів для різних секторів бізнесу [15].

Таблиця 2.6 - Загальний кошторис витрат

п/п	Види витрат	Сума, грн.
1	Придбання, технічне обслуговування та експлуатація програмного та апаратного забезпечення	26 415,14
2	Основна заробітна плата	52660
3	Додаткові витрати на заробітну плату	7899
4	Соціальна потреба у відрахуваннях	18167,7
Загалом		105141,84

2.3. Планування та моніторинг проекту

При розгляді процесу управління проектом планування та моніторинг проекту є важливим кроком проекту. Як тільки розпочинається виконання проекту, починається і моніторинг проекту.

Моніторинг проекту включає відстеження показників проекту, хід виконання та пов'язаних завдань, щоб переконатися, що все виконано вчасно, в рамках бюджету та відповідно до вимог та стандартів проекту. Моніторинг проекту також включає розпізнавання та виявлення перешкод або проблем, які можуть виникнути під час виконання проекту, та вжиття заходів щодо усунення цих проблем[16].

При плануванні свого часу викладач повинен враховувати час на використання додатків та методів навчання, які застосовуються у даному програмному додатку, наприклад завантаження навчальних матеріалів, виготовлення презентацій та підготовка мультимедійних анімацій. Так це дійсно займає трохи часу та спочатку викладачам буде незвично цим користатися, але в майбутньому, коли вже буде все автоматизовано, це буде значно економити час.

Також необхідно враховувати значення середньої вартості робочого часу вчителів.

Оскільки програмне забезпечення, що розробляється, покликане полегшити роботу вчителів, які також є клієнтами, окремо додаткових витрат на оплату праці не буде.

Опрацювання документів, складання тестів $t_{n1} = 3\Gamma$, за допомогою веб-додатку робота зі складання тестів буде меншою, $t_{n2} = 1\Gamma$.

Економія часу становитиме:

$$t_{n1} - t_{n2} = 3\Gamma - 1\Gamma = 2 \text{ години на день.}$$

За один місяць вчитель зекономив час: $T_{pm1} = 26 * 2\text{Год} = 52\text{Г}$.

Це означає, що за рахунок економії часу можна збільшити навчальні години і вигода в грошовому вираженні складе:
 $R_{m1} = 168 * 26 * 52\text{Г} = 8749,52\text{грн}$. на місяць. Далі розраховуємо термін окупності системи: поточний $T_{ok} = 105141.84 / 8749.52 = 12.01$ місяць.

Виходячи з принципів планування та зовнішніх зобов'язань Організації необхідно застосовувати два рівні планування:

Перший рівень – стратегічне планування, великоблочне планування проектів з урахуванням Правил планування проектів.

Другий рівень – тактичне планування спринтів та визначення коротких цілей.

Така розбивка пов'язана з принципом “необхідності та достатності”, невизначеністю змісту проекту та непередбачуваності поведінки людей.

Рівень планування проектів забезпечує планування черговості виконання робіт Організацією, задає зрозумілі фокусуючі цілі та забезпечує контроль за виконанням зобов'язань.

Рівень тактичного планування спринтів забезпечує короткі та зрозумілі цілі для співробітників, що реалізують проект і враховує високу невизначеність способу реалізації.

Планування це процес

Планування як процес не повинен обмежуватися сесіями у форматі “зробив і забув”. Це процес, який вимагає стільки ж зусиль, як і проектування Продукту. Планування та вибір пріоритетів майбутніх робіт має відбуватися в той час, поки поточні роботи ще не завершено. Плануванням можна займатися тільки вийшовши з “потіка, що створює” – Потіку Реалізації в Потік Управління.

Правила:

Планування майбутніх проектів та розвитку Продуктів виконується регулярно.

Сесії планування "вшити" у робочий процес.

У плануванні може взяти участь будь-який співробітник, якщо бачить свою користь.

Наради щодо планування збираються регулярно для синхронізації всіх планів та бачення рішень.

Планування проектів

Коли ви будете розклад проектів, ви програмуєте прийняття рішень Організації на тривалий період. Маючи календарний розклад проекту та черговість виконання завдань, не

потрібно витратити зусилля на вибір найважливішого і пріоритетного завдання щоразу, коли це необхідно. Це заощаджує зусилля на прийняття рішень.

Розклад проекту – це правило, яке визначає пріоритет завдань, якого слід дотримуватися. Якщо щось змінилося у зовнішньому чи внутрішньому середовищі, слід перепланувати для обліку всіх зацікавлених сторін і зобов'язань Організації. Продовження роботи за старим планом є небезпечним для Організації. Наявність календарного розкладу дозволяє краще розуміти приблизні терміни досягнення цілей, на основі цього синхронізувати інші роботи.

Стратегія планування проектів

При плануванні виконання проектів слід враховувати кілька факторів:

- Тривалість проекту
- Прямий прибуток від реалізованого проекту.

Непрямий прибуток від реалізованого проекту, або сума зміни доходу компанії за 12 місяців при впровадженні продукту, створеного в рамках

проекту. Економія ресурсів на виконанні роботи – також прибуток Організації.

Вартість затримки постачання результату у обіцяний термін.

Можливість отримання прибутку від розроблюваного Продукту до завершення всього проекту створення Продукту.

Виходячи з цих факторів, необхідно вибрати стратегію реалізації всього портфеля проектів.

Варіанти стратегій:

Усі ресурси організації фокусуються однією прибутковому проекті скорочення часу його виконання.

Проекти каскадуються (вирівнюються) за ресурсом-обмеженням.

Стратегія фокусу на одному проекті

Фокус на одному проекті забезпечує високий рівень взаємодії всіх учасників проекту та взаємодопомогу. Навіть, коли співробітник не зайнятий жодною роботою, він може допомагати іншим її виконувати. За рахунок цього скорочується час виконання проекту.

Мінуси такого підходу в тому, що створюється ілюзія простою ресурсів. Особливо коли деякі ресурси потрібні лише на початку проекту, а потім не потрібні. Але це також ілюзія. Наприклад:

У IT-проекті є фази: аналіз, розробка, тестування. Аналіз виконують аналітики, розробку – розробники, тестування – випробувачі. Якщо вся робота з аналізу виконана спочатку проекту, то є ілюзія, що аналітики простоюватимуть. Але насправді вони постійно потрібні для роз'яснення вимог Виробу або їх уточнення. Звичайно, не в повному обсязі, але потрібні.

У будівельному проекті на фазі "реалізація" є "авторський нагляд" від проектувальників споруди. Тобто регулярна перевірка відповідності результату технічному завданню та проектній документації.

Пам'ятайте, що відсутність своєчасної допомоги від колег знижує мотивацію виконавця, відповідно і швидкість реалізації Продукту.

Деякими проектами легко керувати, інші ж вимагають величезних зусиль. У будь-якому випадку наявність грамотно складеного плану допоможе вам забезпечити його бездоганну реалізацію. Якщо у вас є чітке уявлення про все, що має бути зроблено і про те, як це буде реалізовано в рамках заданого періоду часу, ви зможете якщо і не виключити, хоча б мінімізувати ймовірність провалу. У цій публікації ми надамо список методів, які ви можете використовувати для складання якісного плану проекту та його успішної реалізації. Ми розглянемо різні аспекти планування: організація процесу планування, планування ресурсів, управління інформацією, побудова карти процесів, прийняття рішень та управління ризиками.

Процес планування

На першому етапі планування проекту необхідно визначити цілі, скласти план проекту та проаналізувати бюджет. Для швидкого виконання цих завдань можна використовувати наведені нижче методи планування проекту.

SMART-цілі

SMART – це критерій для постановки цілей та завдань для проектів. Відповідно до SMART, цілі вашого проекту мають бути конкретними, вимірними, досяжними, актуальними та прив'язаними до термінів.

Структура декомпозиції робіт

Структура декомпозиції робіт (WBS) – це інструмент управління проектами, який використовується для візуалізації результатів проекту в ієрархічній структурі. Він допомагає керівникам проектів розбити складну проектну діяльність на прості та керовані завдання.

Як його використовувати

Крок 1. Першим кроком створення структури декомпозиції робіт є розуміння масштабу проекту. Це означає, що необхідно визначити цілі проекту, результати, завдання, терміни і т.д.

Крок 2. Визначте ключові результати. До ключових результатів належать такі, які є важливими для успіху проекту; вони повинні бути на рівні 2 вашої структури розбивки робіт.

Крок 3. Визначте пакети робіт чи результати проекту на найнижчому рівні структури. Зберіть проектну команду та зацікавлені сторони, щоб спільно визначити весь обсяг робіт, необхідних для досягнення ключових результатів.

Крок 4: Створіть розшифрування структури, яка містить визначення та обсяг елементів (пакетів робіт) у складі діаграми.

Матриця пріоритетності

Матриця пріоритетності - це інструмент планування проекту, який використовується для визначення того, які заходи перебувають у пріоритеті, яких слід уникати, щоб максимально ефективно використовувати свій час і ресурси. Цей інструмент допомагає приймати оптимальні рішення.

Як його використовувати

У матриці пріоритетів дій по осі x відкладаються зусилля, необхідні для тієї чи іншої діяльності, а по осі y — її результативність. Матриця пріоритетів пріоритетності пропонує 4 варіанти категоризації:

1. Швидкі результати/Терміново — це дії, які приносять велику віддачу та не вимагають великих зусиль для їхнього завершення. Краще максимально зосередитись на таких завданнях.

2. Великі проекти/Не термінові — це важливі завдання/заходи, які потребують великих зусиль. Їх не можна виконувати на шкоду завданням із

швидким результатом та необхідно ретельно планувати до початку виконання.

3. Поточна робота/Делегувати – це низькопріоритетні завдання, які не є важливими та не потребують великих зусиль. Ви можете працювати над ними, коли у вас є час у перервах між завданнями перших двох категорій або делегувати іншій команді.

4. Каторжний труд/Отложить чи ігнорувати — це завдання, які приносять низьку віддачу, але потребують великих зусиль. Рекомендується уникати їх виконання, а якщо уникнути їх неможливо, вдаватися до допомоги фахівців, які зможуть ефективно впоратися з ними.

Структура розбиття витрат

Структура розбивки витрат (на основі структури декомпозиції робіт) — це один із методів планування проекту, який використовується для розподілу витрат за різними частинами проекту (за видами діяльності на найнижчому рівні структури розбивки робіт). Він використовується для порівняння фактичних видатків з бюджетом.

Як його використовувати

Крок 1: У своїй структурі розбивки робіт переконайтеся, що всі комплекси робіт оформлені як окремі завдання, що розмежовуються. Призначте кожному дату початку та закінчення для кожного комплексу робіт.

Крок 2: Визначте конкретну суму бюджету кожного комплексу робіт і призначте відповідальних власників бюджету.

Крок 3: Вивчіть витрати і визначте, скільки вам коштуватиме проект на кожному рівні структури розбивки робіт.

Планування ресурсів

Для ефективнішого виконання проектів необхідно правильно розподіляти ресурси. На етапі планування ресурсів необхідно звернути

увагу до типи ресурсів, необхідні виконання завдань проекту, і визначити тривалість виконання кожного завдання.

Діаграми Ганта

Діаграми Ганта – це інструмент та методика, яка використовується для планування та контролю виконання завдань за проектом. Також цей метод можна використовувати для фіксації витрат та витрат на різних стадіях проекту.

Як його використовувати

Крок 1: Розбийте проект на окремі завдання та призначте рядки на діаграмі Ганта.

Крок 2: Кожне завдання має тривалість у часі, і ви можете виділити їх потрібним кольором. Поверх тимчасової шкали розташована лінія прогресу, за допомогою якої можна відзначити, скільки роботи було виконано за кожним із завдань.

Організаційна структура

Схеми організаційної структури - це інструмент планування проекту, який застосовується для візуалізації проектної команди та розподілу ролей та обов'язків усередині неї. З їхньою допомогою можна як зобразити ієрархію і структуру підзвітності, а й використовувати його під час розподілу завдань між співробітниками.

Як використати цей метод

Крок 1: Вивчіть доступні вам ресурси та співвіднесіть їх із завданнями.

Крок 2: Розберіть у ієрархії/структурі підзвітності всередині команди проекту.

Крок 3: Використовуючи шаблон оргструктури, подібний до наведеного нижче, ви можете призначити кожному співробітнику своє місце. Ви можете внести в організаційну структуру необхідну інформацію,

наприклад, ролі та обов'язки, доступність і т.д. На підставі цих даних ви можете надавати кожному учаснику відповідні завдання.

Управління знаннями

Працюючи над окремими частинами проекту команді доводиться звертатися до різних ресурсів, як-от практичні керівництва, протоколи нарад, документи з процесам тощо. Якщо є єдине місце (щось на зразок бази даних), де можна отримати доступ до всієї необхідної інформації, це допоможе заощадити час.

Діаграми зв'язків (mind maps)

Діаграми зв'язків — інструмент, який використовується для організації інформації. Проектні команди використовують діаграми зв'язків під час мозкових штурмів, щоб зафіксувати вільний потік ідей та структурувати їх у зрозумілій формі. Крім того, їх можна використовувати як ментальні карти, на яких зберігається необхідна інформація.

Як його використовувати

Крок 1: Зберіть ресурси, до яких команда повинна звертатися, та розподіліть їх за категоріями. Наприклад, в рамках проекту створення веб-сайту записи в блозі, відео, інфографіка з планування веб-сайту можуть бути віднесені до категорії "Довідкові матеріали з планування веб-сайту".

Крок 2: Використовуючи Creately, можна додати відповідні посилання. Просто клацніть фігуру, відкрийте меню "Властивості" та додайте своє посилання у розділі "Посилання на URL-адресу або діаграму"

Крок 3: Якщо ви вставляєте діаграму на веб-сторінку, опублікуйте її як інтерактивну діаграму за допомогою програми Creately Viewer.

Картування процесів

Якщо ви покроково пояснили, як виконати те чи інше завдання, то будь-хто, незалежно від того, знайомий він з процесом чи ні, зможе виконати його легко. Для цього й потрібна побудова карти процесів.

Карти процесів

Карты процесів – це інструмент, який використовується для візуалізації етапів робочої діяльності та людей, які за неї відповідають. Щоб дізнатися, що таке карти процесів, символи карт процесів, типи карт процесів і як намалювати карту процесів, ознайомтеся з нашим простим посібником зі складання карт процесів.

Діаграма PERT

PERT розшифровується як Project Evaluation Review Technique (Метод коригування та оцінки проекту). Він допомагає від початку до кінця побудувати схему ваших процесів. Діаграма PERT допомагає командам швидко виявляти проблемні аспекти проектів, які можна миттєво усунути. Це допомагає заощадити багато часу в довгостроковій перспективі і не чекати, доки проблеми з'являться самі собою. У цьому докладному посібнику з діаграм PERT ми розповіли, як використовувати цей метод планування проекту і які переваги дасть вам його застосування.

Мозковий штурм/прийняття рішень

Як з'ясувалося, сесії мозкового штурму та прийняття рішень займають чимало часу у процесі планування проекту та на всьому протязі його реалізації. Мозковий штурм не тільки необхідний для генерування творчих ідей, але і є незамінним прийомом при прийнятті рішень та вирішенні проблем.

Діаграми спорідненості

Діаграми спорідненості використовуються для угруповання великого набору ідей/даних/думок у рамках певної теми або категорії на основі їх відповідності одна одній. Його можна використовувати для логічного упорядкування ідей, отриманих під час мозкового штурму.

Як його використовувати

Крок 1: Зберіть компетентну команду для мозкового штурму.

Крок 2: Запишіть кожну ідею/думку, висловлену під час мозкового штурму.

Крок 3: Сортуйте ідеї за рівнем їхньої подібності.

Крок 4: Згрупуйте їх разом та дайте кожній групі назву.

Діаграма PISCK

Діаграма PISCK це ще один метод планування проекту, який можна використовувати для організації ідей після мозкового штурму. Це допоможе вам визначити, які ідеї легко реалізувати та які мають високу віддачу, або, іншими словами, які ідеї є найкориснішими.

Як його використовувати

Згідно з діаграмою PISCK, ви можете позначити свої ідеї як:

- Можливі: це ідеї, які легко здійснити, але з низькою віддачею
- До реалізації: ідеї в цій частині діаграми легко реалізуються та мають високу віддачу.

Виклики: тут ідеї важче реалізувати, а віддачу важко визначити

Без шансів: ці ідеї важче реалізувати, і вони мають низьку віддачу

Після того як ви розмістите ідеї, отримані в результаті мозкового штурму, на матриці стане ясніше, над якою ідеєю вам слід працювати в першу чергу, а від якої краще зовсім відмовитися.

Дерево прийняття рішень

Коли ви пропонуєте кілька варіантів дій, ви можете використовувати дерево прийняття рішень для аналізу різних результатів цих рішень.

Як його використовувати

Крок 1: Виявіть та сформулюйте питання та помістіть його на початок діаграми.

Крок 2: Запишіть можливі відповіді на лініях, що йдуть на наступний рівень фігур дерева рішень.

Крок 3: Покладіть результати відповіді на ці запитання у фігурах, до яких йдуть лінії. Ви можете використовувати овал для позначення кінцевих результатів. Якщо результат призводить до нового питання, помістіть його у прямокутник.

Крок 4: Зберіть команду осіб, які приймають рішення, і перегляньте результати аналізу, що вийшов, щоб зробити остаточний вибірУправління ризиками

Ризики - неминучий елемент проектів, і найкращий спосіб мінімізувати їхній вплив на досягнення ваших цілей - підготуватися до них заздалегідь. Заздалегідь виявляючи та аналізуючи ризики, ви можете уникнути негативних результатів, які можуть вплинути на результати проекту. У нашому повному списку візуальних методів управління ризиками ми розглянули 11 ефективних методів управління ризиками, які можна використовувати для виявлення, оцінки та контролю ризиків.

Висновки до розділу

У цьому розділі аналізується розроблене програмне забезпечення з точки зору актуальності та економічної доцільності його впровадження, а не використовуваних засобів. По-перше, визначається складність і час роботи по створенню програмного додатку.

