

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: «Розробка інформаційної системи для забезпечення ефективності  
дистанційного навчального процесу з використанням мови C#»

Виконав(ла): студент(ка) 2 курсу, групи СПд-2

спеціальності № 121 Інженерія

програмного забезпечення

(шифр і назва спеціальності)

Шевчук В. В.

(підпис)

(прізвище та ініціали)

Керівник

Бойко І. В.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Гвоздик М.В.

(підпис)

(прізвище та ініціали)

Завідувач кафедри

Петрик М.Р.

(підпис)

(прізвище та ініціали)

Рецензент

Савків В.Б

(підпис)

(прізвище та ініціали)

## АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи містить 68 сторінку, 8 таблиць, 11 рисунків, список використаної літератури з 23 найменувань, 4 додатки.

Через пандемію коронавірусу галузь освіти шукає нові шляхи, щоб освітній процес тривав та був якісним і безпечним. Дистанційне навчання є перспективною сферою розвитку освіти в сучасних умовах.

Сьогодні, майже кожен учень володіє смартфоном та іншими пристроями. Гаджети, підключені до Інтернету, та програмне забезпечення, завантажене на них, дають можливість персоналізованого навчання. Програмне забезпечення може виявити недоліки у навчанні кожного студента та запропонувати викладачу, як навчальний матеріал можна персоналізувати, щоб допомогти кожному студенту. За допомогою програмного забезпечення викладач може бути в курсі успіхів кожного студента в режимі реального часу.

Незалежно від того, де ви знаходитесь, якщо у вас є цифровий пристрій та підключення до Інтернету, перед вами відкривається цілий світ знань.

Тому, було прийняте рішення створити систему дистанційної підтримки освіти. При створенні системи було використано ASP.net MVC і мову програмування C#

Ключові слова: Дистанційне навчання, C#, ASP.net MVC, об'єктно-орієнтована мова програмування.

## **ABSTRACT**

The explanatory note to the qualification work contains 68 pages, 8 tables, 11 figures, the list of the used literature from 23 names, 4 appendices.

Through the coronavirus pandemic, the field of education is looking for new ways for the educational process to continue and be of high quality and safe. Distance learning is a promising area of education in modern conditions.

Today, almost every student owns a smartphone and other devices. Internet-connected devices and the software downloaded on them, enable personalized learning. The software can spot each student's learning shortcomings and suggest to the instructor how the learning material can be personalized to assist each student. The software can keep the teacher abreast of every student's progress in real time.

No matter where you are, if you have a digital device and a connection to the internet, a whole world of knowledge opens up to you.

Therefore, it was decided to create a system of distance education support. ASP.net MVC and C # were used when creating the system

Keywords: Distance education, C #, ASP.net MVC, object-oriented programming language.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

UML – Unified modelling language, уніфікована мова моделювання, уніфікована мова графічного представлення та об'єктного моделювання в області розробки програмного забезпечення парадигми об'єктноорієнтованого програмування.

ІТ – Інтернет технологія

MVC - Model-View-Controller модель , уявлення і контролер.

БД – база даних.

ООП – об'єктно-орієнтоване програмування, парадигма програмування, в якій основою є класи та об'єкти, які між собою взаємодіють.

СУБД – система управління базою даних.

СКБД – система контролю базою даних.

ТНТУ – Тернопільський національний технічний університет імені Івана Пулюя

## ЗМІСТ

<b>ВСТУП.....</b>	<b>6</b>
<b>1. РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ .....</b>	<b>8</b>
1.1. Аналіз вимог до програмної системи .....	8
1.1.1. Аналіз предметної області .....	8
1.1.2 Постановка задачі.....	9
1.1.3 Пошук актантів та варіантів використання .....	9
<b>1.2. Проектування програмної системи .....</b>	<b>10</b>
1.2.1 Вибір процесу розробки.....	10
1.2.2 Побудова схеми бази даних .....	14
1.2.3 Побудова UML-діаграми класів .....	17
<b>1.3. Конструювання програмної системи .....</b>	<b>18</b>
1.3.1 Вибір мови та середовища розробки.....	18
1.3.2 Вибір СУБД та опис її фізичної моделі .....	25
1.3.3 Реалізація основних класів та методів .....	31
<b>2. ВИКОРИСТАННЯ ПРОГРАМНОЇ СИСТЕМИ .....</b>	<b>32</b>
2.1 Розгортання програмної системи та системні вимоги .....	32
2.2 Опис типових схем використання системи .....	33
<b>3. ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ.....</b>	<b>37</b>
<b>4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....</b>	<b>47</b>
<b>ВИСНОВКИ.....</b>	<b>52</b>
<b>ПЕРЕЛІК ПОСИЛАНЬ.....</b>	<b>54</b>
<b>ДОДАТОК А .....</b>	<b>58</b>
<b>ДОДАТОК Б.....</b>	<b>64</b>
<b>ДОДАТОК В.....</b>	<b>67</b>
<b>ДОДАТОК Г .....</b>	<b>68</b>

## ВСТУП

**Актуальність** В березні 2020 року Україна, як і увесь світ зіткнулися з пандемією Covid19. Перед викладачами, учителям постали нові виклики. Як проводити навчання в умовах карантину?

Що краще: дистанційне навчання чи традиційна школа? Це питання дуже актуальне, але через ризики, які стосуються здоров'я і життя кожної людини та її рідних дистанційне навчання вийшло на передній план. Тому в кожному навчальному закладі шукають можливості забезпечення якості дистанційної освіти. “Ми розуміємо, що дистанційна освіта не може замінити справжнього очного навчання. Діти мають соціалізуватися, особливо в початковій школі. Через пандемію коронавірусу весь світ шукає альтернативні шляхи для економічного та соціального життя. І галузь освіти – не виняток, адже для нас важливо, щоб освітній процес тривав та був якісним і безпечним. Дистанційне навчання є перспективною сферою розвитку освіти в умовах її цифровізації. Водночас таке навчання у разі потреби дозволяє забезпечити індивідуальну освітню траєкторію здобувача освіти, а також неперервність освітнього процесу у випадках надзвичайних обставин, які об’єктивно унеможливають відвідування закладів освіти”, – зазначив т.в.о. Міністра освіти і науки Сергій Шкарлет.