Наступним кроком є розрахунок вартості впровадження та впровадження розробленого веб-додатку. Заключним етапом є розрахунок економічного ефекту від впровадження інформаційної системи.

Також враховується доцільність планування часу викладача, який викладач, як користувач повинен витратити на підготовку до заняття. Враховується час на використання додатків та методів навчання, які

застосовуються у даному програмному додатку, наприклад завантаження навчальних матеріалів, виготовлення презентацій та підготовка мультимедійних анімацій. Так це дійсно займає трохи часу та спочатку викладачам буде незвично цим користатися, але в майбутньому, коли вже буде все автоматизовано, це буде значно економити час при перевірці робіт, потрібно буде лише один раз завантажити навчальні матеріали та інколи вносити зміни, якщо це необхідно.

Проведені розрахунки дозволяють говорити про доцільність та економічну вигоду з точки зору витрат часу впровадження інформаційної тестової системи.

РОЗДІЛ 3. ЕТАПИ РОЗРОБКИ ТА СТВОРЕННЯ ПРОГРАМНОГО ДОДАТКУ

3.1. Опис алгоритму та середовища для розробки

Перед тим як розпочати безпосередню розробку додатку потрібно пройти декілька кроків:

Аналіз бізнесу клієнта

Перед тим, як приступити до написання технічної документації, варто детально вивчити бізнесу клієнта. Під час цього етапу ми фокусуємось на 3-х складових:

- Поринаємо в бізнес замовника. Всі мають свої особливості: у взаємодіях з клієнтами, у бізнес-процесах, у внутрішній кухні. Важливо з'ясувати всі деталі бізнесу, щоб майбутня мобільна програма максимально задовольняла завдання проекту.
- Розглядаємо поточні біль бізнесу та цілі, досягнення яких стоїть перед створюваним додатком. Часто після цього змінюється концепція програми, оскільки раніше обране ІТ-рішення перестає працювати задля досягнення позначених цілей.
- Аналізуємо конкурентів та їх ІТ-рішення. Це допомагає запозичити чужий досвід і не робити помилки, яких інші припустилися.

Виходячи з отриманих даних, формуємо сценарії використання програми та її функціональні можливості.

Складання технічного завдання

Після того, як отримано чітку картину бізнесу, наша команда переходить до написання технічного завдання. Це документ, який включатиме всі вимоги та етапи розробки мобільного додатка: від самої концепції програми та до тих елементів, які потрібно буде протестувати на платформах.

7 головних елементів, які потрібно враховувати:

- Системи зсередині. Вивчаємо програмні рішення, які впроваджені у компанії, особливо ті, з якими доведеться інтегруватися надалі.
- Подібні рішення. Наш бізнес-аналітик аналізує програми з вашої сфери або інших сфер зі схожим функціоналом, формує перелік їх слабких або сильних сторін.
- Побажання щодо дизайну. Під час цього етапу фіксуються всі вимоги до дизайну майбутньої мобільної програми. Тут важливо розуміти, що деякі запити по анімації або особливостям дизайну можуть вплинути на вибір технологій, що використовуються.
- User stories. Спеціаліст прописує всі основні та додаткові сценарії використання програми користувачами з метою виявлення недостатнього функціоналу, реалізація якого необхідна для повноцінного функціонування проекту.
- Вивчення цільової аудиторії. Бізнес-аналітик спільно з маркетингологом аналізують особливості аудиторії, яка користуватиметься додатком: локація (мовне питання, а також сезонні пропозиції/акції), платоспроможність та девайси, що використовуються. За рахунок цих даних ваш мобільний продукт буде більш ефективним та затребуваним для вашої ЦА.
- Вибір технологій. Існує нативна (окремі програми для Android або IOS) та кроссплатформова (єдиний код для двох систем) технології розробки мобільних додатків. Необхідний бюджет на проект при використанні кроссплатформенного фреймворку буде меншим, ніж при написанні двох нативних додатків. Однак варто враховувати той факт, що якщо майбутній проект складний, вимагає роботи з файловою системою, використання 3D-графіки

або відбитків пальців, то з вирішенням подібних завдань значно впорається нативна технологія.

Наприклад, при розробці програми конвертера валют було встановлено, що воно міститиме функції, які можуть бути реалізовані тільки нативним інструментом, тому використання кросплатформового рішення для завдань, що не буде рентабельним.

Фіналізація. Коли ТЗ буде на завершальній стадії, важливо ще раз перевірити, чи вирішує застосування болу та потреби, які були виявлені на старті. Часто буває, що переробки та доопрацювання функціоналу програми відводять від первісної проблеми, і в результаті вона так і не буде вирішена.

Складання прототипу.

Створення прототипу є важливим етапом, оскільки саме візуалізація є одним з основних факторів успішності застосування. У цьому етапі завдання менеджера відмалювати всі екрани взаємодії.

Якщо програма зручна (а фундамент зручностей закладається саме при створенні UX макета - прототипу), то шанс того, що програма сподобається користувачам, явно зростає.

Крім того, саме на етапі створення прототипу проводиться тестування прописаних у ТЗ сценаріїв використання. Також за допомогою прототипу може бути виявлено функціонал, який було втрачено, а також якісь логічні невідповідності, які були непомітні при написанні ТЗ.

Під час розробки прототипу ми враховуємо такі нюанси:

- Основні елементи навігації та управління мають бути внизу, щоб їх було зручно натискати великим пальцем;
- Для економії простору логотип можна використовувати не на всіх екранах, а лише на головному та замість прелоадера завантаження;
- Варто уникати розміщення підказок у неочевидних елементах інтерфейсу;

- Елементи навігації не повинні бути розміщені надто близько один до одного, так при натисканні користувач може натискати відразу на кілька кнопок, запитуючи таким чином виконання непотрібної дії;
- Чи буде програма доступна в різних положеннях екрана — портретному та горизонтальному чи ні (можна вибрати один вид відображення та закріпити його на програмному рівні);

Дизайн проекту

Для мобільних програм крім функціональних елементів потрібен дизайн, який буде відповідати всім тенденціям і вимогам UX. На ринку сьогодні велика конкуренція і важливо, щоб саме ваш товар запам'ятовувався в голові у покупця.

Проробка та створення дизайну включає два важливі компоненти: потреби користувача та елементи самого UX. Що стосується потреби користувача — то сюди можна віднести: навігацію, зв'язок екранів, черговість компонентів, тоді як для UX необхідно дотримуватися поєднання кольорів, фірмового стилю, розмірів тексту та композиції об'єктів між собою.

Особливості, які ми обов'язково враховуємо під час розробки дизайну наших додатків:

- Розмір шрифту основних елементів не повинен бути менше 14px, інакше він може бути погано читаним;
- Кольори інтерфейсних елементів повинні підказувати користувачеві, як працювати з програмою. Наприклад, які елементи клікабельні, а які ні, чи правильно виконав дії, чи щось потрібно виправити;
- Фінальний дизайн слід адаптувати під 5 найпопулярніших розмірів екранів, щоб програма однаково добре відображалася у всіх користувачів.

Програмування

Після готового ТЗ та відмальованого дизайну для застосування робота над проектом переміщується на стадію програмування. Під час цього етапу створюється функціонал самого додатку. Цей процес поділяється на дві частини: Front-end (клієнтська) та Back-end (серверна) частини.

Для написання Front-end програми ми використовуємо такі технології: для нативних програм - Kotlin для Android і Swift для IOS; для кросплатформових додатків - Flutter.

Основним стеком для написання Back-end частини є мова програмування Python, фреймворк Django. Цей фреймворк підходить як для розробки простих додатків, стартапів, інтернет-магазинів, так і для високонавантажених сервісів (наприклад, найпопулярніші сервіси Instagram, Spotify, Youtube написані з використанням мови програмування Python Django).

Під час розробки обох частин проекту існують певні особливості, про які також не слід забувати:

Оскільки екран мобільного додатка набагато менший за Desktop, то неможливо видати таку ж кількість фільтрів і категорій, які є на стандартному сайті. Для цього робляться спеціальні динамічні або фільтри, що ховаються.

Front-end розробники відповідальні отримання даних від користувачів (геолокація, камера, доступом до камері і тп). Ми прописуємо кожен елемент у коді окремо, щоб ваша програма могла отримати всі необхідні дані від користувачів як для IOS, так і Android

Якщо ви хочете, щоб ваша програма працювала навіть у ті моменти, коли відсутня інтернет, розробник повинен буде створити локальну базу даних для програми. Наприклад, користувачі програми інтернет-магазину зможуть переміщатися магазином і додавати бажані товари в кошик без будь-якого інтернет-з'єднання. Оплату вони зможуть зробити як тільки з'явиться інтернет.

Крім того, що в додатку прописуються користувальницькі сценарії, не варто забувати про такі важливі елементи, як свайпи або скроли. Якщо ви хочете, щоб у користувача програми була така можливість, то цей елемент необхідно закласти на стадії дизайну.

Back-end

Ця частина програми, яка не менш важлива ніж Front-end, оскільки в залежності від швидкості завантаження програми та відображення елементів залежить популярність вашої програми.

1. Згідно з дослідженням Back-end, відсоток користувачів, які видаляли програми через повільне завантаження, дорівнював 70%. Для того, щоб back-end працював та віддавав усі відповіді на запити швидко, необхідно заздалегідь подумати про оптимізацію коду та кешування даних, а також закласти час на рефакторинг.

2. Якщо вашим додатком будуть користуватися люди з різних країн, то важливо зробити так, щоб push-повідомлення надходили для них у зручний час.

3. Ще один важливий момент, який потрібно налаштувати на back-end частині проекту – це відображення спеціальних акцій чи товарів залежно від місцезнаходження.

Тестування

За рахунок цього етапу усуваються всі баги та технічні невідповідності у додатку. Ігнорувати даний етап означає втратити перших користувачів після запуску, оскільки якісь модулі можуть не працювати на тому чи іншому пристрої. Як і при тестуванні сайтів, у веб-додатку спочатку тестується верстка та її відповідність дизайну, після чого перевіряється функціональна частина проекту. При тестуванні програмної частини наша команда звертає увагу на два компоненти:

1. Версія операційної системи. Під час тестування програми, не так важливо дозвіл екранів або рік випуску смартфона. Насамперед, важлива

версія ПЗ. Технічні вимоги до ПЗ мають бути зазначені у технічному завданні. Відразу перевіряємо останню, тобто потрібну версію.

2. Внутрішня оболонка програм. При розробці Android-додатків виникають деякі нюанси, пов'язані з особливостями операційної системи, вбудованої оболонки пристрою. Зокрема такі особливості є у смартфонів VIVO, Samsung, Realme, Xiaomi, OPPO, Lenovo. Під час розробки велика частка часу йде саме на дебаг "дрібниць", викликаних цими оболонками.

Запуск

Після того, як програма готова, її потрібно додати на маркетплейс App store та Play market. На жаль, цей етап не позбавлений проблем, оскільки додаток має відповідати політиці платформ, а деякі вимоги буває складно дотриматися. Так, для відповідності політиці платформ App Store або Play Market важливо пам'ятати, що додаток повинен містити тільки якісні, авторські картинки, а також контент не порушувати політику платформи.

Для того, щоб програма активно завантажувалася, потрібно додати зображення та відео самого інтерфейсу програми в опис на платформу. Так користувач зрозуміє його зміст та складові компоненти. Не варто забувати про рейтинги та коментарі. Чим нижчі рейтинги, тим менша ймовірність, що Ваш додаток вийде в топ за категорією, для якої ви розробляєте програму (навчання, бізнес, ігри).

Авторське інструментальне середовище ToolBook Instructor фірми SumTotal (раніше - фірми Asymetrix), що є потужним пакетом, що добре зарекомендував себе як дуже зручний засіб розробки мультимедійних програм, що навчають.

До безперечних переваг цього середовища ставляться:

- простота та наочність при компонуванні навчального курсу,
- підтримка великої кількості різноманітних медіа форматів (включаючи звук, анімацію, цифрове відео, стислі зображення тощо),
- наявність вбудованої мови сценаріїв Open Script,

- редактор Actions Editor,
- Можливість застосування DHTML,
- велика бібліотека (каталог) готових об'єктів,
- Відкритість середовища для створення нових об'єктів,
- можливість організувати безліч різновидів завдань у контрольному блоці,
- Захист навчального курсу від несанкціонованого доступу до даних та ін.

ToolBook Instructor дозволяє створювати ефективні навчальні та освітні продукти, у тому числі працюючі дистанційно в Інтернеті. Для роботи з ними достатньо простого браузеру. Розроблені мультимедіа програми можуть розповсюджуватися на CD-ROM, безпосередньо через Інтернет або бути впроваджені в Web-сторінку.

Слід однак відразу помітити, що фактором, що гальмує широке використання цього середовища (як і інших зарубіжних авторських систем) є його досить висока ціна. Це звужує коло покупців.

Середовище розробки - це набір процесів та інструментів, які використовуються для розробки вихідного програмного коду або програмного продукту. Це включає в себе все середовище, яке підтримує наскрізний процес, включаючи сервери розробки, підготовки та виробництва. Середовище розробки автоматизує або спрощує процедури, пов'язані зі створенням, тестуванням, налагодженням, встановленням виправлень, оновленням та супроводом програмного забезпечення, включаючи довгострокове обслуговування.

Даний методичний посібник покликаний полегшити завдання освоєння цього авторського пакета, щоб англomовний інтерфейс системи став каменем спотикання для користувачів, особливо непрофесіоналів в інформаційних технологіях.

Мультимедіа-додаток у цьому середовищі формується як комплект із книг, кожна з яких асоціюється з окремим файлом з розширенням .tbk. Книги діляться на сторінки, причому на екрані зазвичай з'являється лише одна з них. Та чи інша сторінка може містити текст, а також символічні елементи (об'єкти), що позначають клавіші, поля прокручування, текстові поля, графічні зображення тощо.

Відео- та аудіо-фрагменти в процесі компонування не вбудовуються всередину tbk-файлу, а встановлюється лише зв'язок з ними за допомогою спеціальних об'єктів - плеєрів, що розміщуються на сторінці у вигляді кнопок включення/вимкнення даного медіафрагменту.

Є можливість не вбудовувати також і графічні об'єкти всередину tbk-файлу, а лише визначати їхнє розташування на екрані. Цей момент важливий під час створення Internet-версій навчальних курсів серед ToolBook Instructor. І тут графіка може зберігатися в отдельных.gif и.jpg файлах і завантажуватися у процесі роботи учня з курсом.

Матеріал на сторінці також може містити гіперпосилання на інші сторінки, які завантажуються або вже у існуюче вікно, або додаткове. Крім того, можна встановлювати гіперпосилання на документи, які відкриваються в інших додатках - наприклад, файли у форматі .pdf, .doc і т.д. Сторінки, у свою чергу, складаються з 2-х шарів: заднього (фонового) та переднього плану. Кожен із шарів має власні об'єкти, які мають як власним "уявленням" (зображенням) на екрані, так і своєю "поведінкою" (сценарієм). Стандартні об'єкти, запозичені з ресурсів ToolBook і пов'язані з оформленням сторінок додатка, мають зумовлені зображення на екрані, структуру та властивості, які успадковані від об'єктів самої системи.

У вбудованій бібліотеці (каталозі) є великий набір готових елементів - текстові поля, кнопки, графічні заготовки, інтерактивні питання, анімовані об'єкти, тригери, що управляють, активізують роботу окремих елементів, і

т.п., які дозволяють швидко скомпонувати мультимедійний вміст на сторінці.

Найчастіше для складання навчальних курсів зазвичай буває цілком достатньо об'єктів, що у каталозі середовища. Властивості, взяті з бібліотеки об'єктів, розробник можна змінювати в процесі компонування сторінки та адаптувати до вимог своєї програми. Якщо ж цього виявляється недостатньо і об'єкти все ж таки не повною мірою задовольняють встановленим вимогам, то в ToolBook Instructor є можливість написання скриптів на об'єктно-орієнтованій мові сценаріїв Openscript, який є дуже потужним засобом програмування інтерактивності.

Спеціально розроблені скрипти описують реакції об'єкта деякі події, визначають поведінку кожного об'єкта сторінках проекту. Як такі події можуть бути призначені будь-які дії під час роботи з додатком. Найчастіше це клацання кнопки миші, натискання клавіші та ін. За допомогою скрипту можна змінити інтерактивну поведінку об'єкта, створити анімацію або задати певну послідовність появи об'єктів на сторінці. Слід зазначити, що мова OpenScript дозволяє не тільки описувати реакцію на події для окремих об'єктів, але й створювати програми обміну повідомленнями між об'єктами на одній або різних сторінках.

Таким чином, скрипти значно розширюють можливості даного інструментального середовища. У цьому успішні рішення впроваджуються й інші розробки, тобто. вибудовується деяка лінійка товарів єдиного іміджу, який згодом називається "фірмовим"

Середовища розробки програмного забезпечення - це комбінація програмного забезпечення, призначеного для написання (створення) програмних продуктів. -Середовище розробки включає: компілятор, інтерпретатор, відладчик, засоби автоматизації складання і текстовий редактор[17].

Компілятор - це програма, яка зчитує вихідний код, написаний програмістом, і перетворює їх у програму.

Інтерпретатор – це програма, яка зчитує команди у вихідному коді та негайно їх виконує.

Коли всі перераховані вище компоненти присутні в середовищі розробки програмного забезпечення, таке середовище називається інтегрованим. Такі середовища розробки збільшують швидкість та зручність розробки за рахунок: автоматизації, можливості проводити повний цикл створення та розробки програмного забезпечення[18].

Зазвичай середовище розробки програмного забезпечення призначене для розробки лише однією мовою програмування. І таке середовище розробки, як інтегроване, дає творцеві право вибору мови програмування для розробки, зручного для розробника (з мов, що підтримуються цим середовищем).

Для реалізації управління знаннями в освітніх проектах біло обрано середу розробки Microsoft Visual Studio.

Microsoft Visual Studio - одне з інтегрованих середовищ розробки, розроблена на C++ та C#, що підтримується ОС Windows. Це середовище розробки перекладено десятима мовами.[19] У Visual Studio творець може розробляти веб-сайти, веб-служби, писати консольні програми та програми з графічним інтерфейсом. VS також підтримує різні надбудови. Найбільш відомі доповнення: ReSharper (пошук помилок у коді при написанні програмного коду розробником перед компіляцією); Visual Assist (на відміну від ReSharper також підтримує C++); AnkhSVN (використовує систему контролю версій Visual Studio під назвою Subversion).[20]

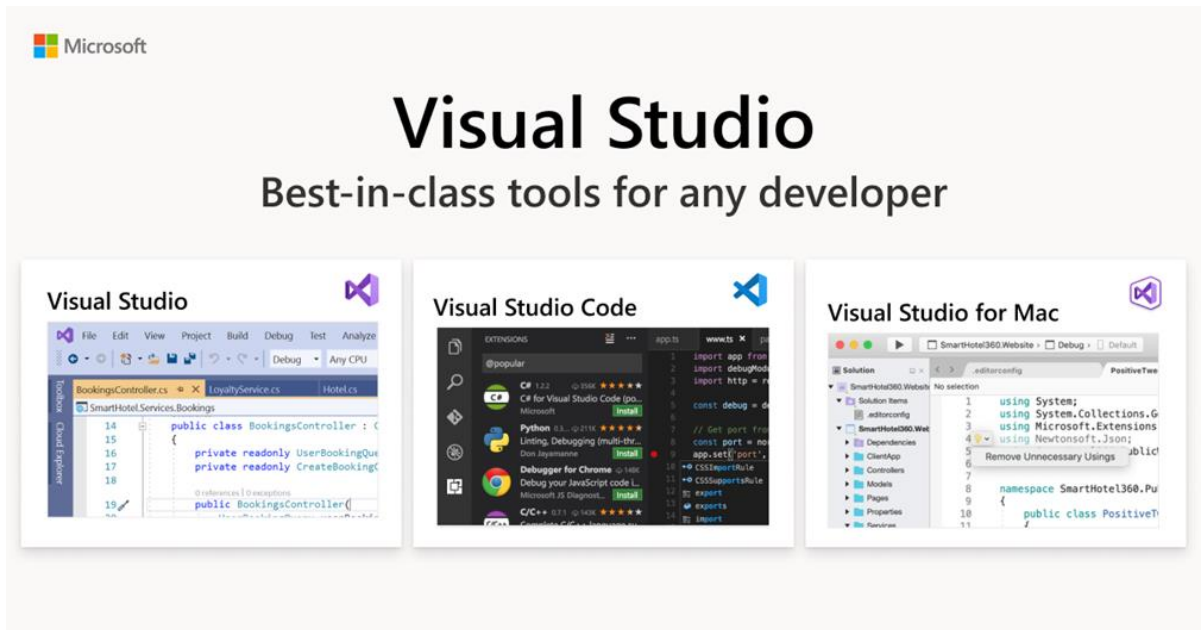


Рисунок 3.1 Visualstudio

Середовище VisualStudio 2019 дозволяє розробляти програми, використовуючи різні мови програмування: Visual C#, Visual Basic, Visual F#, Visual C++, Python і т.д. Також є можливість розробляти програми не тільки під Windows, а й під інші популярні платформи: Android, iOS[21].

Створення програмного продукту відбувається з дотриманням певних правил, які встановлюються під час виборів однієї з моделей розробки програмного забезпечення. Задля реалізації поставленого завдання було обрано каскадна модель (водоспадна модель). Під моделлю зазвичай мається на увазі структура, що визначає послідовність виконання та взаємозв'язку процесів, дій та завдань протягом життєвого циклу. Схема каскадного процесу розробки програмного забезпечення зображено на схемі [22].

Завдяки стрімкому розвитку інформатики та її проникненню в різні галузі народного господарства слово «алгоритм» стало найпоширенішим і вживаним поняттям у житті переважної більшості фахівців. І, з переходом до інформаційного суспільства, алгоритми стають одним із найважливіших елементів цивілізації [23].

Як ми всі знаємо, математична теорія алгоритмів була повністю розроблена зі швидким розвитком інформатики та комп'ютерних технологій, і глибина математичної логіки, здається, вирішує її проблеми. По-перше, це мало великий вплив на світогляд математиків і їх науку.

Проте взаємодія теоретичних галузей, пов'язаних з обчислювальною технікою та теорією алгоритмів, також не викликає сумнівів.

Алгоритмічна теорія вплинула на теоретичне програмування. Зокрема, моделі обчислювальних автоматів відіграють важливу роль у теоретичному програмуванні, що фактично є обмеженням для тих репрезентативних обчислювальних моделей у ранній теорії алгоритмів [24]. Програми інтерпретуються як обчислювальні об'єкти, оператори (послідовне виконання, розгалуження, повторення), що використовуються для складання структурованих програм, запрограмованих відповідно до теорії алгоритмів. Наприклад, протилежний ефект доводить, що потрібно створювати та розвивати теорію обчислювальної складності алгоритмів. Тому можна сказати, що теорія алгоритмів застосовна не тільки до інформатики, а й до інших галузей знань.

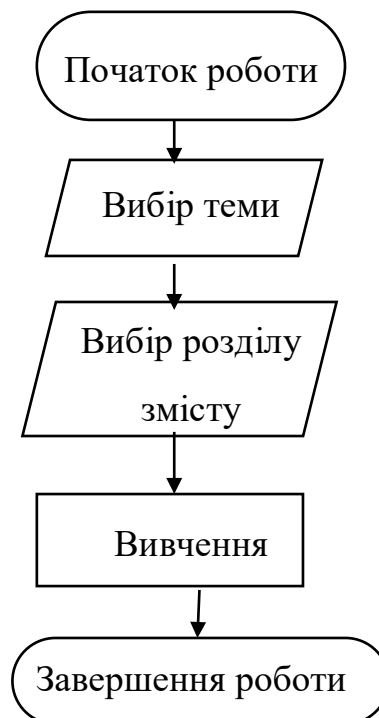
Поняття алгоритму та його властивості

Слово «алгоритм» означає «процес або набір правил, яких слід дотримуватися під час обчислення або іншої операції вирішення проблеми». Таким чином, алгоритм відноситься до набору правил/інструкцій, які поступово визначають, як потрібно виконувати роботу для отримання бажаного результату [25].

Це можна зрозуміти на прикладі приготування за новим рецептом. Щоб приготувати новий рецепт, вам потрібно прочитати інструкції та кроки та виконувати наведений порядок один за одним. Отриманий таким чином результат – ідеально приготована нова страва. Подібним чином алгоритми можуть допомогти вам із завданнями програмування, щоб отримати бажані результати [26].

Розроблені алгоритми є мовними агностиками, тобто це прості інструкції, які можна реалізувати будь-якою мовою, але результати будуть такими ж, як очіувалося[27].

В програмі управління знаннями в освітніх проектах було розроблено наступний алгоритм (спрощена схема)



3.2. Розробка та опис функціоналу власного продукту

Портальне рішення управління проектами – це інформаційне середовище, яке забезпечує:

управлінцям – своєчасне надання звітів та візуалізацію ходу проекту загалом;

користувачам – уніфікований доступ до інформації щодо програми та проектів, що дає можливість зручної спільної роботи.

Перевага портальної технології в тому, що вона дозволяє імпортувати та експортувати дані з різних систем та представляти їх в єдиному форматі за допомогою Web-інтерфейсу.

Було створено портальне рішення, яке забезпечує всебічне управління завданнями та програмами в рамках єдиної інтегрованої системи управління проектами. Для організації більш ефективної роботи було розроблено інтуїтивно-зрозумілий для користувача інтерфейс, який простий та зручний у роботі.

Також інтерфейс пропонує ряд уявлень, у яких можна побачити всю інформацію про проект. Наприклад, Діаграма Ганта є зображенням календарного графіка завдань у проекті, яка дозволяє візуалізувати і контролювати тривалість цих завдань.

Впровадження порталу дозволить співробітникам організувати роботу з управління усіма бізнес-процесами в рамках управління проектами, моніторингу та контролю виконання поточних завдань, аналізу та перегляду аналітичних звітів у розрізі індикаторів ефективності.

Портальне рішення включає наступні модулі:

Управління Програмою

Управління програмою включає такі функції:

Створення та заповнення всіх необхідних атрибутів картки програми

Прив'язка до програми індикаторів, якими оцінюватиметься ефективність цієї програми. У картці програми для індикатора Користувач може встановити базове та планове значення індикатора в розрізі необхідних періодів. Планові значення вказуються всім періодів, якими має збиратися індикатор. Планове значення індикатора може бути задано для різних рівнів збору: на рівні програми, сфери, підписаного договору з виконавцем. Задавши планове та базове значення, користувач надалі після

завершення програми користувач матиме можливість відстежити різницю запланованого та досягнутого значення. Побудувати тренд зміни величини індикатора від періоду до періоду, сформувавши звіт.

Управління звітами за програмою, формування звіту для глобального фонду за ключовими індикаторами, вказаними на картці програми. Формування звіту за програмою здійснюється наприкінці періоду, після завершення програми. У звіті прописано значення за індикаторами, які вказані у картці програми.

Управління завданнями та підзавданнями в рамках програми. У рамках управління завданнями здійснюється виконання схожих з програмою функцій, таких як прив'язка індикаторів, що характеризують ефективність виконання даного завдання, формування звіту, подання задачі та пов'язаних з даним завданням підзадач у вигляді Діаграми Ганта, управління договорами по задачі та звітами по кожному етапу договору в розрізі необхідних індикаторів.

Управління договорами. Управління договорами має на увазі під собою ведення та зберігання договорів. Картка Договору є сполучною ланкою між карткою програми та карткою завдання. В атрибутах картки договору прописуються, у тому числі й ресурси, які виконуватимуть роботи з виконання цієї програми. Користувач системи має можливість потрапляти до картки договору як з Картки програми, так і з Картки Завдання.

Звіт по етапу договору формується наприкінці завершення кожного окремого етапу договору. У звіті прописуються значення за індикаторами, які вказані у картці програми та у картці завдання. Індикатори, що характеризують ефективність виконання договору, збираються з індикаторів, які прописані в картках завдань, підзадач, заходів, які повністю завершені до кінця цього етапу.

При прив'язці індикаторів до звіту користувач задає частоту розрахунку індикатора і прив'язує цей розрахунок або до цілого періоду етапу, або до тимчасової шкали (раз на тиждень, раз на місяць, раз на квартал тощо). Це дозволяє отримувати точні значення індикаторів без порушення математичної логіки їхнього розрахунку.

Керування індикаторами. Управління індикаторами ефективності реалізації складає всіх рівнях – лише на рівні програми, завдання, підзавдання, етапу договору. У системі існує поняття картки індикатора – на якій прописується вся необхідна інформація про індикатор, логіку його розрахунку та методологію збору. Далі при закладенні картки програми, завдання, підзавдання або картки етапу договору з довідника індикаторів вибирається потрібний індикатор і прив'язується до картки.

Управління закупівлею. Модуль управління закупівлями включає наступні розділи:

Управління довідником товарів (даний розділ описано вище)

Управління довідником складів

Відстеження запасів на складах

Управління замовленнями постачання

Управління переміщенням товарів між складами

Управління списанням товарів

Управління звітами

Формування необхідних звітів складає всіх рівнях – програми, завдання, підзавдання, етапу договору.

Завдання, що вирішуються порталом, можна систематизувати наступним чином:

Створення уніфікованої бази даних індикаторів, заходів та матеріалів, що належать до програми;

Здійснення збору, зберігання та обробки інформації про хід реалізації програми та досягнення планових показників та її верифікацію;

Створення уніфікованої бази даних індикаторів, заходів та матеріалів, що належать до програми;

Здійснення збору, зберігання та обробки інформації про хід реалізації Програми та досягнення планових показників та її верифікацію;

Формування звітів про хід реалізації програми в регіонах та на Федеральному рівні, звітів щодо проведеної оцінки програми та інших звітів, необхідних для управління Програмою.

Переваги

Впровадження порталу дозволить:

- створити єдиний інформаційно-управлінський простір, який інтегрує роботу всіх територіально-віддалених учасників процесу;
- скоротити трудомісткість взаємодії всіх учасників процесу обліку та узгодження звітів за програмою, завданнями, підзавданнями, етапами договорів;
- забезпечити сучасний рівень управління інформацією, гарантувати її достовірність та актуальність, скоротити час на її пошук;
- створювати звіти, які є ефективним інструментом аналізу інформації та прийняття рішень на її основі;
- зробити роботу користувачів набагато простіше за рахунок зрозумілого та зручного інтерфейсу.

Розробка структури даних

У системі управління знаннями в освітніх проектах застосовується СУБД MySQL. База даних складається з таблиці quiz.

Усі дані зберігаються на локальному сервері в окремих базах даних

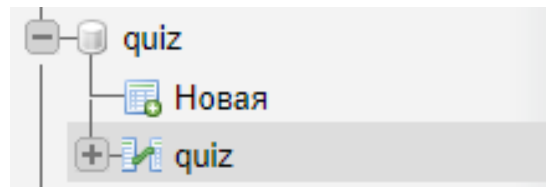


Рис. 3.2 - Структура БД(Загальна)

Нижче описано структуру БД, яка була створена для додатку

+ Options																	
	id	fotourl	sername	name	test1	test2	test3	test4	test5	test6	test7	test8	test9	test10	login	pass	
<input type="checkbox"/>	1	1.jpg	Іванов	Іван	0	1	0	0	0	0	0	0	0	0	0	1	1
<input type="checkbox"/>	2	2.jpg	Петров	Петр	0	0	1	0	0	0	0	0	0	0	0	2	2
<input type="checkbox"/>	3	3.jpg	Демідова	Тетяна	0	1	0	0	0	0	0	0	0	0	0	3	3
<input type="checkbox"/>	4	4.jpg	Неандерталець	Неандерталець	0	1	1	0	0	0	0	0	0	0	0	4	4
<input type="checkbox"/>	5	5.jpg	Сидоренко	Максим	1	1	1	0	0	0	0	0	0	0	0	5	5
<input type="checkbox"/>	6	10.jpg	Іванов	Іван	1	1	0	0	0	0	0	0	0	0	0	6	6
<input type="checkbox"/>	7	9.jpg	Іванов	Іван	1	0	1	0	0	0	0	0	0	0	0	7	7
<input type="checkbox"/>	8	10.jpg	Іванов	Іван	0	0	0	0	0	0	0	0	0	0	0	10	10

Рис. 3.3 - Структура БД(quiz)

Підключення до бази даних здійснюється за допомогою функції які знаходяться у додатку А.

Написання програмного продукту

Перед початком роботи було підключено такі бібліотеки для роботи з програмою.

Далі необхідно було реалізувати завантаження клієнтів(викладачів, учнів) із бази даних.

Для запуску програми з механізмів управління знаннями в освітніх проектах необхідно перейти за наступним шляхом і запустити програму quiz(рис. 3.3)

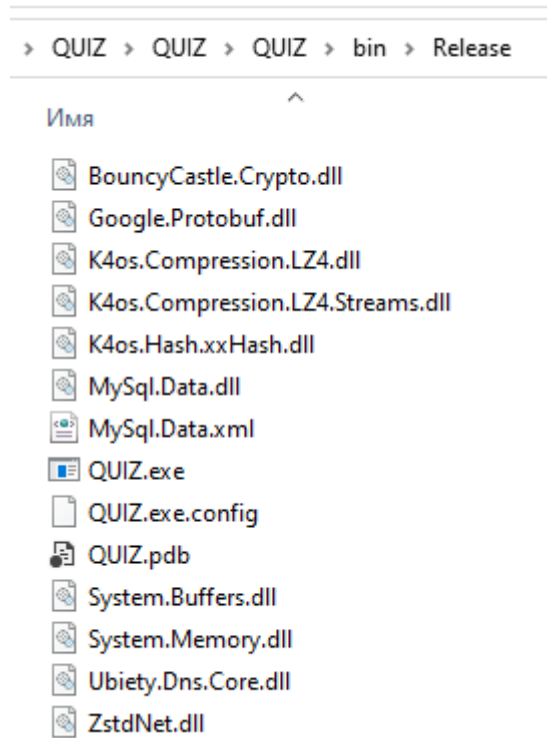


Рис. 3.3 -

Запустивши програму, перед вами з'явиться наступне вікно (рис. 3.4)

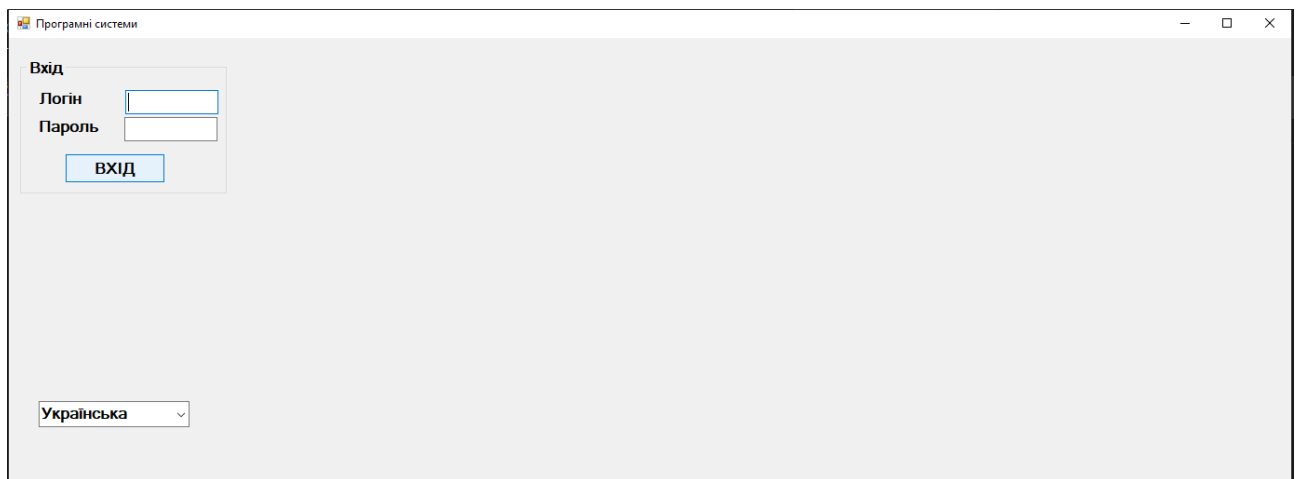


Рис. 3.4 – Стартове вікно програми

Для реалізації було зроблено вхід з логіном «12345» та паролем «12345», в іншому випадку ви зайдете як звичайний користувач. Ввівши

ЛОГІН та пароль, ви можете увійти як звичайний користувач (рис. 3.5) так і як адміністратор (рис. 3.6).