На сьогоднішній день є велика кількість систем, які допомагають викладачам працювати з учнями дистанційно. Наприклад, classroom від компанії Google, е-навчання електронне навчання в ТНТУ, online система дистанційної підтримки навчання у школах, ліцєях та гімназіях України від лабораторії інформаційних та комунікаційних технологій, Вінницького міського центру дистанційної освіти школярів та інші. Усі вони допомагають організувати процес дистанційного навчання, але кожен заклад бажає мати своє середовище, в якому можна додавати

нові можливості враховуючи індивідуальні особливості закладу освіти, зручності викладачів та покращення якості дистанційного навчання.

Тому виникла ідея створення власної інформаційної системи для забезпечення ефективності дистанційного навчального процесу.

**Мета роботи.** Вирішення проблем системи освіти. Використання такої системи дозволить створити умови для роботи викладачів в умовах дистанційного навчання. Вони зможуть якісніше використовувати час для того, щоб краще підготувати навчальний матеріал, ретельніше перевіряти роботи студентів, шукати індивідуальний підхід до кожного студента та багато іншого. Перевага такої системи, що є можливість удосконалення її роботи відповідно до потреб викладача. Що буде сприяти постійному покращенню якості освіти.

Об'єктом дослідження є об'єктно-орієнтовані мови програмування, веб-технології та методи обробки даних в мережі Інтернет.

Предметом роботи є методи та засоби розробки інформаційної системи для забезпечення ефективності дистанційного навчального процесу

Завдання роботи є розробка інформаційно системи для забезпечення ефективності дистанційного навчального процесу, а саме розробка веб-сайту з доступним та зручним інтерфейсом. як для студента, так і для викладача; Також для забезпечення захисту інформації буде створена база даних.

# 1. РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

## 1.1. Аналіз вимог до програмної системи

До системи було поставлено перелік вимог:

- можливість використання на різних комп'ютерах (на стаціонарних ПК, планшетах, смартфонах)
- мультиплатформеність (з операційними системами Windows, Android, MacOS, IOS);
- простота інтерфейсу;

Також, програма має виконувати свою пряму функцію як середовище для роботи з студентами в умовах дистанційного навчання.

Розроблена система має бути інтуїтивно зрозумілою, щоб студент та викладач могли розібратися без читання інструкції чи проходження спеціального навчання користуванням.

### 1.1.1. Аналіз предметної області

В навчальному процесі з давніх давен беруть участь мінімум дві особи. Їх можна називати по різному. Перша особа - викладач, викладавець, професор, вихователь, направник, метр, учитель, інструктор, тьютор, педагог. Та друга особа це учень, студент, слухач, школяр та інше.

За звичай в одного викладача не один студент і навпаки. Тому для оптимізації роботи вчителів студентів об'єднують в групи. В кожній групі є наставник, який допомагає студентам в організації їхньої роботи.

Звичайно в навчальному процесі беруть участь і батьки, і адміністрація навчального закладу.

Вчителі мають спеціальність або предмет з якого проводять уроки і оцінюють роботу студентів.

Для дистанційного навчання нічого не змінюється всі учасники навчального процесу продовжують працювати в тому самому стилі, але використовуючи сучасні інформаційні технології.



### 1.1.2 Постановка задачі

Інформаційна система повинна передбачати реєстрацію користувачі з різними правами доступу.

Викладачі та вчителі мають можливість легко та швидко створювати завдання учнів та студентів у електронній формі.

Вчитель/викладач має можливість створити курс а в ньому уроки а також додавати створювати групи та додавати до них студентів.Надалі вони зможуть редагувати список груп

Студенти зможуть переглядати завдання та додавати роботи тільки тих груп в які їх додали викладачі.

Для реалізації таких можливостей потрібно передбачити роботи з БД.

### 1.1.3 Пошук актантів та варіантів використання

Для створення архітектури системи, визначимо акторів (таблиця 1.1)

Актор	Позначення	Дії
Викладач Адмін	Admin	Може додавати матеріали до сайту та редагувати їх
Студент	Student	Може переглядати сайт

Для початку реалізуємо процес тільки для двох акторів що спростить нашу роботу. Викладач зможе працювати в режимі admin. А студенту для роботи потрібно зареєструватися та записатися на курс.

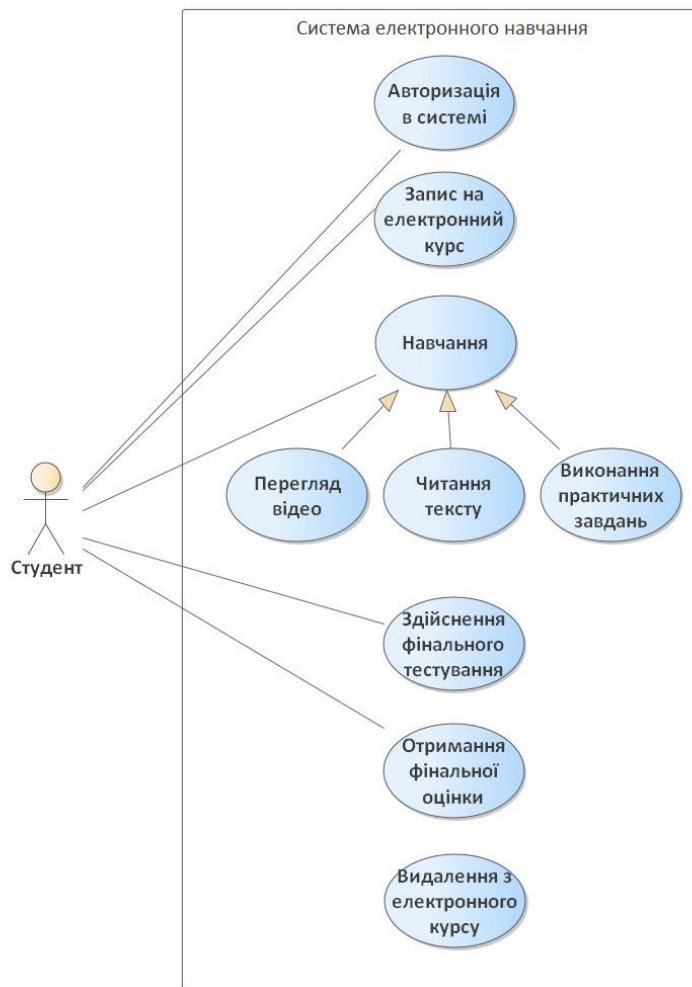


Рисунок 1.1 Діаграма роботи студента

## 1.2. Проектування програмної системи

### 1.2.1 Вибір процесу розробки

Для усунення проблеми розподілеості користувачів, зручними засобами для одночасного доступу до інформації можуть бути Інтернет технології.