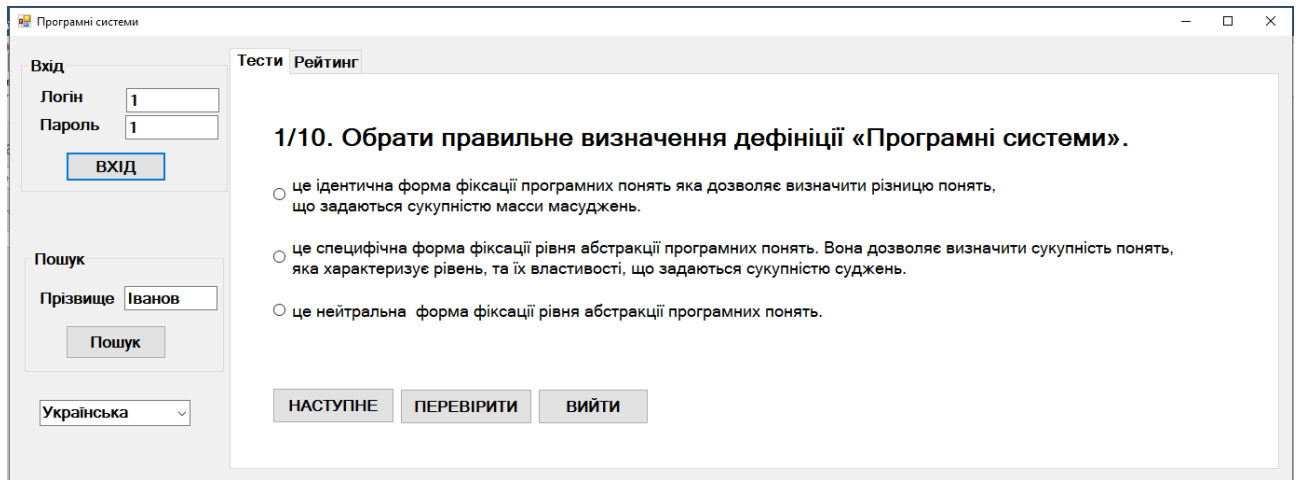


Рис. 3.5.1 – Вид програми звичайного користувача

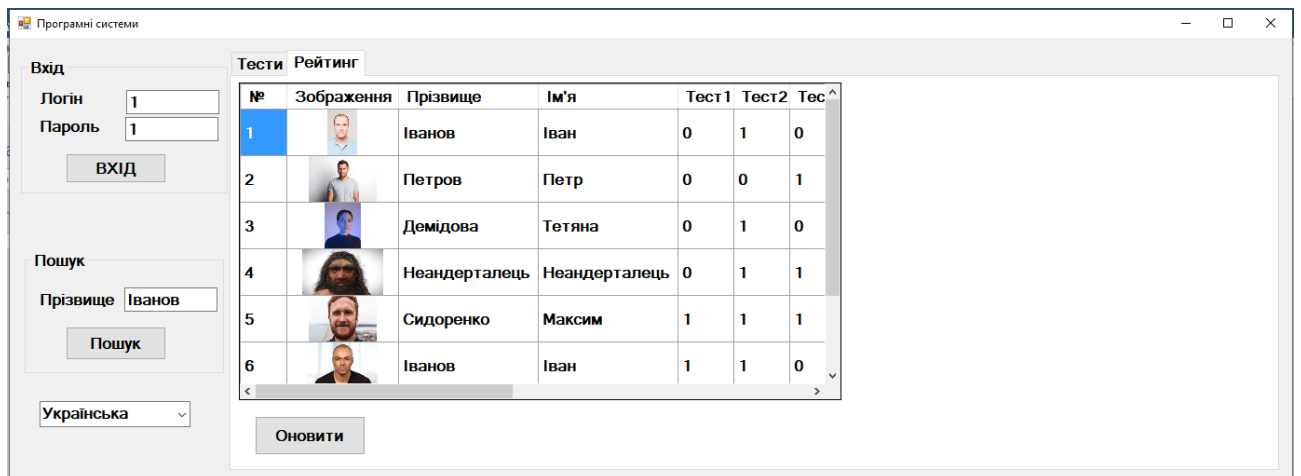


Рис. 3.5.2 – Вид програми звичайного користувача

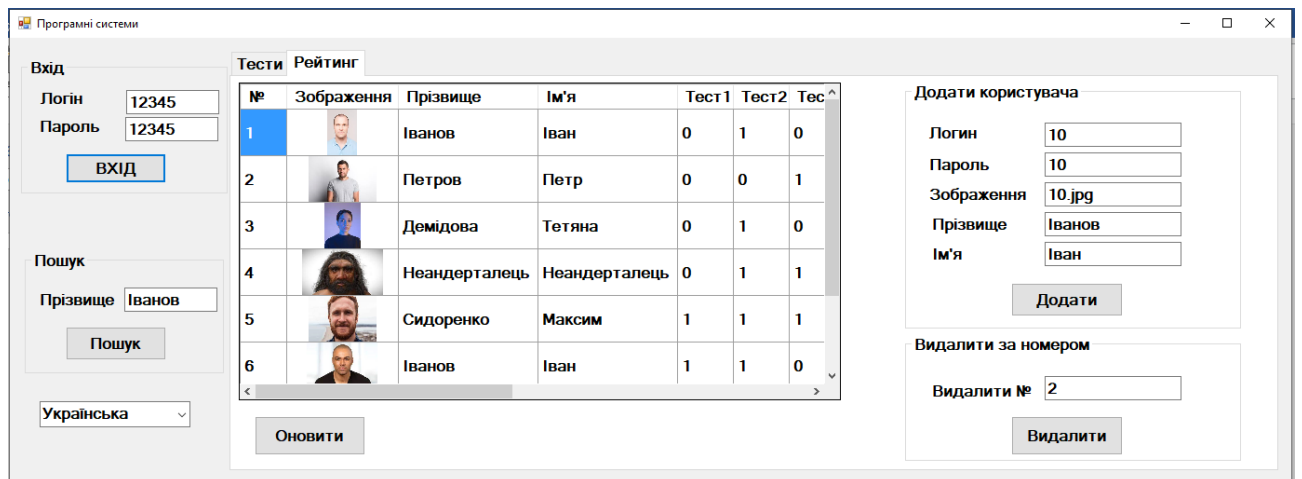


Рис. 3.6 – Вид програми адміністратора

Також була реалізована функція для пошуку клієнта(учень) за прізвищем.

Для пошуку користувача необхідно ввести прізвище, і пошукова система знайде відповідних клієнтів (рис. 3.7)

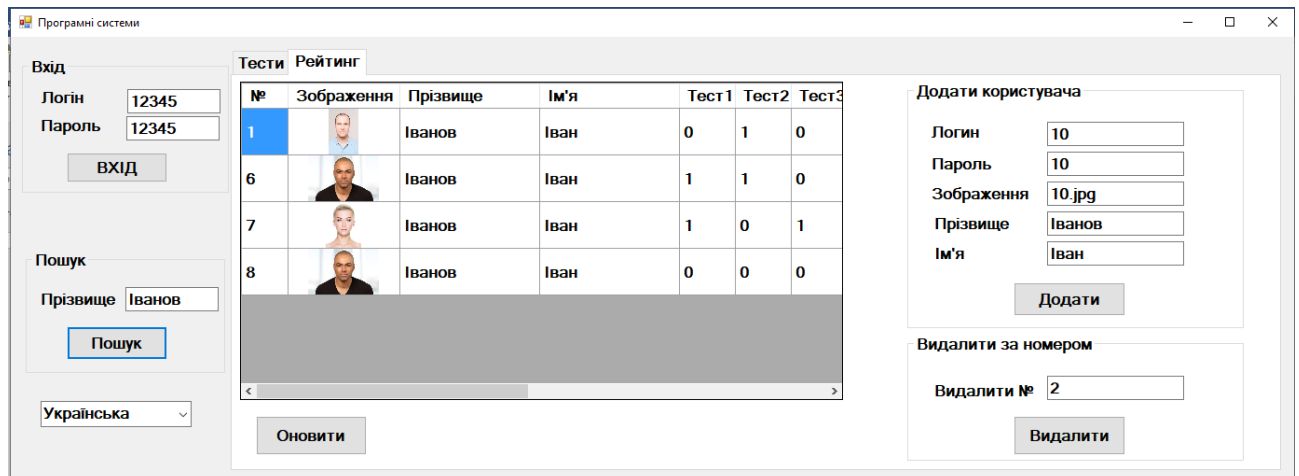
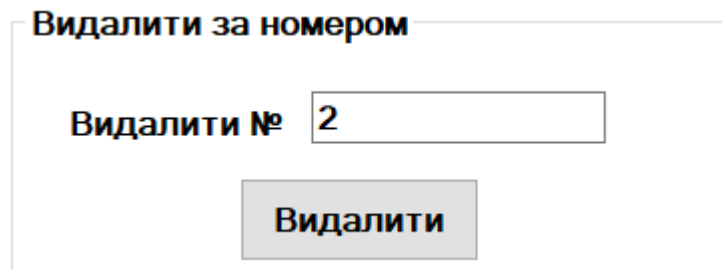


Рис. 3.7 – пошук користувача

Наступним кроком було реалізація видалення клієнта із БД.

Для видалення клієнта з бази даних необхідно ввести номер користувача, який хочете видалити у відповідне поле (рис. 3.8)

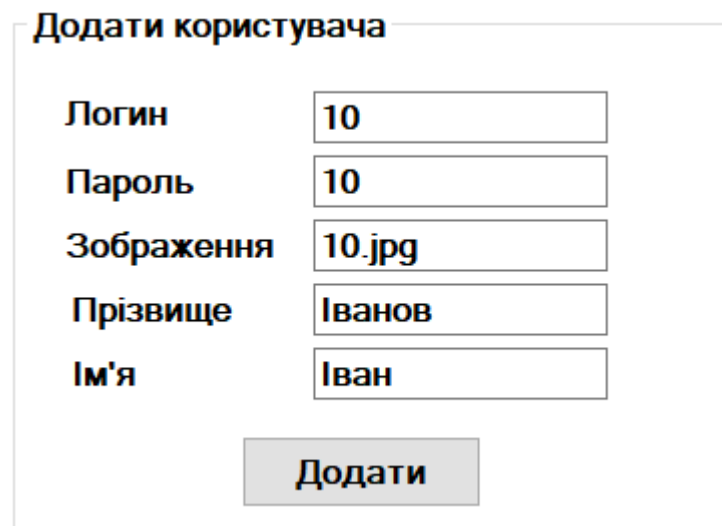


Видалити за номером

Видалити №

Рис. 3.8 – Видалення користувача

Для додавання користувача необхідно заповнити поля та натиснути на кнопку «Додати» (рис. 3.9)



Додати користувача

Логин

Пароль

Зображення

Прізвище

Ім'я

Рис. 3.9 – Додавання клієнта

3.3. Тестування

Завдання етапу - забезпечити якість продукту: стійкість до відмов, надійність, безпека.

Оскільки існує безліч пристроїв, операційних систем та екранів, а також різні конфігурації встановленого ПЗ (додатків) та експлуатаційних характеристик пристроїв, запобігти помилкам заздалегідь неможливо. Тому зазвичай у процес створення програми включається тестувальник, який

опрацьовує різні сценарії роботи програми і пропускає їх у реліз лише тоді, як у ньому відсутні значні, з погляду специфікації, дефекти.

Тестування передбачає перевірку якості системи фахівцем, який брав участь у безпосередній розробці ПЗ, що підвищує ймовірність виявлення дефектів ще процесі розробки. А вартість години такого фахівця значно нижча за вартість години розробника. Таким чином, тестування здешевлює та прискорює процес, підвищуючи якість кінцевого продукту.

Результатом етапу може бути звіт з деталізацією виявлених багів, їх критичності та подальшими виправленнями.

Щодо термінів і вартості, зазвичай йде не менше 40 людиногодин (1 тижні)

Під час тестування програми механізмів управління знаннями в освітніх проектах, всі помилки, що виникали, були виправлені. Нижче наведено приклади тестування програми.

Програмні системи

Enter

Login 12345

Password 12345

Enter

Search

Surname Ivanov

Search

English

Tests Rating

2/10. Functionally nominative system:

- data algebra: V is the carrier of this algebra, which are specified by names from K using the dec mapping
- the algebra of functions are given names from Z using the bek mapping
- data algebra: D is the carrier of this algebra, and algebra operations are functions, which are specified by names from Z using the den mapping

NEXT CHECK EXIT

Рис. 3.10 – тестування програми

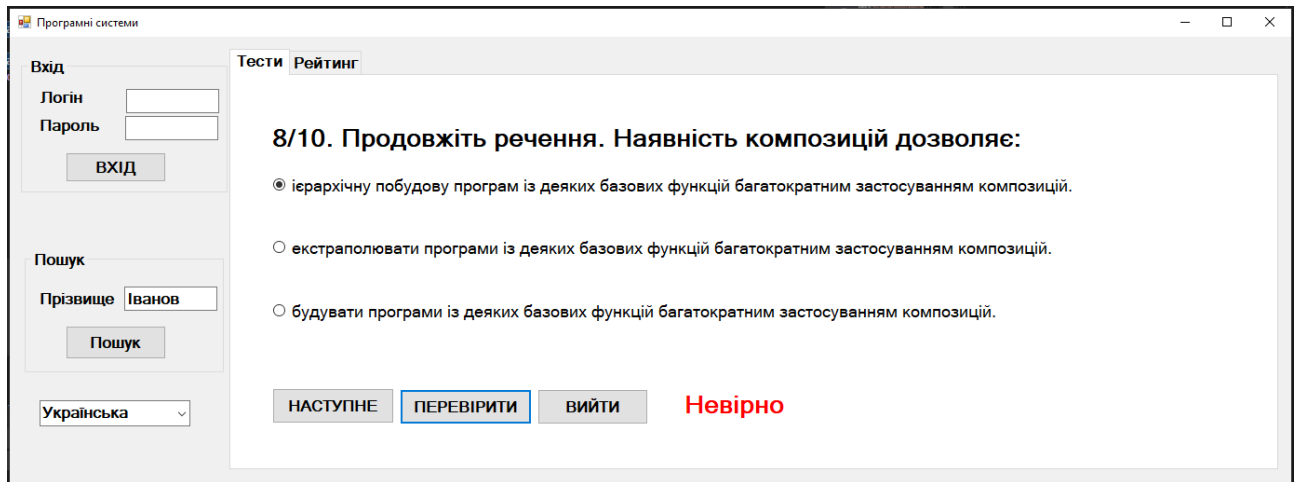


Рис. 3.11 – тестування програми

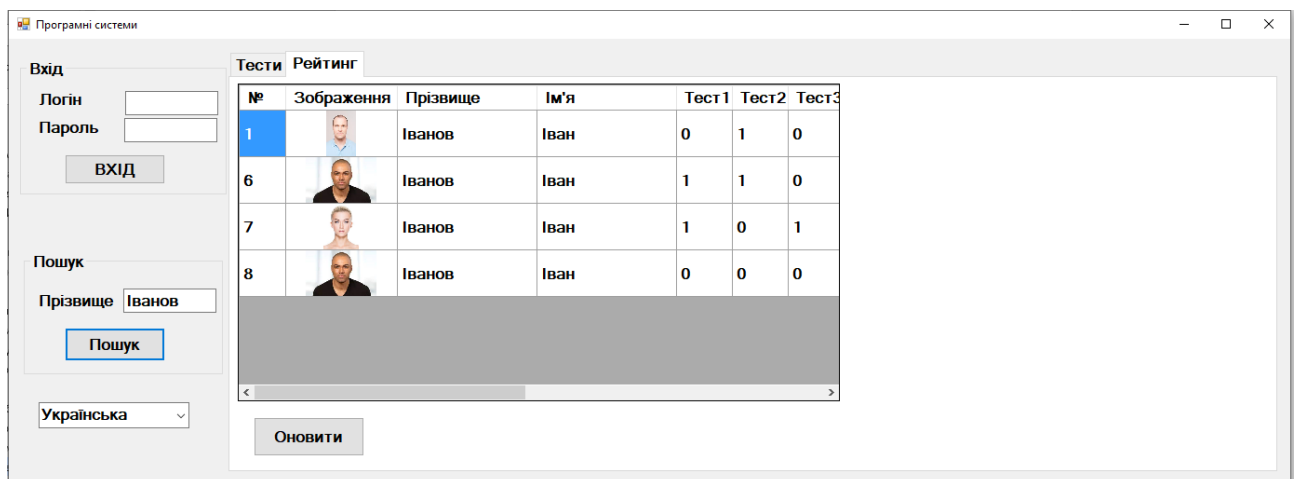


Рис. 3.12 – тестування програми

Висновки до розділу

Цей розділ охоплює розробку програмних алгоритмів, процес розробки програми з управління знаннями в освітніх проектах, розробку структур даних і написання самого продукту.

Структура даних — це спосіб організації та зберігання даних для ефективного виконання операцій. Доступ, вставка, видалення, пошук і сортування даних є одними з основних операцій, які можна виконувати за допомогою структур даних.