Електронне навчання стає звичайною практикою в українській шкільній системі. Застосування Інтернет технологій в освіті набирає обертів, і наслідки цього використання величезні.

Розвиток мобільних технологій та Інтернет технологій дозволяє школам підвищити безпеку, відстежувати ключові ресурси і розширювати доступ до інформації. Вчителі можуть навіть використовувати цю технологію для створення «розумних планів уроків», замість традиційних стоїчних планів минулих років.

Інтернет технології це всі технології які дозволяють нам працювати з інформацією в інтернеті. Інтернет це глобальна мережа яка дозволяє з'єднати різні пристрої по всьому світу. інформація передається за допомогою веб-сайтів, та різних сервісів. За допомогою мережі Інтернет сьогодні люди можуть: отримувати інформацію, купувати або продавати, спілкуватися, навчатися та багато іншого. Вся ця робота організована за певними правилами, використовуючи різні технічні засоби та програм. Раніше основним джерелом інформації були книги. Але з появою глобальної мережі лідером з розповсюдження інформації став інтернет. Основним інструментом для навчання були книги та підручники та на заміну їм пришли Інтернет технології.

В основі усіх інтернет-технологій лежать веб-сайти, які в свою чергу складаються веб-сторінок. Сторінки створюються за допомогою мови розмітки гіпертексту HTML. Вигляд веб-сторінки можна покращити використанням до неї нової таблиці стилів CSS та створити інтерактивного вигляду за допомогою мови Javascript.

Ідея створення комп'ютерної мережі з'явилась одразу після створення перших комп'ютерів. З початку мережу використовували тільки для передачі даних. Після того як було створенно глобальну мережу її в основному і використовували для пересилання повідомлень та файлів. Але в останні десять років Інтернет дозволяє виконувати такі дії, які ми могли бачити тільки в фантастичних фільмах. Інтернет поглинає цілі сфери діяльності суспільства. Важко назвати такий вид діяльності людини в якому не можна використати Інтернет. Та з кожним днем все більше сфер не можуть уявити своє існування без глобальної мережі. Освіта не є виключенням з правила. Можливо пройде ще не багато часу і ми не зможемо уявити звичайну школу як клас в якому сидять учні перед ними дошка і вчитель. А під словом «школа» будемо розуміти веб сервіс який надає освітні послуги.

В наш час вже існує велика кількість інформаційних систем, які сприяють електронному навчанню та дистанційній освіті. Ці системи роблять перші кроки для майбутнього переходу повністю на електронне навчання. Свої системи дистанційного навчання створюють в першу чергу заклади вищої освіти, до цього процесу підключаються і заклади професійної освіти і деякі школи. Але більшість шкіл України можуть користуватися їхніми розробками, або сервісами які створюють різні корпорації України. Звичайно мрія кожного вчителя, щоб була надійна система до якої за його бажанням можна було додавати нові можливості.

Для розв'язання задачі поставленої нам в процесі написання магістерської роботи, а саме створення системи для дистанційного навчання необхідно визначитись із відповідними методами та технологіями, які будуть використовуватись в процесі роботи над проектом.

На сьогодні існує велика кількість методів і технологій створення веб-сайтів та сервісів роботи в Інтернеті. Всі їх можна умовно розділити на дві великі групи

Перша група – це створення сайтів в ручному режимі. Для цього можна використати як спеціалізовані програми так і звичайний тестовий редактор. Для цього розробнику потрібно добре знати мову розмітки гіпертексту HTML, вміти користуватись CSS стилями. Бажанно знати як працювати з Javascript.

HTML (HyperText Markup Language – мова розмітки гіпер тексту) є найосновнішим будівельним матеріалом на просторах мережі Інтернет. Він визначає значення та структуру веб-вмісту. Інші технології, зазвичай використовуються для опису зовнішнього вигляду веб-сторінки (CSS) або функціональності поведінки (JavaScript). Якщо продовжити порівняння, HTML - це цегла та бетон з яких ми будуємо будинок, CSS це спосіб оформлення ,який визначає якого кольору

будуть стіни в нашому будинку, JavaScript дозволить додати інтерактиву, наприклад, змінювати колір будинку в залежності від погоди. Для створення сучасного звичайного статичного веб сайту цих технологій може і вистачити. Але щоб додати більше можливостей для роботи з сайтом потрібно знати і використовувати мови програмування для Веб – ASP, PHP, PERL або інші. Працювати і редагувати файли .php можна в будь якому текстовому редакторі так і спеціалізованих рограмах. Для створення сайту за допомогою ASP.NET вам доведеться додатково інсталбвати на свій ПК програму Visual Studio, яку можна безкоштовно отримати від компанії Microsoft. За допомогою мов веб-програмування пишуться програми веб-додатки. Що краще веб додаток чи веб-сайт? Веб сайт постійно потрібно завантажувати на ПК, а веб додаток має мати стабільний інтернет-з'єднання.

Щоб додавати до сайту функціонал використання HTML, CSS і JavaScript не достатньо, тому потрібно використовувати і мову програмування. Щоб наша система для забезпечення ефективності дистанційного навчального процесу могла отримувати нові можливості потрібно використовувати цей мови програмування.

Друга група – це використання автоматизованих систем розробки веб сайтів. Для роботи з ними не потрібне глибоке знання HTML чи CSS. Але й змінити такий сайт повністю під свої бажання ми не зможемо. Автоматизовані системи, як правило дають змогу одразу публікувати веб-сайт в Інтернеті.

Ще такі автоматизовані системи створення веб-сайтів називають конструкторами. І справді, ми можемо зібрати наш сайт з готових блоків. Також як правило в основу веб-сайту ляже готовий шаблон в якому внести зміни, або неможливо, або дуже проблематично. Також вибір стилів оформлення веб сторінки є дуже обмеженим. Перевагою таких цієї групи методів є відносна простота в створенні сайту і забезпечення стабільної роботи сайту. В таких системах відбувається

поділ на дві частини «дизайн» і «контент». На початку роботи ми обираємо який зовнішній вигляд буде мати наш сайт. А потім працюємо з додаванням вмісту і його редагуванням. Звичайно потім можна змінити і дизайн, але це ні в якому разі не пошкодить контент.

При ручному створення сайту такої можливості поділу на «дизайн» і «контент» не має хоча є шанс такого ефекту досягнути використовуючи мови програмування.

Зважаючи на перелічені особливості методів веб-розробки було прийняте рішення користуватися першою групою методів. Ручне створенн сайту дозволить можливість швидкої адаптації розробки до нових вимог, які переднами буде ставити час. Отже, як інструментальний засіб для розробки сайту було обрано мову програмування C# та мову веб програмування ASP.Net Core MVC. Для успішної роботи з якими потрібно встановити Microsoft Visual Studio.

#### 1.2.2 Побудова схеми бази даних

Для створення інформаційної системи забезпечення ефективності дистанційного навчального процесу відповідно до аналізу предметної області було обранно реляційну базу даних.

Структура системи бази даних описана формальною мовою, відноситься до організації даних для створення і побудови бази даних з розподілом на таблиці називається схемою баз даних.

В схемі баз даних відображаються сутності предметної області які подаються у вигляді таблиць та зв'язків між таблицями. Таблиці мають поля(стовпці) до яких можна вносити записи (рядки). Гарна логічна схема баз даних буде забезпечувати коректну роботу усієї систему та цілісність даних. Признаком доброї логічної схеми бази даних буде її простота.

Для нормалізації бази даних потрібно таблиці з великою кількістю полів, розділяти на менші пов'язані таблиці. Чим краще побудована схема бази даних, тим краще буде з нею працювати.

Для забезпечення цілісності даних в кожній таблиці рекомендується використання первинного ключа, який дозволить створити якісні зв'язки між таблицями. Також обов'язково усі ключі в таблицях повині бути узгодженні.

Наша схема баз даних містить чотири таблиці: Студенти(Student), групи(Group), курси (Curs), уроки (Lesson). Студенти пов'язані з групами, групи – з курсами, курси – з уроками.

Дана схема дуже проста що забезпечить якісну роботу над проектом

Схема бази даних розроблена, зображена на рисунку 1.2.

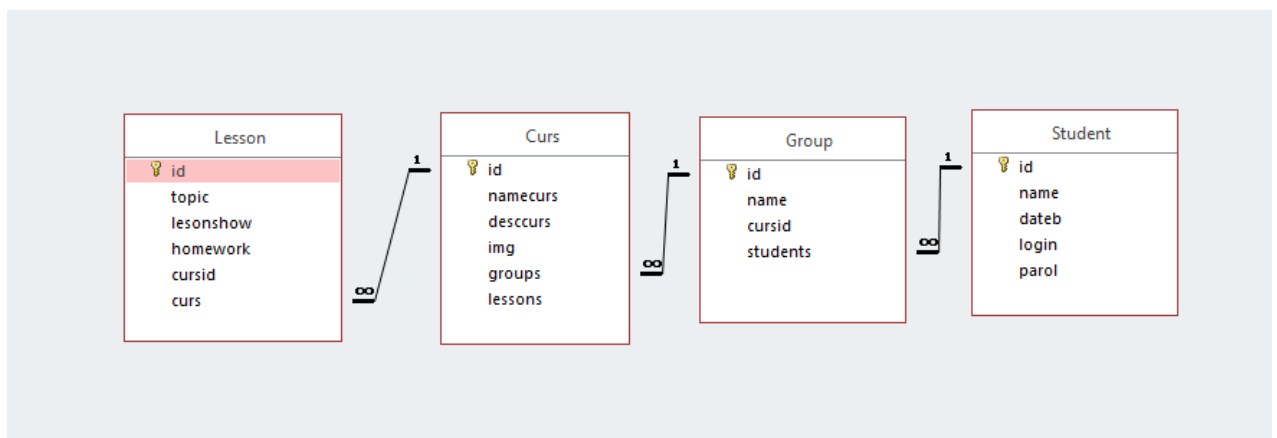


Рисунок 1.2. - База даних

Для початку реалізуємо основні сутності бази даних

Student – таблиця, що містить інформацію про студента. Структуру полів наведено в таблиці 1.2.

Таблиця 1.2 – Структура полів таблиці Student

Назва поля	Тип	Зміст поля
id	int	Ключове поле для ідентифікації запису

name	nvarchar(255)	Прізвище, ім'я, по-батькові студента
dateb	date	Дата народження студента
login	int	Логін
parol	int	Код ключа доступу

Group – таблиця, що містить інформацію про групи, в яких навчаються студенти. Структуру полів наведено в таблиці 1.3.

Таблиця 1.3 – Структура полів таблиці Group

Назва поля	Тип	Зміст поля
id	int	Ключове поле для ідентифікації запису
name	nvarchar(25)	Назва групи
cursid	int	Код групи
students	Student	Список студентів групи студентів

Curs таблиця, що містить інформацію про курси на які може записатися студент. Структуру полів наведено в таблиці 1.4.

Таблиця 1.4 – Структура полів таблиці Curs

Назва поля	Тип	Зміст поля
id	int	Ключове поле для ідентифікації запису
namecurs	nvarchar(255)	Назва курсу
descrcurs	nvarchar(255)	Опис курсу
img	nvarchar(255)	Фото
lessons	Lesson	Список Уроків
groups	Group	Про групи

Lesson – таблиця, що містить інформацію про уроки. Структуру полів наведено в таблиці 1.4.

Таблиця 1.4 – Структура полів таблиці Lesson



Назва поля	Тип	Інформаційний зміст
id	int	Ключове поле для ідентифікації запису
topic	nvarchar(MAX)	Тема заняття
lesonshow	bool	Показувати урок
cursid	int	Код групи
curs	virtual	Інформація про самий курс

### 1.2.3 Побудова UML-діаграми класів

Діаграма класів – створюється для того щоб показати статичні елементи, такі як: типи даних, класи, їх вміст та зв'язки між ними. Крім того діаграма класів в окремих випадках може показувати деякі елементи поведінки, але це можна відобразити за допомогою інших діаграм.

UML - це спосіб візуалізації програмного забезпечення за допомогою колекції діаграм. Діаграми класів раніше використовувалася лише для об'єктно-орієнтованого проектування, але сьогодні вони були розширені для охоплення більш широкого кола програмних інженерних проектів. UML прийнята Object Management Group (OMG) як стандарт для моделювання створення програмного забезпечення.

Сьогоднішні стандарти вимагають тринадцять різних типів діаграм. Їх умовно можна розділити на дві різні групи:

- Структурні діаграми
- Та діаграми взаємодії або поведінки

Діаграма класів відноситься до структурних діаграм. Діаграми класів є основою майже кожного об'єктно-орієнтованого методу. Вони описують статичну структуру системи.