Він також описує функціональні можливості програми та процедури тестування. Загалом програма показує хороші результати, і можна працювати далі.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Аналіз потенційної небезпеки

Нижче перераховані основні потенційні ризики під час роботи в офісі:

- небезпека ураження електричним струмом через недотримання правил електробезпеки або несправність електрообладнання;
- порушення функцій опорно-рухового апарату внаслідок тривалих статичних навантажень при роботі з ПК.
- стреси і психічні перевантаження через постійні контакти з клієнтами, колегами, керівництвом при вирішенні робочих проблем, які можуть носити конфліктний характер і призводити до емоційного дискомфорту, внутрішнього роздратування, емоційної нестійкості і неврологічних розладів;
- незадовільна ергономіка робочого місця внаслідок непродуманого планування робочого місця, що може призвести до механічних ушкоджень, ураження електричним струмом та порушень опорно-рухового апарату;
- негативний вплив на зір і продуктивність працівників через недостатнє освітлення робочих зон через несправність освітлювального обладнання або неправильну конструкцію системи освітлення;
- негативний вплив на здоров'я працівників через незадовільні параметри повітряного середовища в робочій зоні внаслідок неправильного проектування системи вентиляції або її несправності;
- негативний вплив підвищених рівнів шуму на психоемоційний стан працівників, що пов'язано з використанням застарілої

периферії, кондиціонерів, копіювальної техніки, освітлювального обладнання;

- небезпека виникнення пожежі внаслідок несправності електрообладнання, несправності обслуговуючого персоналу або порушення правил пожежної безпеки, що може призвести до пожежі.
- неправомірна поведінка персоналу в аварійних ситуаціях.

Заходи безпеки

Офісні приміщення, в яких знаходяться працівники сфери управління персоналом, відносяться до місць, що не підвищують ризик ураження електричним струмом.

Обладнання, що використовується в цих місцях, є споживачем, живиться від мережі змінного струму 220 В і відноситься до закритих електроустановок до 1000 В. Ступінь захисту згідно ДСТУ 12.2.007.0-75* (2001) і II (обладнання освітлення, кондиціонування повітря, теплове обладнання, портативні комп'ютери, сканери).

Згідно з Правилами улаштування електроустановок (далі - ПУЕ) виконуються такі комплекси заходів з електробезпеки:

Конструктивні заходи запобігають випадковому доторканню до струмоведучих частин за допомогою ізоляції та захисних кожухів.

У засобах захисту II класу за ДСТУ 12.1.009-76 (1999) використовується подвійна ізоляція - електрична, включаючи робочу і додаткову ізоляцію.

Відповідно до НПАОП 40.1-1.32-01 Правила улаштування електроустановок. Електрообладнання для спеціальних установок» Більшість офісних приміщень відноситься до категорії пожежної небезпеки П-Па (місця з твердими горючими речовинами), тому передбачений

ізоляційний захист обладнання IP44. Орієнтовні та конструктивні заходи. Призначений для захисту від ураження електричним струмом при торканні металевих корпусів, які можуть опинитися під напругою в разі аварії.

Експлуатація електроустановок та електрообладнання повинна здійснюватися згідно з НПАОП 40.1-1.01-97 Правила безпечного користування електроустановками (далі - ПБЕЕ) та НПАОП 40.1-1.21-98 Правила безпечного користування електроустановками.

Щоб запобігти статистичному навантаженню під час використання ПК, рекомендується 10-хвилинна перерва в роботі. Кожні дві години. Синдром зап'ястного каналу, або синдром зап'ястного каналу, може бути наслідком хронічної травми і виникає у людини в результаті тривалої роботи з мишами: постійна напруга і тиск викликають мікротравми, а прилеглі навколишні тканини здавлюють нерви, що викликає набряк.

Щоб тунельний синдром вас не турбував, необхідно дотримуватися кількох правил організації робочого місця:

- оптимальна висота клавіатури від землі 65-75 см;
- надати вам ергономічну та зручну мишу та клавіатуру; - Відрегулювати висоту та нахил клавіатури (відстань від робочого столу до середини клавіатури не перевищує 30 мм, а кут підйому клавіатури від 2° до 15°). °);
- клавіатура має підлокітники;
- наявність килимків для миші із захистом від тунельного синдрому (спеціальні виступи забезпечують правильне положення руки);
- надати стілець або крісло з підлокітниками;

Існують також певні правила, яких слід дотримуватися при використанні миші та клавіатури. Під час друку рука має бути зігнута під прямим кутом (90°) у лікті, а під час роботи з мишкою слідкуйте за тим, щоб кисть була прямою та розміщена на столі якомога далі від краю. До

речі, час роботи за комп'ютером потрібно обмежити тим часом, який дійсно потрібен.

Для профілактики тунельного синдрому супернику потрібно виконувати спеціальні вправи - чим частіше, тим краще. Ці вправи допоможуть поліпшити кровообіг м'язів і розтягнути їх. Цей комплекс вправ повторюється приблизно кожні 45 хвилин, кожна вправа триває 1-2 хвилини.

Нервові напруження впливає на серцево-судинну систему, підвищуючи артеріальний тиск і пульс, а також на терморегуляцію і емоційний стан організму працівника. У попередженні втоми працівників особливу роль відіграють професійний відбір, організація робочого місця, правильна робоча поза, трудовий ритм, раціоналізація трудового процесу, використання емоційних стимулів, виконання раціонального режиму праці та відпочинку тощо. Боротьба з втомою зводиться насамперед до підвищення гігієни виробничого середовища (усунення забруднення повітря, шуму, вібрації, нормалізація мікроклімату, розумне освітлення тощо).

Крім того, застосовуються спеціальні методи профілактики втоми працівників, у тому числі методи відновлення функціонального стану органів зору і рухової системи, зменшення енергетичних розладів, посилення мозкового кровообігу, оптимізації розумової діяльності.

ДСТУ ISO 9241-1:2003 «Ергономічні вимоги до використання відеотерміналів в офісах» встановлює загальні ергономічні вимоги. частина 1. Загальні положення» Організація робочого місця включає: правильне розміщення робочого місця у виробничій зоні. Розумна робоча позиція з урахуванням виробництва меблів.

Антропометричні характеристики людини; раціональне розташування обладнання робочого місця; врахування характеру та особливостей трудової діяльності.

У зв'язку з впровадженням технологічної трансформації на основі інформаційних технологій у приміщенні, розташованому на другому поверсі виробничого корпусу, планується встановити комп'ютер. Визначимо, скільки комп'ютеризованих робочих місць, обладнаних дисплейними терміналами (ВДТ), можна встановити в цьому приміщенні та як їх облаштувати відповідно до встановлених норм і правил охорони праці. Розміри приміщення: довжина $a = 5,1$ м, ширина $b = 4,1$ м, висота $h = 2,5$ м.

Перше, що потрібно зробити - це проаналізувати, чи підходить приміщення для комп'ютеризованого робочого місця.

Відповідно до НПАОП 0.00-1.28-10 «Правила охорони праці при експлуатації електронно-обчислювальних машин» не допускається розміщення приміщень для використання ВДТ у підвальних і цокольних поверхах. Також заборонено встановлювати заводський комп'ютер (ПК), а також над або під такими приміщеннями. Крім того, виробничі майданчики з використанням ВДТ не повинні примикати до майданчиків, де рівень шуму та вібрації перевищує допустимі значення.

4.2. Промислова гігієна та заходи гігієни праці

Відповідно до ДБН В.2.5-28-2006 "Інженерне обладнання будинків і споруд. Природне та штучне освітлення" В офісах використовується природне та штучне освітлення. Природне освітлення здійснюється через світлові вікна, забезпечуючи коефіцієнт природного освітлення (КПО) не менше 1,5%. Для запобігання попаданню прямих сонячних променів, які створюють прямі та відбиті відблиски на поверхнях екрана та клавіатури, використовуються сонцезахисні пристрої, а на вікна встановлюються жалюзі або штори. Штучне освітлення приміщення забезпечується загальною рівномірною системою освітлення. Освітленість поверхні робочого столу, на якій розміщена область для документів, становить 300-

500 лк. В якості джерел штучного освітлення приміщень використовуються люмінесцентні лампи типу ЛБ.

Показники мікроклімату робочих місць відповідають ДСН 3.3.6-042-99 «Гігієнічні норми мікроклімату промислових об'єктів», ДСТУ 12.1.005-88 (1991) і ГН 2152-80. «Промислові та громадські місця «Гігієнічні нормативи допустимих рівнів іонізації повітря»:

- Робота в об'єкті туристичного обслуговування відноситься до категорії І: - легка, тому встановлені оптимальні значення наступних параметрів мікроклімату:
- у холодні періоди року: температура 21-23 °С, відносна вологість повітря: 40-60%, швидкість руху повітря: 0,1 м/с;
- у теплий період року: температура 22-24 °С, відносна вологість повітря: 40-60%, швидкість руху повітря: 0,2 м/с.

Оптимальні рівні позитивних іонів (n^+) і негативних іонів (n^-) у повітрі приміщень з ПК нормуються згідно з ГН 2152-80 «Гігієнічно-гігієнічні нормативи допустимих рівнів іонізації повітря виробничих і громадських місць» і складають: $n^+=1500-30000$ (шт. на 1 см³); $n^- = 3000-5000$ (шт. на 1 см³). За допомогою біполярного коронного небулайзера забезпечується оптимальна концентрація позитивних і негативних іонів у повітрі робочого місця.

Вимоги цих документів забезпечуються обладнанням приміщень і транспортних засобів системами кондиціонування та вентиляції, дезодорації повітря, опалення згідно з вимогами ДБН В.2.5-67:2013 «Опалення, вентиляція та кондиціонування повітря».

Рівні звукового тиску в октавному діапазоні, рівні звуку та еквівалентні рівні звуку в робочих приміщеннях у закритих приміщеннях устанавлюються згідно з ДСанПіН 3.3.2.007-98 «Національний санітарно-гігієнічний кодекс та специфікація щодо використання терміналів

електронного комп'ютерного зору» та ДСН 3.3.6.037-. 99 «Гігієнічні умови щодо виробничого шуму, ультразвуку та інфразвуку».

Зниження рівня шуму в приміщенні здійснюється за рахунок:

- використовувати більш сучасне обладнання;
- розташування принтерів і різного обладнання колективного користування далеко від робочих місць більшості співробітників;
- переводити жорсткий диск у сплячий режим (очікування), якщо комп'ютер деякий час не працює;
- використовуйте блок живлення ПК з вентилятором на гумовому підвісі.

Гігієнічна класифікація умов праці Для встановлення пріоритету в проведенні оздоровчих заходів використовується «Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу», затверджена наказом МОЗ України від 27.12.2001 р. № 528. Виходячи з принципів Гігієнічної класифікації, умови праці діляться на 4 класи – оптимальні, допустимі, шкідливі та небезпечні (екстремальні).

1 2 Травмонебезпечність (вибухопожежонебезпека, сейсмічна небезпека, небезпека травмування рухомими частинами машин й обладнання, що рухаються) Ступінь небезпеки, бали 3. Естетичні Гармонійність у робочій зоні світлокольорової композиції, звукового середовища Ароматичність запахів повітряного середовища Гармонійність робочих поз і трудових рухів Естетичний рівень*, бали Ступінь ароматичності, бали Конструктивні рішення обладнання робочих місць, бали Траєкторія, ритм і варіантність трудових рухів, бали 4. Соціально-психологічні ** Спорідненість колективу Характер міжгрупових стосунків у колективі Рівень взаємозаміни в процесі праці, товариської взаємодопомоги, дисципліна праці, бали Рівень конфліктності, бали 1 клас – ОПТИМАЛЬНІ умови праці – такі умови,

при яких зберігається не лише здоров'я працюючих, а й створюються передумови для підтримання високого рівня працездатності. Оптимальні гігієнічні норматива виробничих факторів встановлені для мікроклімату і факторів трудового процесу. Для інших факторів за оптимальні умовно приймаються такі умови праці, за яких несприятливі фактори виробничого середовища не перевищують рівнів, прийнятих за безпечні для населення. 2 клас – ДОПУСТИМІ умови праці – характеризуються такими рівнями факторів виробничого середовища і трудового процесу, які не перевищують встановлених нормативів, а можливі зміни функціонального стану організму відновлюються за час регламентованого відпочинку або до початку наступної зміни та не чинять несприятливого впливу на стан здоров'я працюючих та їх потомство в найближчому і віддаленому періодах. 3 клас – ШКІДЛИВІ умови праці – характеризуються такими рівнями шкідливих виробничих факторів, які перевищують норматива і здатні чинити несприятливий вплив на організм працюючого та/або його потомство. Шкідливі умови праці за ступенем перевищення гігієнічних нормативів та вираженості можливих змін в організмі працюючих поділяються на 4 ступені: 1 ступінь (3.1) – умови праці характеризуються такими рівнями шкідливих факторів виробничого середовища та трудового процесу, які, як правило, викликають функціональні зміни, що виходять за межі фізіологічних коливань (останні відновлюються при тривалішій, ніж початок наступної зміни, перерві контакту з шкідливими факторами) та збільшують ризик погіршення здоров'я; 2 ступінь (3.2) – умови праці характеризуються такими рівнями шкідливих факторів виробничого середовища і трудового процесу, які здатні викликати стійкі функціональні порушення, призводять у більшості випадків до зростання виробничо-обумовленої захворюваності, появи окремих ознак або легких форм

професійної патології (як правило, без втрати професійної працездатності), що виникають після тривалої експозиції (10 років та більше); 3 ступінь (3.3) – умови праці характеризуються такими рівнями шкідливих факторів виробничого середовища і трудового процесу, які призводять, окрім зростання виробничо-обумовленої захворюваності, до розвитку професійних захворювань, як правило, легкого та середнього ступенів важкості (з втратою професійної працездатності в період трудової діяльності); 4 ступінь (3.4) – умови праці характеризуються такими рівнями шкідливих факторів виробничого середовища і трудового процесу, які здатні призводити до значного зростання хронічної патології та рівнів захворюваності з тимчасовою втратою працездатності, а також до розвитку важких форм професійних захворювань (з втратою загальної працездатності); 4 клас НЕБЕЗПЕЧНІ (ЕКСТРЕМАЛЬНІ) умови праці – характеризуються такими рівнями шкідливих факторів виробничого середовища і трудового процесу, вплив яких протягом робочої зміни (або ж її частини) створює загрозу для життя, високий ризик виникнення важких форм гострих професійних уражень. Ступінь шкідливості умов праці встановлюється за величиною перевищення граничнодопустимих концентрацій шкідливих речовин; класом та ступенем шкідливості чинників біологічного походження; залежно від величин перевищення чинних нормативів шуму, вібрації, інфра- та ультразвуку; за показником мікроклімату, який отримав найвищий ступінь шкідливості з врахуванням категорії важкості праці за рівнем енергозатрат, або за інтегральним показником теплового навантаження середовища; за величиною перевищення граничнодопустимих рівнів електромагнітних полів та випромінювань; за параметрами радіаційного фактору відповідно до Норм радіаційної безпеки; за показниками природного та штучного освітлення; за величиною недодержання необхідної кількості

іонів повітря і показника їх полярності. Оцінка важкості трудового процесу здійснюється на підставі обліку фізичного динамічного навантаження, маси вантажу, що піднімається і переміщується, загального числа стереотипних робочих рухів, величини статичного навантаження, робочої пози, ступеню нахилу корпусу, переміщень в просторі. Оцінка напруженості трудового процесу здійснюється на підставі обліку факторів, що характеризують напруженість праці, а саме, інтелектуальні, сенсорні, емоційні навантаження, ступінь монотонності праці, режим роботи. Робота в умовах перевищення гігієнічних нормативів (3 та 4 клас умов праці) може бути дозволена тільки при застосуванні засобів колективного та індивідуального захисту і скороченні часу дії шкідливих виробничих факторів (захист часом). Робота в небезпечних (екстремальних) умовах праці (4 клас) не дозволяється, за винятком ліквідації аварій, проведення екстрених робіт для попередження аварійних ситуацій. Ця робота повинна виконуватись у відповідних засобах індивідуального захисту та регламентованих режимах виконання робіт. Санітарно-гігієнічні вимоги до виробничих та допоміжних приміщень

Вимоги до виробничих приміщень. Вибір типу приміщення визначається технологічним процесом та можливістю боротьби з шумом, вібрацією і забрудненням повітря. Виробничі приміщення відповідно до вимог чинних нормативів мають бути забезпечені достатнім природним освітленням. Обов'язковим є являється також улаштування ефективної за екологічними і санітарно-гігієнічними показниками вентиляції. Висота виробничих приміщень повинна бути не менше 3,2 м, а об'єм і площа – 15 м³ та 4,5 м² відповідно на кожного працівника (для користувачів комп'ютерів на одного працюючого повинно бути не менше: площі - 6 м² і об'єму - 20 м³). Приміщення чи ділянки виробництв з надлишками тепла , а також зі значними виділеннями

шкідливих газів, пару чи пилу слід, як правило, розміщувати біля зовнішніх стін будівель, а у багатоповерхових будівлях – на верхніх поверхах. Підлога на робочих місцях має бути рівною, теплою, щільною та стійкою до ударів, мати неслизьку та зручну для очистки поверхню; бути стійкою до дії хімічних речовин і не вбирати їх. Стіни виробничих та побутових приміщень мають відповідати вимогам шумо- і теплозахисту; легкому піддаватись прибиранню та миттю; мати покриття, що виключає можливість поглинення чи осадження отруйних речовин (керамічна плитка, олійна фарба). Приміщення, де розміщені виробництва з виділенням шкідливих та агресивних речовин (кислоти, луги, ртуть, бензол, сполуки свинцю та ін.), повинні мати стіни, стелю та конструкції, виконані і оздоблені так, щоб попереджувалась сорбція (осідання) цих речовин та забезпечувалась можливість очищення та миття цих поверхонь. У приміщеннях з великим виділенням пилу (шліфування, заточка тощо) слід передбачити прибирання за допомогою пилососів чи гідрозмивання. Колір інтер'єрів приміщень має відповідати вимогам технічної естетики.

4.3. Заходи пожежної безпеки

Український «Закон про пожежну безпеку» визначає загальні правові, економічні та соціальні основи забезпечення пожежної безпеки в Україні, регулює взаємовідносини державних органів, юридичних і фізичних осіб у цій сфері незалежно від виду діяльності та форми власності.