Діаграма класів веб-сервісу інформаційної системи для забезпечення ефективності дистанційного навчального процесу наведена на рисунку 1.3.

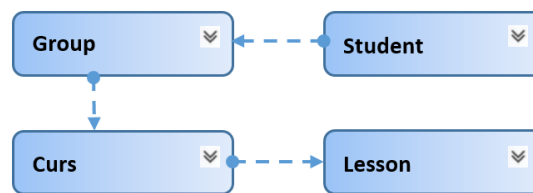


Рисунок 1.3 - Діаграми класів

### 1.3. Конструювання програмної системи

#### 1.3.1 Вибір мови та середовища розробки

У світі існує більше тисячі мов програмування. При створенні інформаційної системи забезпечення ефективності дистанційного навчального процесу було обрано об'єктно-орієнтовану мову програмування C # з використанням технології ASP.NET MVC .

C # створена в двохтисячному році Андерсом Хейлсбергом в корпорації Microsoft як більш швидкий та безпечний варіант C і як заміник Java На початку Java і C # мови були дуже схожі, але в процесі розвитку все більше почали віддалятися одна від одної. C# повністю інтегрований із програмним забезпеченням Microsoft .NET, що підтримує розробку програм для Windows, плагінів браузера та мобільних пристроїв. C # пропонує спільні бази кодів, велику бібліотеку коду та різноманітні типи даних. Завдяки просуванню C# корпорацією Microsoft ця мова стала однією з найпопулярніших мов програмування у світі. Нас зацікавило що C# може використовуватися для розробки веб-додатків і стає все більш популярною і для мобільних розробок [8]. C# дозволяє на основі однієї предметної області реалізувати різні програмні продукти: консольне представлення, віконна форма, веб-сайт та інші. Мова програмування C# один з найпопулярніших способів створення додатків у світі. Конкурентами C# є мови програмування Python, Java,

JavaScript. С# мова програмування високого рівня, разом з тим вона є дуже простою в читанні і використанні. Тому логічно її використовувати для навчання студентів. Студенти дуже швидко опановують мову програмування С#. Хтось зауважить, що синтаксис написання коду за допомогою Python ще простіший, але після python важко вивчати інші мови програмування. Після опанування мови С# ви зможете легко вивчити і користуватись С, С++, Java та інші. На мові програмування С# періодично пишуть код близько тридцяти відсотків розробників у світі. Якщо це невеликий проект розробники прописують код на Python, щоб перевірити алгоритм, а потім вже реалізують його на С#. Тому що С# буде більш оптимальним і продуктивним. Серед українських програмістів в 2020 році мова С# посіла третє місце за рейтингом опублікованим порталом Dou. Хоча в світі С# за різними оцінками в трійку лідерів не входить.

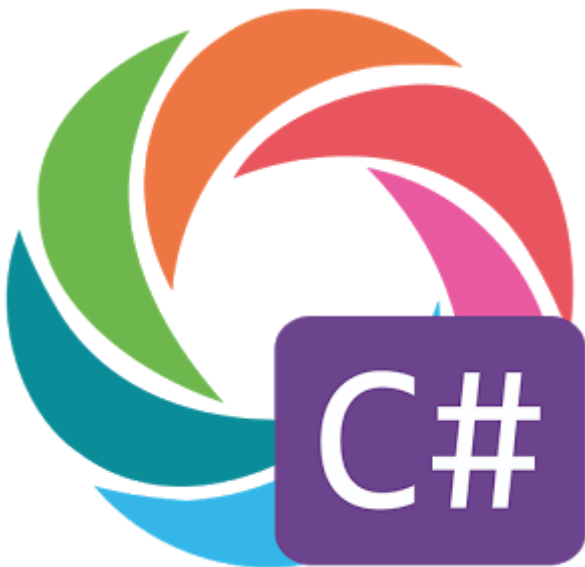


Рисунок 1.4 – С#

Для реалізації проекту у вигляді веб сайта будемо користуватись ASP.NET Core MVC побудований на основі маршрутизації ASP.NET Core, потужного компонента відображення URL-адрес, який дозволяє створювати програми, які мають зрозумілі та доступні для пошуку URL-адреси. Це дозволяє вам визначити шаблони імен URL-адрес

вашого додатка, які добре працюють для оптимізації пошукової системи (SEO) та для генерації посилань, не враховуючи, як організовані файли на вашому веб-сервері. Ви можете визначити свої маршрути, використовуючи зручний синтаксис шаблону маршруту, який підтримує обмеження значення маршруту, значення за замовчуванням та необов'язкові значення.

Маршрутизація на основі конвенцій дозволяє глобально визначити формати URL-адрес, які приймає ваша програма, і те, як кожен із цих форматів відображається до конкретного методу дії на даному контролері. Коли отримується вхідний запит, механізм маршрутизації аналізує URL-адресу та узгоджує її з одним із визначених форматів URL-адрес, а потім викликає метод дій відповідного контролера.

ASP.NET Core – покращена версія ASP.NET. Так ASP.Net Core має дещо нові погляди. Сьогодні близько сорока відсотків усіх сайтів створенні за допомогою технології ASP.NET. Платформа веб розробки ASP.NET почала використовуватись близько двадцяти років тому і поділяє з PHP лідируючі позиції. ASP.NET розширює платформу .NET за допомогою інструментів та бібліотек, спеціально призначених для створення веб-програм. Такі як:

- Синтаксис шаблону веб-сторінок, відомий як Razor, для створення динамічних веб-сторінок за допомогою C #.
- Базовий фреймворк для обробки веб-запитів на C # або F #.
- Бібліотеки загальних веб-моделей, таких як Model View Controller (MVC)
- Система автентифікації, яка включає бібліотеки, базу даних та сторінки шаблонів для обробки логінів, включаючи багатфакторну автентифікацію та зовнішню автентифікацію в Google, Facebook та інші.

- Розширення редактора для виділення синтаксису, заповнення коду та інших функціональних можливостей спеціально для розробки веб-сторінок

Можна виокремити наступні функції для створення кращих додатків за допомогою ASP.NET Core:

- Кросплатформеність (Можна працювати у різних операційних ситуаціях);
- Висока продуктивність(Програма легша і швидша);
- Використання асинхронних шаблонів програмування;
- Уніфіковані фреймворки MVC та Web API;
- Різні режими розробки (Можна використовувати різні режими роботи над проектом);
- Міжсайтовий захист підробки;
- Веб-програми, що розміщуються самостійно;
- Фільтри дій (Фільтри використовуються для визначення кешування, обробки помилок, авторизації або будь-якої спеціальної логіки, яку ви хочете застосувати. );
- Розширюване кешування вихідних даних (функція дозволяє ASP.NET кешувати вихідні дані, створені сторінкою, та обслуговувати цей кешований вміст для майбутніх запитів. Він зберігає дані, які не оновлюються часто, і виводить ці дані з кешованого місця);
- Глобалізація та локалізація;
- Робота з доаткамиAPI;

ASP.NET Core був приємним оновленням порівняно з попередніми версіями[9].

ASP.Net Core дозволяє створює веб-сервіси, які можуть використовуватись на звичайних ПК, смартфонах, планшетах в різноманітних браузерах. Дуже добре, що ASP.Net Core використовує

MVC фреймворк (Model-View-Controller). Це дозволить під час тестування, використовувати модульне тестування.

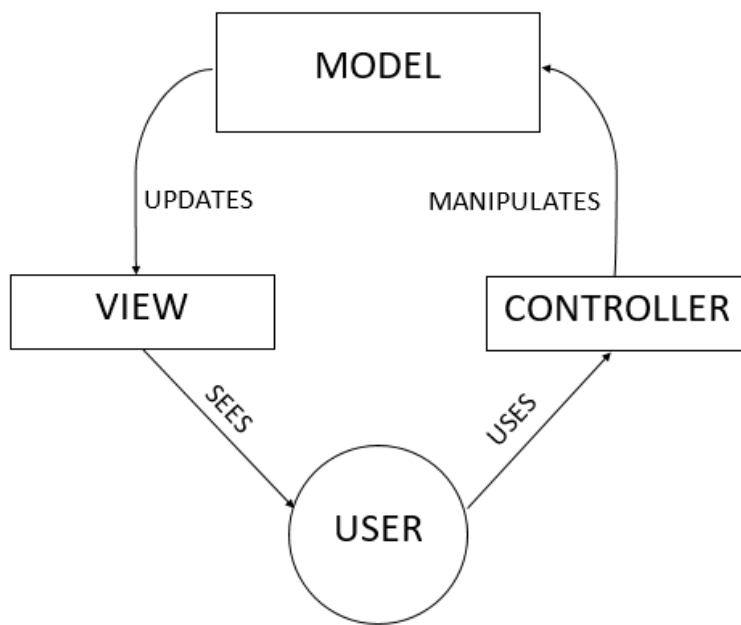


Рисунок 1.5 - Структура паттерну MVC

ASP.NET Core застосовує фреймворк MVC версії 1.0. Патерн MVC це нова ідея в різних додатках, він використовувався ще в кінці сімдесятих років. На приклад компанія Херох застосовувала його через мову Smalltalk, для організації компонентів в програмному забезпеченні для роботи з графікою .

Концепція MVC паттерна використовує поділ в програмного продукту на три компоненти:

- Модель (model)
- Представлення (view)
- Контролер (controller)

Модель це дані про об'єкти предметної та зв'язки між ними. Звичайно, об'єкти моделей найкраще зберігати в базі даних. У MVC моделі поділяються на два основні типи: моделі домену, які описують принцип керування даними, та моделі представлень, які використовуються для представлення та передавання даних. Крім того, модель може крім

даних зберігати логіку керування цими даними. В той самий момент модель не має зберігати логіку взаємодії з адресантом та не повинен містити правила обробки запитів. Ще модель не повина зберігати логіку відображення даних.

Представлення зрозуміло що відповідає за інтерфейс тобто спосіб відображення даних, дуже часто - це веб-сторінка, або додаток або і перше і друге. Ще представлення може мати логіку, яка відповідає за відображення даних. Представлення не може мати правила обробки запитів або керування даними.

Контролер – центральний компонент MVC, який надає зв'язок між користувачами моделю та є частиною інтерфейсу, який відповідає за обробку запитів користувача. Контролер отримує дані від користувача, опрацьовує та зберігає в моделі. Крім того контролер генерує представлення.

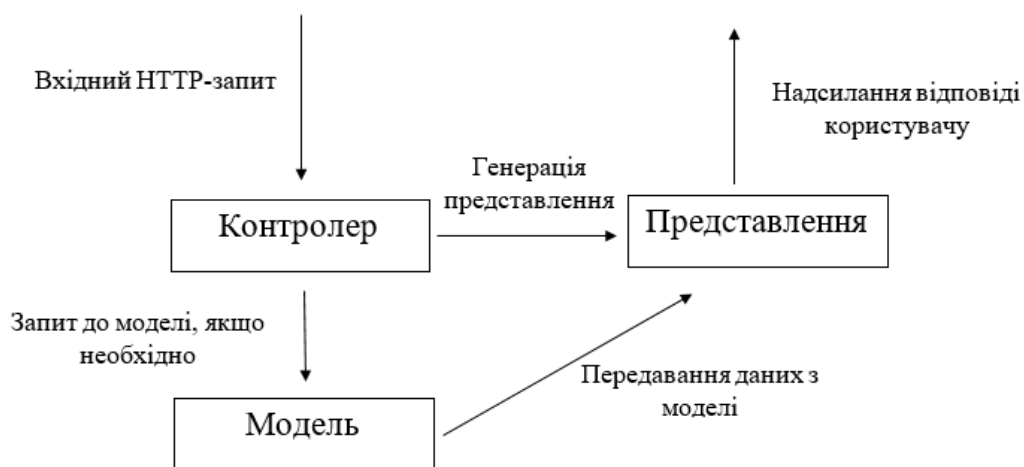


Рисунок 1.6 - Відношення між елементами паттерну MVC

У такій схемі модель завжди незалежна частина. Якщо ви внесете зміни до контролера або подання це ніяк не вплине на модель. Контролер і представлення - відносно незалежні. Так, за допомогою представлення можна звернутися до деякого контролеру, а з контролера створити нове представлення. Але розробники