Пожежна безпека - це стан об'єкта, при якому з заданою ймовірністю виключається можливість виникнення і розвитку пожежі та вплив факторів її ризику на людей при забезпеченні захисту матеріальних цінностей.

Для забезпечення пожежної безпеки закладу проводиться пожежно-профілактична робота, яка включає комплекс організаційних і технічних

заходів, спрямованих на забезпечення безпеки персоналу, попередження виникнення пожеж, обмеження розповсюдження пожежі, створення умов для успішного гасіння.

Для ліквідації пожеж на ранніх стадіях їх розвитку персонал об'єкта використовує первинні засоби пожежогасіння. До них відносяться: вогнегасники, засоби пожежогасіння (чохла з негорючої ізоляційної тканини, ящики з піском, пожежні відра, лопати, ломи, сокири тощо), автоматичні системи пожежогасіння.

Первинний засіб пожежогасіння, залежно від категорії будинку, може розташовуватися як самостійно, так і в складі пожежного огороження.

Відповідно до ДБН В.1.1.7-2002 «Пожежна безпека будівель» за агрегатним станом і характеристиками горіння різних горючих речовин і матеріалів пожежі класифікуються на відповідні ступені.

В офісному приміщенні дерев'яні меблі, електронне обладнання, паперові носії.

Клас пожежності офісних приміщень (згідно з ДБН В.1.1.7-2002 «Протипожежний захист. Пожежна безпека будівель») - пожежі твердих речовин, переважно органічних, горіння яких супроводжується тлінням (дерева, пластмаси, паперу) визначається як вид. Класифікація об'єкта (згідно НАПБ Б.03.002-2007 «Кодекс визначення класифікації вибухо-, пожежо- та пожежонебезпечних приміщень, приміщень і відкритих споруд») – визначено як клас D. Визначення типу та кількості первинних засобів пожежогасіння В.1.1.7-2002 «Протипожежний захист. Пожежна безпека будівель») - Для адміністративного приміщення площею 20,91 кв.м два порошкових вогнегасника типу ВП- 5 (НАПБ Б.03.002-2007 «Класифікація вибухо та пожежонебезпечності, типові визначення будівель і зовнішніх споруд». ДБН В.1.1.7-2002 «Пожежна безпека будівель»)). Крім того, адміністративні приміщення повинні бути обладнані автоматичними

пожежними сповіщувачами, для боротьби з підвищеною температурою, димом і полум'ям. Наприклад, сповіщувачі для моделей DTL, ITM.

РОЗДІЛ 5. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

5.1. Поняття охорони навколишнього природного середовища, об'єкти, основні принципи та завдання

Охорона навколишнього середовища— раціональне використання природних ресурсів, охорона особливо цінних і унікальних природних комплексів та забезпечення екологічної безпеки. Це комплекс загальнодержавних, адміністративних, правових, економічних, політичних і соціальних заходів, спрямованих на раціональне використання, відтворення та охорону природних ресурсів землі та обмеження негативного впливу людської діяльності на навколишнє середовище.

Охорона навколишнього природного середовища, раціональне використання природних ресурсів та забезпечення екологічної безпеки діяльності людини є необхідними умовами сталого розвитку економіки та суспільства України. З цією метою Україна реалізує на своїй території екологічну політику, спрямовану на охорону безпечного середовища існування живої і неживої природи, захист життя і здоров'я людей від негативного впливу забруднення навколишнього природного середовища, досягнення гармонійної взаємодії суспільства і природи. , охорона, раціональне використання та відтворення природних ресурсів.

Природоохоронні об'єкти:

1. Охороняються та регулюються державою на території України: навколишнє природне середовище як сукупність природних і природних суспільних умов і процесів, природні ресурси, як беруть участь у господарському кругообігу, так і не використовуються в поточний період економіки. (земля, надра, води, атмосфера), ліси та інша рослинність, тваринний світ), ландшафти та інші природні комплекси.

2. Території та об'єкти Фонду природного заповідного фонду України та інші території та об'єкти, визначені законодавством України, перебувають під особливою охороною держави.

3. Здоров'я і життя людей також охороняються державою від негативного впливу несприятливих умов навколишнього середовища.

Принципи охорони довкілля є керівними засадами організації та впливу на суспільні відносини у сфері екологічного права.

Основними принципами охорони навколишнього середовища є:

- пріоритет вимог екологічної безпеки, обов'язковість дотримання екологічних стандартів, норм і обмежень використання природних ресурсів при провадженні господарської, господарської та іншої діяльності;
- екологічно безпечне середовище для захисту життя і здоров'я людей;
- профілактичні заходи охорони навколишнього середовища;
- екологізація матеріального виробництва на основі комплексного вирішення питань охорони навколишнього середовища, використання та 2 відтворення природних ресурсів, широкого впровадження новітніх технологій;
- охороняти просторово-видове різноманіття та цілісність природних об'єктів і комплексів;
- науково узгоджувати екологічні, економічні та соціальні інтереси суспільства на основі міждисциплінарних знань екологічних, соціальних, природничих і технологічних наук і прогнозів стану навколишнього природного середовища;
- обов'язковість надання висновків національних екологічних інспекцій;
- Прозорість і демократичність у прийнятті рішень, реалізація яких впливає на стан навколишнього природного середовища та формує екологічний світогляд населення;
- наукове регулювання впливу господарської та іншої діяльності на навколишнє природне середовище;

- загальне безоплатне та платне використання природних ресурсів для господарської діяльності;
- відшкодування збитків, завданих порушенням законодавства про охорону навколишнього природного середовища;
- вирішення питань охорони навколишнього природного середовища та використання природних ресурсів з урахуванням масштабів антропогенних змін території, сукупної дії факторів, що негативно впливають на екологічну обстановку;
- поєднання заходів стимулювання та відповідальності за охорону навколишнього природного середовища;
- вирішення питань охорони навколишнього природного середовища на основі широкого міждержавного співробітництва;
- Екологічний податок, плата за спеціальне використання води, плата за спеціальне використання лісових ресурсів, плата за користування надрами встановлюються відповідно до податкового законодавства України.

Дотримання встановлених принципів створює необхідні умови для відтворення природних ресурсів, забезпечує екологічну безпеку, запобігає та усуває негативний вплив господарської та іншої діяльності на навколишнє природне середовище, охороняє природні ресурси, генетичний фонд біологічної природи, історія-культура України Спадщина пов'язана з ландшафтами та іншими природними об'єктами.

5.2. Права та обов'язки громадян та органів державної влади щодо охорони навколишнього природного середовища

Важливою ланкою забезпечення якості навколишнього середовища є система моніторингу стану навколишнього середовища, що включає: моніторинг стану навколишнього середовища та прогнозування змін,

виявлення та оцінку джерел забруднення, запобігання явищу підвищення рівня забруднення.

Контроль якості атмосферного повітря в густонаселених районах здійснюється згідно з ДСТУ 17.2.3.01-86, який визначає стаціонарні, маршрутні та пересувні станції контролю забруднення атмосферного повітря.

Стаціонарні станції спостереження призначені для забезпечення безперервної реєстрації забруднюючих речовин (CO₂, CO, пилу тощо) та регулярних проб повітря для подальшого аналізу.

Маршрутні станції призначені для проведення регулярного відбору проб повітря в кількох точках місцевості за тимчасовим графіком.

Розміщення фіксованих і маршрутних колів повинно допомогти виявити максимальні концентрації забруднень. Як правило, стовпці розташовані концентричними колами на перехрестях із радіальними лініями, що показують аспекти світу. Центром кола має бути джерело забруднення.

ДСТУ 17.1.3.07-82 встановлює правила контролю якості води водойм і водостоків, а ДСТУ 17.1.3.08-82 — правила контролю якості морської води.

Контроль якості довкілля є основним завданням Міністерства екології та природних ресурсів України. Контроль за забрудненням атмосфери необхідно проводити у великих містах та інших діючих великих промислових підприємствах, інтенсивних транспортних та інших населених пунктах.

У великих містах України контроль за станом навколишнього природного середовища здійснюється через автоматизовані системи якості атмосферного повітря та води. Наявність постійної інформації про стан навколишнього середовища цих міст дозволяє своєчасно вживати необхідних заходів щодо усунення надлишкового забруднення та

зменшення викидів шкідливих промислових підприємств і транспортних потоків.

Правові заходи забезпечення екологічної безпеки [7].

Екологічна безпека в Україні забезпечується здійсненням комплексу взаємопов'язаних політичних, економічних, технічних, організаційних, національно-правових та інших заходів. Зміст національних правових заходів неоднаковий. Залежно від спрямованості дії їх можна поділити на декілька типів: організаційне запобігання, регулятивне стимулювання, виконавче примусове, відновлення захисту та забезпечення. Вони утворюють особливий правовий механізм, під яким слід розуміти систему національних правових інструментів, спрямованих на регулювання рівня екологічної безпеки, запобігання погіршенню екологічного стану та діяльності, що є небезпечною для населення та природних систем, а також локалізацію прояви екологічної небезпеки.

Організація та профілактичні заходи. Вони спрямовані на визначення територій, територій, об'єктів і видів діяльності, що створюють екологічну небезпеку для навколишнього природного середовища та здоров'я людей, а також здійснення окремих заходів щодо запобігання виникненню екологічної небезпеки. До них належать:

- обліково-підготовчий;
- реєстраційний;
- експертний огляд;
- інформаційно-прогностичний.

Крім того, в Україні розвиваються екологічний аудит та екологічне страхування.

Облікові заходи передбачають виявлення, інвентаризацію та класифікацію небезпечних зон, об'єктів, територій і джерел.

Реєстраційні заходи включають сертифікацію екологічно небезпечних речовин, сертифікацію, підтвердження відповідності, ліцензування та

реєстрацію екологічно небезпечних джерел. Якщо виробляється продукція, яка є шкідливою для навколишнього середовища, вона повинна бути сертифікована. У процесі сертифікації видається сертифікат відповідності, який підтверджує відповідність продукції українським стандартам.

Обов'язкова сертифікація продукції передбачена безпосередньо Законом України "Про захист прав споживачів" від 15 грудня 1993 року. Закон України «Про підтвердження відповідності» від 17 травня 2001 року визначає закони та організаційні принципи підтвердження відповідності продукції, систем управління якістю, систем управління навколишнім середовищем, персоналу та спрямований на забезпечення єдності підтвердження національної технічної політики. у наступних сферах відповідності.

Послідовний облік екологічно небезпечних джерел відповідно до чинного законодавства. Екологічно шкідливі види діяльності підлягають ліцензуванню, у тому числі заходи, спрямовані на регулювання та обмеження екологічно шкідливих видів діяльності шляхом запровадження ліцензійної системи та встановлення ліцензійних умов провадження такої діяльності. Екологічне ліцензування регулюється Законом України від 1 червня 2000 р. «Ліцензування певних видів господарської діяльності», постановою Кабінету Міністрів України від 10 серпня 1992 р. No 459 «Положення про порядок видачі документів дозвільного характеру» Спец. Використання природних ресурсів» та інші нормативно-правові акти.

Третю групу організаційно-профілактичних заходів щодо забезпечення екологічної безпеки складають заходи експертної оцінки. Це екологічне обстеження об'єктів і комплексів (у тому числі військових і оборонних споруд), що становлять екологічну загрозу навколишньому природному середовищу, життю і здоров'ю людей, попередні оцінки екологічного впливу цих об'єктів, громадські слухання та громадські обговорення екологічних питань. шкідливої діяльності, яка передбачається реалізувати.

Екологічні перевірки таких об'єктів регулюються законами України «Про охорону навколишнього природного середовища» (ст. 27), «Екологічні інспекції»

Остання група – інформаційно-прогностичні заходи. До них належать прогнозування, планування, моніторинг, повідомлення та інші заходи, які вважаються функціями управління в екологічній сфері.

Регулювання і стимул. Вони являють собою систему правових норм і правил, призначених для регулювання відносин і забезпечення дотримання пріоритетів, норм, стандартів, обмежень та інших вимог у сфері екологічної безпеки. Відповідно до положень чинного законодавства розробляються наступні екологічні стандарти (ст. 32 Закону про охорону навколишнього природного середовища України); екологічні стандарти (ст. 33); екологічні обмеження; правила проектування та експлуатації небезпечних установок, поводження з ними. з речовинами та джерелами, шкідливими для навколишнього середовища.

5.3. Основні принципи охорони навколишнього середовища

Основними принципами охорони навколишнього середовища є:

- Господарська діяльність, яка впливає на навколишнє середовище;
- Повага до прав людини у сприятливому середовищі;
- Забезпечення умов, сприятливих для життєдіяльності людини;
- Інтегрувати економічні, екологічні та соціальні інтереси держави, суспільства та людей для досягнення сталого розвитку в екологічно безпечному середовищі;
- Раціональне використання природних ресурсів є необхідною умовою забезпечення доброго довкілля та екологічної безпеки;

- Обов'язки державних органів та органів місцевого самоврядування щодо забезпечення екологічної безпеки та створення доброго довкілля;
- Плата за природне користування, у тому числі відшкодування шкоди, завданої довкіллю;
- Контрольна незалежність у сфері охорони навколишнього природного середовища;
- Обов'язковість оцінки екологічного впливу окремих видів діяльності при прийнятті рішень;
- Врахування природних і соціально-економічних особливостей території при провадженні господарської діяльності на території;
- Надати пріоритет захисту природних екосистем;
- Поважати права громадян на достовірність екологічної інформації;
- Нести відповідальність за протиправні дії у сфері охорони навколишнього природного середовища;
- Розвивати систему екологічного виховання та виховання екологічної культури;
- Міжнародне співробітництво України в галузі охорони навколишнього природного середовища.

ВИСНОВКИ

В ході проведення дослідження поставленні цілі та завдання були виконанні, а саме: розробка віртуального навчального посібника; визначення основних етапів розробки програми; оцінка стильових моделей програм; розробка власного дизайну з урахуванням функціональності та стильового наповнення.

Проведено дослідження ефективності використання різних технік для реалізації функціональних і готових до використання додатків. З цією метою аналізувалися переваги програмного забезпечення.

Також розглядається необхідність такої системи в наш час, описується важливість покращення студентського середовища та своєчасного виявлення проблем у навчанні, що покращить ефективність навчання та здобутих знань в сучасному світі.

Розглянуті та використані методи показують, що правильний вибір методик розробки проекту може полегшити процес навчання та зробити його цікавим в реаліях сучасного світу, як для викладачів так і для студентів.

В результаті кваліфікаційної роботи отримано надійну програму, яку можна використовувати як систему моніторингу студентів.

Немає сумніву, що цей проект продовжуватиме розвиватися, і в планах буде розробка готових функцій, додавання нових функцій, надання можливості інтеграції проекту з іншими проектами, покращення зовнішнього вигляду та зручності використання.

Розроблені додатки відповідають усім вимогам предметної області, а створені таблиці бази даних відповідають вимогам нормалізації, що дає можливість гарантувати цілісність і несуперечність інформації.

Для виконання цього завдання в середовищі Visual Studio 2022 за допомогою Windows Forms було вибрано таку мову програмування C#.

Демонстрація виконаної роботи підтверджує працездатність створеного додатку. В цілому цілі роботи досягнуті.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кормен Т., Лейзерсон Ч., Рівест Р. Алгоритми: побудова та аналіз: навч. посіб. М.: МЦМНО, 2001. 960 с.
2. Шеремет В.В., Павлюченко, В. М., Шапіро В. Д. та ін. Управління інвестиціями: навч. посіб. М.: Вища школа, 1998. 401 с.
3. Шапіро В.Д. та ін. Управління проектами: підручник для вузів. СПб.: Дватри, 1996. 354 с.
4. Буч Г., Об'єктно-орієнтований аналіз та Управління проектами: підручник для вузів - СПб., 1996. 215 с.
5. Шапіро В. Д. Управління проектами: тлумачний англо-російський словник - довідник. М.: Вища школа, 2000. 305 с.
6. . Harold Kerzner. Project Management. 6th edition, John Wiley & Sons, Inc., 1999. 6 p.
7. Мазур І. І., Шапіро В. Д. Реструктуризація підприємств та компаній. довідковий посібник. М.: Вища школа, 2000. 256 с.
8. Harvey Maylor. Project Management. 2nd edition, Financial Times / Pitman Publishing, 1999. 5 p.
9. Мовчан І. Н. Деякі аспекти використання сучасних технологій дистанційного навчання у вузі. Збірник наукових праць Sworld. 2013. №. 4. - С. 77-80. URL: https://www.vspu.edu.ua/faculty/imad/files/z/v_20_08.pdf
10. Полат Е. С. Нові педагогічні та інформаційні технології в системі освіти: навчальний посібник для студентів вузів. М.: Академія, 2008. 269 с.
11. Околелов О. П. Дидактика дистанційної освіти: навч. посіб. М.: Директ-Медіа, 2013. 98 с.
12. Алексеєв Г.В., Бриденко І.І., Верболоз Є.І., Дмитриченко М.І. Основи розробки електронних навчальних видань СПб.: Лань, 2016. 144 с.

13. Стеценко, І. В. Моделювання систем: навч. посіб. М-во освіти та науки України: Черкаси: ЧДТУ, 2010. 399 с.
14. Доманов А.Т. Стандарт підприємства СТП. навч. посіб. Мінськ: БДУІР, 2017. 169 с.
15. Навроцький, А.А. Основи алгоритмізації та програмування серед Visual C#: навч.-метод. посібник Мінськ: БДУІР, 2014. 160 с.
16. Сазерленд Джефф. Scrum. Революційний метод управління проектами. М: Манн, Иванов и Фербер, 2015. 288с.
17. Тарасюк Г.М. Управління проектами : навчальний посібник для студентів вищих навчальних закладів. Київ : Каравела, 2006. 320с.
18. Документація з Visual Studio. URL: <https://docs.microsoft.com/ru-ru/visualstudio/ide/?view=vs-2017> 2017.
19. Todd Hoff C# Coding Standard. URL:: http://www.possibility.com/Cpp/c++_coding_standards.pdf
20. Google C# Style Guide. URL:: <https://google.github.io/styleguide/cppguide.html>
21. Введення в мови програмування C та C++ URL: <http://www.intuit.ru/studies/courses/1039/231/info>
22. Верифікація програмного забезпечення: URL: <http://www.intuit.ru/studies/courses/1040/209/info>
23. Інструменти, алгоритми та структури даних URL: <http://www.intuit.ru/studies/courses/683/539/info>
24. Мюссер, Д. С Дердж, А. Сейни # та STL: довідник. М.: Дім "Вільямс", 2010. 116 с.
25. Сатгер, Г. Стандарти програмування на C #: серія "C # In-Depth": навч. посіб. М.: Дім "Вільямс", 2008. 213 с.
26. Седжвік, Р. Алгоритми програмування: навч. посіб., М.: Дім "Вільямс", 2010. 85 с.