намагаються зробити так, щоб зміни в контролері ніяк не впливали на відображення і навпаки. Такий розділ частин програмного продукту дозволяє швидко вносити зміни до конкретного компоненту тестувати його та полегшує подальшу роботу з додатком. Також за кожен компонент може відповідати окрема команда розробників, яким не потрібно сильно вникати в принципи роботи інших. Таким чином ми можемо тестувати тільки представлення і не звертатися до контролера, або навпаки перевірити контролер не звертаючи увагу на представлення. В ASP.NET Core не обов'язково використовувати MVC. Можна створити проект, який не буде містити окремо компонентів моделі, контролера, чи представлення а усе реалізувати в одному класі. Але зрозуміло що застосування MVC набагато полегшує розробку можливість швидкої зміни та підтримки програмного продукту. Під час роботи з MVC зникає потреба в окремій розробці бази даних. Так за допомогою модуля Entity Framework створити базуу даних з об'єктів автоматично. ASP.NET Core легко дозволяє використати інші різні фреймворки в тому числі ckeditor і Bootstrap. [9]

Технологія ASP.NET MVC сьогодні на ринку займає близько 1%, де лідерство займають технології ASP.NET і PHP. Ми можемо навести приклади сайтів, які використовували ASP.NET MVC для створення:

- Microsoft <https://www.microsoft.com/uk-ua>
- DELL <https://www.dell.ua/>
- Приват банк <http://practive.pb.ua/>
- Foxtrot <https://www.foxtrot.com.ua/>
- Оформлення віз в електронному вигляді для в'їзду в Україну <http://evisa.mfa.gov.ua/>[10]

Microsoft Visual Studio[11] від корпорації Майкрософт – інтегроване середовище розробки програмного забезпечення, який дозволяє працювати з різними мовами програмування, на різних операційних



системах. Visual Studio дозволяє розробляти як консольні програми, програми з графічним інтерфейсом, в тому числі веб-сайти або Windows Forms. Є можливість розробки додатків для Android, iOS, Mac, Windows, а також хмарних і веб-додатків. Майкрософт гарантує з використанням Visual Studio: швидке написання коду, налагодження та тестування продуктів без особливих зусиль, постійне оновлення, розширення та налаштування на власний смак, а також можливість ефективної спільної роботи. На даний момент остання версія Visual Studio 2019. Загальний вигляд вікна, яке відображається при завантаженні Visual Studio 2019, початок роботи та створення нового проекту (рисунк 1.7) Visual Studio Community надає доступ до широкого спектру послуг безкоштовно.

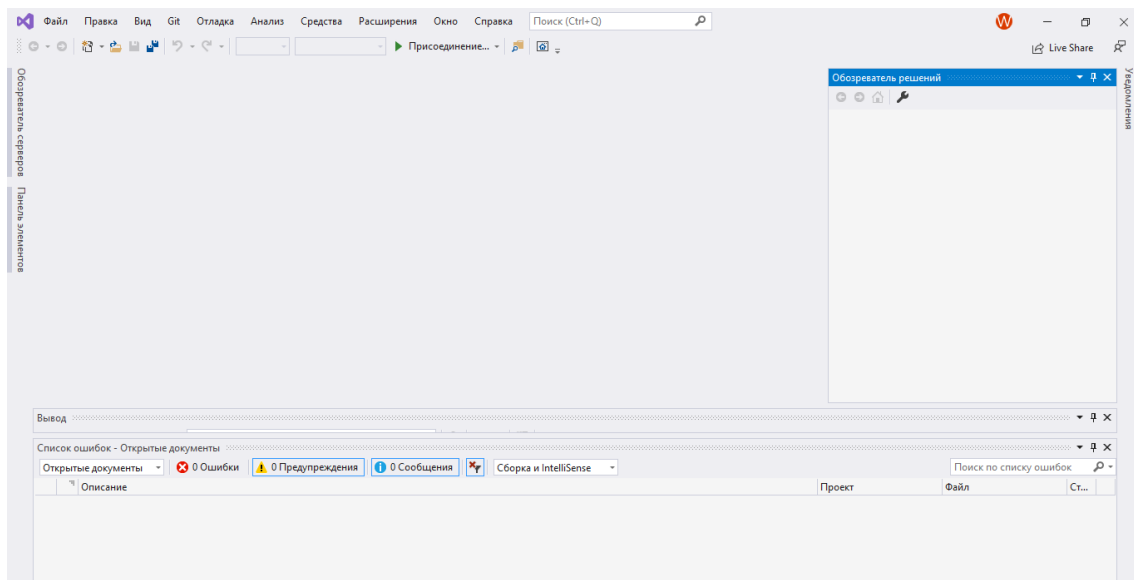


Рисунок 1.7— Visual Studio 2019

Отже вибрано мову програмування C#, та технологію ASP.NET Core з застосуванням патерну MVC. А створюватися система підтримки дистанційного навчання буде в середовищі Visual Studio 2019

### 1.3.2 Вибір СУБД та опис її фізичної моделі

Які є типи баз даних

Реляційні бази даних. Реляційні бази даних стали домінуючими у 1980-х. Елементи реляційної бази даних організовані як набір таблиць із

стовпцями та рядками. Технологія реляційних баз даних забезпечує найбільш ефективний та гнучкий спосіб доступу до структурованої інформації.

Об'єктно-орієнтовані бази даних. Інформація в об'єктно-орієнтованій базі даних представлена у вигляді об'єктів, як і в об'єктно-орієнтованому програмуванні.

Розподілені бази даних. Розподілена база даних складається з двох або більше файлів, розташованих на різних сайтах. База даних може зберігатися на декількох комп'ютерах, розташованих в одному фізичному місці або розкиданих по різних мережах.

Сховища даних. Центральне сховище даних, сховище даних це тип бази даних, спеціально розроблений для швидких запитів та аналізу.

Бази даних NoSQL. NoSQL, або нереляційна база даних, дозволяє зберігати та маніпулювати неструктурованими та напівструктурованими даними (на відміну від реляційної бази даних, яка визначає, яким чином повинні бути складені всі дані, введені в базу даних). Бази даних NoSQL стали популярнішими, оскільки веб-програми стали більш поширеними та складнішими.

Графічні бази даних. База даних графіків зберігає дані в термінах сутностей та взаємозв'язків між сутностями.

Бази даних OLTP. База даних OLTP це швидка, аналітична база даних, призначена для великої кількості транзакцій, що виконуються декількома користувачами.

Архітектуру бази даних інформаційної системи для забезпечення ефективності дистанційного навчального процесу з використанням мови C# було обрано реляційну.

Таблиця являє собою набір елементів даних (значення).

Частину даних у файлі називають записом.

Кожен елемент запису називається полем.