27. Шілдт, Г. С++: методики програмування Шілдта: навч. посіб., М.: Дім "Вільямс", 2008. 117 с.
28. Девіс, С. Програмування для "чайників". 6-те вид. навч. посіб., М: Дім "Вільямс", 2010. 409 с.

ДОДАТКИ

Додаток А(DB.cs)

```
using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace library
{
    class DB
    {
        MySqlConnection connection = new
        MySqlConnection("server=localhost; port = 3306;
        username=root; password=root; database=quiz");

        public void openConnection()
        {
            if (connection.State == System.Data.ConnectionState.Closed)
                connection.Open();
        }

        public void closeConnection()
        {
            if (connection.State == System.Data.ConnectionState.Open)
                connection.Close();
        }

        public MySqlConnection getConnection()
        {
            return connection;
        }
    }
}
```

Додаток Б(Form1.cs)

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

namespace QUIZ
{
    public partial class Form1 : Form
    {
        int count = 0;
        string login;
        string password;
        string t1 = "0";
        string t2 = "0";
        string t3 = "0";
        string t4 = "0";
        string t5 = "0";
        string t6 = "0";
        string t7 = "0";
        string t8 = "0";
        string t9 = "0";
        string t10 = "0";

        public Form1()
        {
            InitializeComponent();
            if (comboBox1.SelectedIndex == 0)
            {
                label1.Text = "1/10. Обрати правильне визначення дефініції  
«Програмні системи».";
            }
        }
    }
}
```

```

        radioButton1.Text = "це ідентична форма фіксації програмних
понять яка дозволяє визначити різницю понять," +
Environment.NewLine + "що задаються сукупністю маси
масуджень.";
        radioButton2.Text = "це специфічна форма фіксації рівня
абстракції програмних понять. Вона дозволяє визначити
сукупність понять," + Environment.NewLine + "яка характеризує
рівень, та їх властивості, що задаються сукупністю суджень.";
        radioButton3.Text = "це нейтральна форма фіксації рівня
абстракції програмних понять.";
        label2.Text = "";
    }
    if (comboBox1.SelectedIndex == 1)
    {
        label1.Text = "1/10. Choose the correct
definition of the definition «Software systems».";
        radioButton1.Text = "this is an identical
form of fixation of program concepts that allows to determine
the difference of concepts," + Environment.NewLine + "which are
given by the set of mass masses.";
        radioButton2.Text = "this is a specific form
of fixing the level of abstraction of program concepts. It
allows you to define a set of concepts," + Environment.NewLine
+ "which characterizes the level and their properties given by
a set of judgments.";
        radioButton3.Text = "it is a neutral form of
fixing the level of abstraction of program concepts.";
        label2.Text = "";
    }
}

private void button4_Click(object sender, EventArgs
e)
{
    tabControll1.Visible = true;
    login = textBox1.Text;
    password = textBox2.Text;
    count = 0;
    if (comboBox1.SelectedIndex == 0)
    {
        label1.Text = "1/10. Обрати правильне визначення дефініції
«Програмні системи».";
        radioButton1.Text = "це ідентична форма фіксації програмних
понять яка дозволяє визначити різницю понять," +
Environment.NewLine + "що задаються сукупністю маси
масуджень.";
        radioButton2.Text = "це специфічна форма фіксації рівня
абстракції програмних понять. Вона дозволяє визначити
сукупність понять," + Environment.NewLine + "яка характеризує
рівень, та їх властивості, що задаються сукупністю суджень.";
        radioButton3.Text = "це нейтральна форма фіксації рівня
абстракції програмних понять.";
    }
}

```

```

label2.Text = "";
    }
    if (comboBox1.SelectedIndex == 1)
    {
        label1.Text = "1/10. Choose the correct
definition of the definition «Software systems».";
        radioButton1.Text = "this is an identical
form of fixation of program concepts that allows to determine
the difference of concepts," + Environment.NewLine + "which are
given by the set of mass masses.";
        radioButton2.Text = "this is a specific form
of fixing the level of abstraction of program concepts. It
allows you to define a set of concepts," + Environment.NewLine
+ "which characterizes the level and their properties given by
a set of judgments.";
        radioButton3.Text = "it is a neutral form of
fixing the level of abstraction of program concepts.";
        label2.Text = "";
    }

    if (textBox1.Text == "12345" && textBox2.Text ==
"12345")
    {
        groupBox2.Visible = true;
        groupBox3.Visible = true;
        groupBox4.Visible = true;
    }
    else
    {
        groupBox2.Visible = false;
        groupBox3.Visible = false;
        groupBox4.Visible = true;
    }
    DB db = new DB();

    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new
MySqlDataAdapter();

    MySqlCommand command = new MySqlCommand("SELECT
* FROM `quiz`", db.getConnection());

    adapter.SelectCommand = command;
    adapter.Fill(table);

    dataGridView1.Rows.Clear();

    dataGridView1.RowCount = table.Rows.Count;
    for (int i = 0; i <table.Rows.Count; ++i)
    {

```

```

        Image image = Image.FromFile("../..\\img\\"
+ table.Rows[i].ItemArray[1].ToString());

        dataGridView1.Rows[i].Cells[0].Value =
table.Rows[i].ItemArray[0].ToString();
        dataGridView1.Rows[i].Cells[1].Value =
image;
        dataGridView1.Rows[i].Cells[2].Value =
table.Rows[i].ItemArray[2].ToString();
        dataGridView1.Rows[i].Cells[3].Value =
table.Rows[i].ItemArray[3].ToString();
        dataGridView1.Rows[i].Cells[4].Value =
table.Rows[i].ItemArray[4].ToString();
        dataGridView1.Rows[i].Cells[5].Value =
table.Rows[i].ItemArray[5].ToString();
        dataGridView1.Rows[i].Cells[6].Value =
table.Rows[i].ItemArray[6].ToString();
        dataGridView1.Rows[i].Cells[7].Value =
table.Rows[i].ItemArray[7].ToString();
        dataGridView1.Rows[i].Cells[8].Value =
table.Rows[i].ItemArray[8].ToString();
        dataGridView1.Rows[i].Cells[9].Value =
table.Rows[i].ItemArray[9].ToString();
        dataGridView1.Rows[i].Cells[10].Value =
table.Rows[i].ItemArray[10].ToString();
        dataGridView1.Rows[i].Cells[11].Value =
table.Rows[i].ItemArray[11].ToString();
        dataGridView1.Rows[i].Cells[12].Value =
table.Rows[i].ItemArray[12].ToString();
        dataGridView1.Rows[i].Cells[13].Value =
table.Rows[i].ItemArray[13].ToString();
    }

}

private void button3_Click_1(object sender,
EventArgs e)
{
    if (comboBox1.SelectedIndex == 0)
        label2.Text = "HeBipHO";
    if (comboBox1.SelectedIndex == 1)
        label2.Text = "Wrong";
    label2.ForeColor = Color.Red;

    if (radioButton3.Checked && count == 0)
    {
        if (comboBox1.SelectedIndex == 0)
            label2.Text = "BipHO";
        if (comboBox1.SelectedIndex == 1)
            label2.Text = "Right";
        label2.ForeColor = Color.Green;
    }
}

```

```
        t1 = "1";
    }
    if (radioButton3.Checked && count == 1)
    {
        if (comboBox1.SelectedIndex == 0)
            label2.Text = "BipHO";
        if (comboBox1.SelectedIndex == 1)
            label2.Text = "Right";
        label2.ForeColor = Color.Green;
        t2 = "1";
    }
    if (radioButton1.Checked && count == 2)
    {
        if (comboBox1.SelectedIndex == 0)
            label2.Text = "BipHO";
        if (comboBox1.SelectedIndex == 1)
            label2.Text = "Right";
        label2.ForeColor = Color.Green;
        t3 = "1";
    }
    if (radioButton2.Checked && count == 3)
    {
        if (comboBox1.SelectedIndex == 0)
            label2.Text = "BipHO";
        if (comboBox1.SelectedIndex == 1)
            label2.Text = "Right";
        label2.ForeColor = Color.Green;
        t4 = "1";
    }
    if (radioButton1.Checked && count == 4)
    {
        if (comboBox1.SelectedIndex == 0)
            label2.Text = "BipHO";
        if (comboBox1.SelectedIndex == 1)
            label2.Text = "Right";
        label2.ForeColor = Color.Green;
        t5 = "1";
    }
    if (radioButton3.Checked && count == 5)
    {
        if (comboBox1.SelectedIndex == 0)
            label2.Text = "BipHO";
        if (comboBox1.SelectedIndex == 1)
            label2.Text = "Right";
        label2.ForeColor = Color.Green;
        t6 = "1";
    }
    if (radioButton1.Checked && count == 6)
    {
        if (comboBox1.SelectedIndex == 0)
            label2.Text = "BipHO";
        if (comboBox1.SelectedIndex == 1)
```

```

        label2.Text = "Right";
        label2.ForeColor = Color.Green;
        t7 = "1";
    }
    if (radioButton3.Checked && count == 7)
    {
        if (comboBox1.SelectedIndex == 0)
            label2.Text = "Вірно";
        if (comboBox1.SelectedIndex == 1)
            label2.Text = "Right";
        label2.ForeColor = Color.Green;
        t8 = "1";
    }
    if (radioButton1.Checked && count == 8)
    {
        if (comboBox1.SelectedIndex == 0)
            label2.Text = "Вірно";
        if (comboBox1.SelectedIndex == 1)
            label2.Text = "Right";
        label2.ForeColor = Color.Green;
        t9 = "1";
    }
    if (radioButton2.Checked && count == 9)
    {
        if (comboBox1.SelectedIndex == 0)
            label2.Text = "Вірно";
        if (comboBox1.SelectedIndex == 1)
            label2.Text = "Right";
        label2.ForeColor = Color.Green;
        t10 = "1";
    }
}

private void button1_Click_1(object sender,
EventArgs e)
{
    count++;
    if (count == 1)
    {
        if (comboBox1.SelectedIndex == 0)
        {
            label1.Text = "2/10. Функціонально номінативна система:";
            radioButton1.Text = "алгебра даних:  $V$  - носій цієї алгебри,
які задаються іменами з  $K$  за допомогою відображення  $des$ ";
            radioButton2.Text = "алгебра функцій задаються іменами з  $Z$ 
за допомогою відображення  $bek$ ";
            radioButton3.Text = "алгебра даних:  $D$  - носій цієї алгебри,
а операціями алгебри є функції," + Environment.NewLine + "які
задаються іменами з  $Z$  за допомогою відображення  $den$ ";
            label2.Text = "";
            radioButton1.Checked = true;
        }
    }
}

```

```

        if (comboBox1.SelectedIndex == 1)
        {
            label1.Text = "2/10. Functionally
nominative system:";
            radioButton1.Text = "data algebra: V is
the carrier of this algebra, which are specified by names from
K using the dec mapping";
            radioButton2.Text = "the algebra of
functions are given names from Z using the bek mapping";
            radioButton3.Text = "data algebra: D is
the carrier of this algebra, and algebra operations are
functions," + Environment.NewLine + "which are specified by
names from Z using the den mapping";
            label2.Text = "";
            radioButton1.Checked = true;
        }
    }
    if (count == 2)
    {
        if (comboBox1.SelectedIndex == 0)
        {
            label1.Text = "3/10. Продовжіть речення. Наявність
композицій дозволяє:";
            radioButton1.Text = "ієрархічну побудову програм із деяких
базових функцій багатократним застосуванням композицій.";
            radioButton2.Text = "екстраполювати програми із деяких
базових функцій багатократним застосуванням композицій.";
            radioButton3.Text = "будувати програми із деяких базових
функцій багатократним застосуванням композицій.";
            label2.Text = "";
            radioButton1.Checked = true;
        }
        if (comboBox1.SelectedIndex == 1)
        {
            label1.Text = "3/10. Continue the
sentence. The presence of compositions allows:";
            radioButton1.Text = "hierarchical
construction of programs from some basic functions by repeated
application of compositions.";
            radioButton2.Text = "to extrapolate
programs from some basic functions by repeatedly applying
compositions.";
            radioButton3.Text = "build programs from
some basic functions by repeatedly applying compositions.";
            label2.Text = "";
            radioButton1.Checked = true;
        }
    }
}
if (count == 3)
{
    if (comboBox1.SelectedIndex == 0)
    {

```

```

    label1.Text = "4/10. Продовжіть речення. Наявність
композицій дозволяє:";
    radioButton1.Text = "ієрархічну побудову програм із деяких
базових функцій багатократним застосуванням композицій.";
    radioButton2.Text = "екстраполювати програми із деяких
базових функцій багатократним застосуванням композицій.";
    radioButton3.Text = "будувати програми із деяких базових
функцій багатократним застосуванням композицій.";
    label2.Text = "";

        radioButton1.Checked = true;
    }
    if (comboBox1.SelectedIndex == 1)
    {
        label1.Text = "4/10. Continue the
sentence. The presence of compositions allows:";
        radioButton1.Text = "hierarchical
construction of programs from some basic functions by repeated
application of compositions.";
        radioButton2.Text = "to extrapolate
programs from some basic functions by repeatedly applying
compositions.";

        radioButton3.Text = "build programs from
some basic functions by repeatedly applying compositions.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
}
if (count == 4)
{
    if (comboBox1.SelectedIndex == 0)
    {
        label1.Text = "5/10. Продовжіть речення. Наявність
композицій дозволяє:";
        radioButton1.Text = "ієрархічну побудову програм із деяких
базових функцій багатократним застосуванням композицій.";
        radioButton2.Text = "екстраполювати програми із деяких
базових функцій багатократним застосуванням композицій.";
        radioButton3.Text = "будувати програми із деяких базових
функцій багатократним застосуванням композицій.";
        label2.Text = "";

        radioButton1.Checked = true;
    }
    if (comboBox1.SelectedIndex == 1)
    {
        label1.Text = "5/10. Continue the
sentence. The presence of compositions allows:";
        radioButton1.Text = "hierarchical
construction of programs from some basic functions by repeated
application of compositions.";
        radioButton2.Text = "to extrapolate
programs from some basic functions by repeatedly applying
compositions.";

```

```

        radioButton3.Text = "build programs from
some basic functions by repeatedly applying compositions.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
}
if (count == 5)
{
    if (comboBox1.SelectedIndex == 0)
    {
        label1.Text = "6/10. Продовжіть речення. Наявність
композицій дозволяє:";
        radioButton1.Text = "ієрархічну побудову програм із деяких
базових функцій багатократним застосуванням композицій.";
        radioButton2.Text = "екстраполювати програми із деяких
базових функцій багатократним застосуванням композицій.";
        radioButton3.Text = "будувати програми із деяких базових
функцій багатократним застосуванням композицій.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
    if (comboBox1.SelectedIndex == 1)
    {
        label1.Text = "6/10. Continue the
sentence. The presence of compositions allows:";
        radioButton1.Text = "hierarchical
construction of programs from some basic functions by repeated
application of compositions.";
        radioButton2.Text = "to extrapolate
programs from some basic functions by repeatedly applying
compositions.";
        radioButton3.Text = "build programs from
some basic functions by repeatedly applying compositions.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
}
if (count == 6)
{
    if (comboBox1.SelectedIndex == 0)
    {
        label1.Text = "7/10. Продовжіть речення. Наявність
композицій дозволяє:";
        radioButton1.Text = "ієрархічну побудову програм із деяких
базових функцій багатократним застосуванням композицій.";
        radioButton2.Text = "екстраполювати програми із деяких
базових функцій багатократним застосуванням композицій.";
        radioButton3.Text = "будувати програми із деяких базових
функцій багатократним застосуванням композицій.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
}

```

```

        if (comboBox1.SelectedIndex == 1)
        {
            label1.Text = "7/10. Continue the
sentence. The presence of compositions allows:";
            radioButton1.Text = "hierarchical
construction of programs from some basic functions by repeated
application of compositions.";
            radioButton2.Text = "to extrapolate
programs from some basic functions by repeatedly applying
compositions.";
            radioButton3.Text = "build programs from
some basic functions by repeatedly applying compositions.";
            label2.Text = "";
            radioButton1.Checked = true;
        }
    }
    if (count == 7)
    {
        if (comboBox1.SelectedIndex == 0)
        {
            label1.Text = "8/10. Продовжіть речення. Наявність
композицій дозволяє:";
            radioButton1.Text = "ієрархічну побудову програм із деяких
базових функцій багатократним застосуванням композицій.";
            radioButton2.Text = "екстраполювати програми із деяких
базових функцій багатократним застосуванням композицій.";
            radioButton3.Text = "будувати програми із деяких базових
функцій багатократним застосуванням композицій.";
            label2.Text = "";
            radioButton1.Checked = true;
        }
        if (comboBox1.SelectedIndex == 1)
        {
            label1.Text = "8/10. Continue the
sentence. The presence of compositions allows:";
            radioButton1.Text = "hierarchical
construction of programs from some basic functions by repeated
application of compositions.";
            radioButton2.Text = "to extrapolate
programs from some basic functions by repeatedly applying
compositions.";
            radioButton3.Text = "build programs from
some basic functions by repeatedly applying compositions.";
            label2.Text = "";
            radioButton1.Checked = true;
        }
    }
}
if (count == 8)
{
    if (comboBox1.SelectedIndex == 0)
    {

```

```

    label1.Text = "9/10. Продовжіть речення. Наявність
композицій дозволяє:";
    radioButton1.Text = "ієрархічну побудову програм із деяких
базових функцій багатократним застосуванням композицій.";
    radioButton2.Text = "екстраполювати програми із деяких
базових функцій багатократним застосуванням композицій.";
    radioButton3.Text = "будувати програми із деяких базових
функцій багатократним застосуванням композицій.";
    label2.Text = "";

        radioButton1.Checked = true;
    }
    if (comboBox1.SelectedIndex == 1)
    {
        label1.Text = "9/10. Continue the
sentence. The presence of compositions allows:";
        radioButton1.Text = "hierarchical
construction of programs from some basic functions by repeated
application of compositions.";
        radioButton2.Text = "to extrapolate
programs from some basic functions by repeatedly applying
compositions.";

        radioButton3.Text = "build programs from
some basic functions by repeatedly applying compositions.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
}
if (count == 9)
{
    if (comboBox1.SelectedIndex == 0)
    {
        label1.Text = "10/10. Продовжіть речення. Наявність
композицій дозволяє:";
        radioButton1.Text = "ієрархічну побудову програм із деяких
базових функцій багатократним застосуванням композицій.";
        radioButton2.Text = "екстраполювати програми із деяких
базових функцій багатократним застосуванням композицій.";
        radioButton3.Text = "будувати програми із деяких базових
функцій багатократним застосуванням композицій.";
        label2.Text = "";

        radioButton1.Checked = true;
    }
    if (comboBox1.SelectedIndex == 1)
    {
        label1.Text = "10/10. Continue the
sentence. The presence of compositions allows:";
        radioButton1.Text = "hierarchical
construction of programs from some basic functions by repeated
application of compositions.";
        radioButton2.Text = "to extrapolate
programs from some basic functions by repeatedly applying
compositions.";

```

```

        radioButton3.Text = "build programs from
some basic functions by repeatedly applying compositions.";
        label2.Text = "";
        radioButton1.Checked = true;
    }
}
}

private void button2_Click_1(object sender,
EventArgs e)
{
    Close();
}

private void button5_Click(object sender, EventArgs
e)
{
    DB db = new DB();

    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new
MySqlDataAdapter();

    MySqlCommand command = new MySqlCommand("SELECT
* FROM `quiz`", db.getConnection());

    adapter.SelectCommand = command;
    adapter.Fill(table);

    int id = table.Rows.Count + 1;
    string img = textBox5.Text;
    string sername = textBox6.Text;
    string name = textBox7.Text;
    string login = textBox3.Text;
    string pass = textBox4.Text;

    DB db2 = new DB();

    DataTable table2 = new DataTable();

    MySqlDataAdapter adapter2 = new
MySqlDataAdapter();

    MySqlCommand command2 = new MySqlCommand("INSERT
INTO `quiz` (`id`, `fotourl`, `sername`, `name`, `test1`,
`test2`, `test3`, `test4`, `test5`, `test6`, `test7`, `test8`,
`test9`, `test10`, `login`, `pass`) VALUES (@id, @img,
@sername, @name, @test1, @test2, @test3, @test4, @test5,
@test6, @test7, @test38, @test9, @test10, @login, @pass)",
db.getConnection());

```

```

        command2.Parameters.Add("@id", MySqlDbType.VarChar).Value =
id;
        command2.Parameters.Add("@img", MySqlDbType.VarChar).Value =
img;
        command2.Parameters.Add("@sername",
MySqlDbType.VarChar).Value = sername;
        command2.Parameters.Add("@name", MySqlDbType.VarChar).Value
= name;
        command2.Parameters.Add("@test1", MySqlDbType.VarChar).Value
= t1;
        command2.Parameters.Add("@test2", MySqlDbType.VarChar).Value
= t2;
        command2.Parameters.Add("@test3", MySqlDbType.VarChar).Value
= t3;
        command2.Parameters.Add("@test4", MySqlDbType.VarChar).Value
= t4;
        command2.Parameters.Add("@test5", MySqlDbType.VarChar).Value
= t5;
        command2.Parameters.Add("@test6", MySqlDbType.VarChar).Value
= t6;
        command2.Parameters.Add("@test7", MySqlDbType.VarChar).Value
= t7;
        command2.Parameters.Add("@test8", MySqlDbType.VarChar).Value
= t8;
        command2.Parameters.Add("@test9", MySqlDbType.VarChar).Value
= t9;
        command2.Parameters.Add("@test10",
MySqlDbType.VarChar).Value = t10;

        command2.Parameters.Add("@login", MySqlDbType.VarChar).Value
= login;
        command2.Parameters.Add("@pass", MySqlDbType.VarChar).Value
= pass;

        adapter2.SelectCommand = command2;
        adapter2.Fill(table2);

        DB db1 = new DB();

        DataTable table1 = new DataTable();

        MySqlDataAdapter adapter1 = new
MySqlDataAdapter();

        MySqlCommand command1 = new MySqlCommand("SELECT
* FROM `quiz`", db.getConnection());

        adapter1.SelectCommand = command1;
        adapter1.Fill(table1);

        dataGridView1.Rows.Clear();

```

```

        dataGridView1.RowCount = table1.Rows.Count;
        for (int i = 0; i <table1.Rows.Count; ++i)
        {
            Image image = Image.FromFile("../..\\img\\"
+ table1.Rows[i].ItemArray[1].ToString());

            dataGridView1.Rows[i].Cells[0].Value =
table1.Rows[i].ItemArray[0].ToString();
            dataGridView1.Rows[i].Cells[1].Value =
image;
            dataGridView1.Rows[i].Cells[2].Value =
table1.Rows[i].ItemArray[2].ToString();
            dataGridView1.Rows[i].Cells[3].Value =
table1.Rows[i].ItemArray[3].ToString();
            dataGridView1.Rows[i].Cells[4].Value =
table1.Rows[i].ItemArray[4].ToString();
            dataGridView1.Rows[i].Cells[5].Value =
table1.Rows[i].ItemArray[5].ToString();
            dataGridView1.Rows[i].Cells[6].Value =
table1.Rows[i].ItemArray[6].ToString();
            dataGridView1.Rows[i].Cells[7].Value =
table1.Rows[i].ItemArray[7].ToString();
            dataGridView1.Rows[i].Cells[8].Value =
table1.Rows[i].ItemArray[8].ToString();
            dataGridView1.Rows[i].Cells[9].Value =
table1.Rows[i].ItemArray[9].ToString();
            dataGridView1.Rows[i].Cells[10].Value =
table1.Rows[i].ItemArray[10].ToString();
            dataGridView1.Rows[i].Cells[11].Value =
table1.Rows[i].ItemArray[11].ToString();
            dataGridView1.Rows[i].Cells[12].Value =
table1.Rows[i].ItemArray[12].ToString();
            dataGridView1.Rows[i].Cells[13].Value =
table1.Rows[i].ItemArray[13].ToString();
        }
    }

    private void button6_Click(object sender, EventArgs
e)
    {
        string id = textBox8.Text;
        DB db = new DB();

        DataTable table = new DataTable();

        MySqlDataAdapter adapter = new
MySqlDataAdapter();

        MySqlCommand command = new MySqlCommand("DELETE
FROM `quiz` WHERE `quiz`.`id` = @id", db.getConnection());
        command.Parameters.Add("@id", MySqlDbType.VarChar).Value =
id;

```

```

adapter.SelectCommand = command;
adapter.Fill(table);

        DB db1 = new DB();

        DataTable table1 = new DataTable();

        MySqlConnection adapter1 = new
MySqlConnection();

        MySqlCommand command1 = new MySqlCommand("SELECT
* FROM `quiz`", db.getConnection());

        adapter1.SelectCommand = command1;
        adapter1.Fill(table1);

dataGridView1.Rows.Clear();

        dataGridView1.RowCount = table1.Rows.Count;
        for (int i = 0; i <table1.Rows.Count; ++i)
        {
            Image image = Image.FromFile("../..\\img\\"
+ table1.Rows[i].ItemArray[1].ToString());

            dataGridView1.Rows[i].Cells[0].Value =
table1.Rows[i].ItemArray[0].ToString();
            dataGridView1.Rows[i].Cells[1].Value =
image;
            dataGridView1.Rows[i].Cells[2].Value =
table1.Rows[i].ItemArray[2].ToString();
            dataGridView1.Rows[i].Cells[3].Value =
table1.Rows[i].ItemArray[3].ToString();
            dataGridView1.Rows[i].Cells[4].Value =
table1.Rows[i].ItemArray[4].ToString();
            dataGridView1.Rows[i].Cells[5].Value =
table1.Rows[i].ItemArray[5].ToString();
            dataGridView1.Rows[i].Cells[6].Value =
table1.Rows[i].ItemArray[6].ToString();
            dataGridView1.Rows[i].Cells[7].Value =
table1.Rows[i].ItemArray[7].ToString();
            dataGridView1.Rows[i].Cells[8].Value =
table1.Rows[i].ItemArray[8].ToString();
            dataGridView1.Rows[i].Cells[9].Value =
table1.Rows[i].ItemArray[9].ToString();
            dataGridView1.Rows[i].Cells[10].Value =
table1.Rows[i].ItemArray[10].ToString();
            dataGridView1.Rows[i].Cells[11].Value =
table1.Rows[i].ItemArray[11].ToString();
            dataGridView1.Rows[i].Cells[12].Value =
table1.Rows[i].ItemArray[12].ToString();

```

```

        dataGridView1.Rows[i].Cells[13].Value =
table1.Rows[i].ItemArray[13].ToString();
    }
}

private void button7_Click(object sender, EventArgs
e)
{
    string search = textBox9.Text;

    DB db = new DB();

    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new
MySqlDataAdapter();

    MySqlCommand command = new MySqlCommand("SELECT
* FROM `quiz`", db.getConnection());

    adapter.SelectCommand = command;
    adapter.Fill(table);

    dataGridView1.Rows.Clear();

    int count = 0;
    for (int i = 0; i <table.Rows.Count; ++i)
        if (search ==
table.Rows[i].ItemArray[2].ToString())
            count++;

    dataGridView1.RowCount = count == 0 ?1 : count;
    count = 0;
    for (int i = 0; i <table.Rows.Count; ++i)
    {
        if (search ==
table.Rows[i].ItemArray[2].ToString())
        {
            Image image =
Image.FromFile("../..\\img\\"
+
table.Rows[i].ItemArray[1].ToString());

            dataGridView1.Rows[count].Cells[0].Value
= table.Rows[i].ItemArray[0].ToString();
            dataGridView1.Rows[count].Cells[1].Value
= image;

            dataGridView1.Rows[count].Cells[2].Value
= table.Rows[i].ItemArray[2].ToString();
            dataGridView1.Rows[count].Cells[3].Value
= table.Rows[i].ItemArray[3].ToString();
            dataGridView1.Rows[count].Cells[4].Value
= table.Rows[i].ItemArray[4].ToString();

```

```

        dataGridView1.Rows[count].Cells[5].Value
= table.Rows[i].ItemArray[5].ToString();
        dataGridView1.Rows[count].Cells[6].Value
= table.Rows[i].ItemArray[6].ToString();
        dataGridView1.Rows[count].Cells[7].Value
= table.Rows[i].ItemArray[7].ToString();
        dataGridView1.Rows[count].Cells[8].Value
= table.Rows[i].ItemArray[8].ToString();
        dataGridView1.Rows[count].Cells[9].Value
= table.Rows[i].ItemArray[9].ToString();

dataGridView1.Rows[count].Cells[10].Value           =
table.Rows[i].ItemArray[10].ToString();

dataGridView1.Rows[count].Cells[11].Value           =
table.Rows[i].ItemArray[11].ToString();

dataGridView1.Rows[count].Cells[12].Value           =
table.Rows[i].ItemArray[12].ToString();

dataGridView1.Rows[count].Cells[13].Value           =
table.Rows[i].ItemArray[13].ToString();
        count++;
    }
}

private void button8_Click(object sender, EventArgs
e)
{
    DB db = new DB();

    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new
MySqlDataAdapter();

    MySqlCommand command = new MySqlCommand("SELECT
* FROM `quiz`", db.getConnection());

    adapter.SelectCommand = command;
    adapter.Fill(table);

    DB db2 = new DB();

    DataTable table2 = new DataTable();

    MySqlDataAdapter adapter2 = new
MySqlDataAdapter();

```

```

        MySqlCommand command2 = new MySqlCommand("UPDATE
`quiz` SET `test1` = @test1, `test2` = @test2, `test3` =
@test3, `test4` = @test4, `test5` = @test5, `test6` = @test6,
`test7` = @test7, `test8` = @test8, `test9` = @test9, `test10`
= @test10 WHERE `quiz`.`login` = @login AND `quiz`.`pass` =
@pass", db.getConnection());
        command2.Parameters.Add("@test1", MySqlDbType.VarChar).Value
= t1;
        command2.Parameters.Add("@test2", MySqlDbType.VarChar).Value
= t2;
        command2.Parameters.Add("@test3", MySqlDbType.VarChar).Value
= t3;
        command2.Parameters.Add("@test4", MySqlDbType.VarChar).Value
= t4;
        command2.Parameters.Add("@test5", MySqlDbType.VarChar).Value
= t5;
        command2.Parameters.Add("@test6", MySqlDbType.VarChar).Value
= t6;
        command2.Parameters.Add("@test7", MySqlDbType.VarChar).Value
= t7;
        command2.Parameters.Add("@test8", MySqlDbType.VarChar).Value
= t8;
        command2.Parameters.Add("@test9", MySqlDbType.VarChar).Value
= t9;
        command2.Parameters.Add("@test10",
MySqlDbType.VarChar).Value = t10;
        command2.Parameters.Add("@login", MySqlDbType.VarChar).Value
= login;
        command2.Parameters.Add("@pass", MySqlDbType.VarChar).Value
= password;

        adapter2.SelectCommand = command2;
        adapter2.Fill(table2);

        DB db1 = new DB();

        DataTable table1 = new DataTable();

        MySqlDataAdapter adapter1 = new
MySqlDataAdapter();

        MySqlCommand command1 = new MySqlCommand("SELECT
* FROM `quiz`", db.getConnection());

        adapter1.SelectCommand = command1;
        adapter1.Fill(table1);

        dataGridView1.Rows.Clear();

        dataGridView1.RowCount = table1.Rows.Count;
        for (int i = 0; i < table1.Rows.Count; ++i)
        {

```

```

        Image image = Image.FromFile("../..\..\img\\"
+ table1.Rows[i].ItemArray[1].ToString());

        dataGridView1.Rows[i].Cells[0].Value =
table.Rows[i].ItemArray[0].ToString();
        dataGridView1.Rows[i].Cells[1].Value =
image;
        dataGridView1.Rows[i].Cells[2].Value =
table.Rows[i].ItemArray[2].ToString();
        dataGridView1.Rows[i].Cells[3].Value =
table.Rows[i].ItemArray[3].ToString();
        dataGridView1.Rows[i].Cells[4].Value =
table.Rows[i].ItemArray[4].ToString();
        dataGridView1.Rows[i].Cells[5].Value =
table.Rows[i].ItemArray[5].ToString();
        dataGridView1.Rows[i].Cells[6].Value =
table.Rows[i].ItemArray[6].ToString();
        dataGridView1.Rows[i].Cells[7].Value =
table.Rows[i].ItemArray[7].ToString();
        dataGridView1.Rows[i].Cells[8].Value =
table.Rows[i].ItemArray[8].ToString();
        dataGridView1.Rows[i].Cells[9].Value =
table.Rows[i].ItemArray[9].ToString();
        dataGridView1.Rows[i].Cells[10].Value =
table.Rows[i].ItemArray[10].ToString();
        dataGridView1.Rows[i].Cells[11].Value =
table.Rows[i].ItemArray[11].ToString();
        dataGridView1.Rows[i].Cells[12].Value =
table.Rows[i].ItemArray[12].ToString();
        dataGridView1.Rows[i].Cells[13].Value =
table.Rows[i].ItemArray[13].ToString();
    }
}

private void comboBox1_SelectedIndexChanged(object
sender, EventArgs e)
{
    if (comboBox1.SelectedIndex == 0)
    {
        groupBox1.Text = "Вхід";
        label3.Text = "Логін";
        label4.Text = "Пароль";
        button4.Text = "Вхід";
        groupBox4.Text = "Пошук";
        label11.Text = "Прізвище";
        button7.Text = "Пошук";
        textBox9.Text = "ІВАНОВ";
        button1.Text = "НАСТУПНЕ";
        button3.Text = "ПЕРЕВІРИТИ";
        button2.Text = "ВИЙТИ";
        tabPage1.Text = "Тести";
        tabPage2.Text = "РЕЙТИНГ";
    }
}

```

```

        groupBox2.Text = "Додати користувача";
groupBox3.Text = "Видалити за номером";
label5.Text = "Логін";
        label6.Text = "Пароль";
        label7.Text = "Зображення";
        label8.Text = "Прізвище";
        label9.Text = "Ім'я";
        label10.Text = "Видалити №";
        button8.Text = "Оновити";
        button5.Text = "Додати";
        button6.Text = "Видалити";
    }
    if (comboBox1.SelectedIndex == 1)
    {
        groupBox1.Text = "Enter";
        label3.Text = "Login";
        label4.Text = "Password";
        button4.Text = "Enter";
        groupBox4.Text = "Search";
        label11.Text = "Sername";
        button7.Text = "Search";
        textBox9.Text = "Ivanov";
        button1.Text = "NEXT";
        button3.Text = "CHECK";
        button2.Text = "EXIT";
        tabPage1.Text = "Tests";
        tabPage2.Text = "Rating";
        groupBox2.Text = "Add user";
        groupBox3.Text = "Delete by number";
        label5.Text = "Login";
        label6.Text = "Pass";
        label7.Text = "Image";
        label8.Text = "Sername";
        label9.Text = "Name";
        label10.Text = "Delete №";
        button8.Text = "Refresh";
        button5.Text = "Add";
        button6.Text = "Delete";
    }
}
}
}
}

```