Таблиця, запис, поле, рядок і стовпець

Один фрагмент даних або запис називається рядком .

Кожен елемент або поле називається стовпцем .

Первинний ключ, унікальний та нульовий.

Полю часто відводиться важлива роль у базі даних, коли це відбувається, ми називаємо це поле Первинним ключем . У нашому прикладі Код студента є первинним ключем.

Унікальне значення є значенням, яке не може бути ні в якому разі повторено. Деякі поля крім первинних ключів можуть бути порожніми або нульовими. Ключ: один або кілька стовпців таблиці бази даних, які використовуються ідентифікації рядків у таблиці. наприклад, якщо ви сортували студентів за датою народження, то ключовим є поле дата народження.

Первинний ключ – це одне з полів, які однозначно ідентифікують рядок у таблиці. Для цього можна вибрати індекс (номер по порядку), ідентифікаційний код, номер паспорта, номер абетки і таке інше. Первинний ключ ні в якому разі не може бути нульовим (порожнім).

Зовнішній ключ – це взаємозв'язок між полями у кількох таблицях базданих (одна з яких індексується), призначених для забезпечення послідовності даних.

Композитний ключ: ключ, що складається з одного або декількох стовпців. Первинний ключ можна сформувати за допомогою полів  
Природний ключ: складений первинний ключ, який складається з атрибутів (полів), які вже існують у реальному світі (номер абетки, номер паспорта, ідентифікаційний номер).

Нормалізація - це процес організації полів і таблиць реляційної бази даних для мінімізації надмірності. Нормалізація, як правило,

передбачає поділ великих таблиць на менші (і менш зайві) таблиці та визначення взаємозв'язків між ними. Мета полягає в ізоляції даних, щоб доповнення, видалення та модифікація поля могли бути зроблені лише в одній таблиці, а потім розповсюджені через решту бази даних за допомогою визначених відносин.

Процес нормалізації, як вперше запропоновано в сімдесятих роках, приймає схему відношень через ряд тестів, щоб підтвердити, чи відповідає вона певній нормальній формі.

Таким чином, процес, який протікає зверху вниз, оцінюючи кожне відношення за критеріями нормальних форм та декомпозиційні відношення, за необхідності, може розглядатися як реляційний дизайн шляхом аналізу. Спочатку Кодд запропонував три

нормальні форми, які він назвав першою, другою і третьою нормальними формами. Більш чітке визначення 3НФ - зване нормальною формою Бойца-Кодда (БКНФ) - було запропоновано пізніше Бойсом і Коддом. Усі ці нормальні форми базуються на одному аналітичному інструменті: функціональних залежностях між атрибутами відношення. Пізніше було запропоновано четверту нормальну форму (4НФ) та п'яту нормальну форму (5НФ), засновану на концепціях багатозначних залежностей та залежності об'єднання відповідно

Нормалізацією даних можна вважати процес аналізу заданих схем відношень на основі їх FD і первинних ключів для досягнення бажаних властивостей мінімізації надмірності та мінімізації аномалій вставки, видалення та оновлення

процес, щоб дизайн мав послідовно кращу якість. Незадовільні схеми відношень, які не відповідають певним умовам, звичайні тести форми розкладаються на менші схеми відношень, які відповідають тестам, і, отже, мають

бажані властивості. Таким чином, процедуру нормалізації надає дизайнер баз даних

з наступним:

- Формальна структура для аналізу схем відносин на основі їх ключів та функціональних залежностей серед їхніх атрибутів
- Серія тестів нормальної форми, які можна проводити на окремих схемах відношень, щоб реляційна база даних могла нормалізуватися до будь-якого бажаного ступеня нормальної форми [12]

MySQL отримав позитивні відгуки, багато користувачів кажуть, що це надзвичайно добре зазвичай, і що інтерфейси для розробників є, і документація (не кажучи вже про відгуки в реальному світі через веб-сайти тощо) дуже, дуже добре. Він також був перевірений як швидкий, стабільний і справжній багатокористувацький багатопотоковий сервер бази даних sql. [13]

SQL використовується для спілкуватися з базою даних. За даними ANSI (Американський національний інститут стандартів), це така стандартна мова для реляційних систем управління базами даних. Використовуються оператори SQL для виконання таких завдань, як оновлення даних у базі даних або отримання даних з бази даних. Дещо Типовими реляційними системами управління базами даних, які використовують SQL, є: Oracle, Sybase, Microsoft SQL Server, Access, Ingres тощо.

Стандарт SQL протягом багатьох років зазнав багато змін, які додали а велика кількість нових стандартних функціональних можливостей, таких як підтримка XML, тригери, звичайні відповідність виразів, рекурсивні запити, стандартизовані послідовності та багато іншого.

Мова SQL базується на декількох елементах. Для зручності розробників SQL усіх необхідними мовними командами у відповідних

системах управління базами даних зазвичай виконується через певний інтерфейс командного рядка SQL. Ці команди можуть бути

згруповані за такими напрямками:

- Речення - речення є компонентами операторів та запитів;
- Вирази - вирази можуть створювати скалярні значення або таблиці, які складаються з стовпці та рядки даних;
- Предикати - вони визначають умови, які використовуються для обмеження ефекту від операторів та запитів, або для зміни потоку програм;
- Запити - запит отримує дані на основі заданих критеріїв;
- Звіти - за допомогою операторів користувач може контролювати транзакції, потік програми, з'єднання, сеанси або діагностика. У системах баз даних для відправки використовуються оператори SQL запити від клієнтської програми до сервера, де зберігаються бази даних. У відповідь, сервер обробляє оператори SQL і повертається до клієнтської програми. Це дозволяє користувачі можуть виконувати широкий спектр операцій швидкої обробки даних із простих даних вхідні дані до більш складних запитів.

Для того, щоб вести студентів через елементарний синтаксис керівництва SQL, існує реляційна модель.

Висока продуктивність.

Не перший рік Microsoft підтверджує високу продуктивність SQL Server транзакційними тестів і тестами продуктивності сховищ даних. Версія 2019 відзначена відмінними результатами в наступних тестах. Підтримка постійної пам'яті (PMEM). Постійна пам'ять (Persistent Memory, PMEM) - це швидка пам'ять, що володіє можливістю зберігати дані після відключення живлення. Вона дозволяє обробляти дані in-memory, позбавляючи від необхідності передавати дані по каналах передачі і прискорюючи обробку запитів на 30% для інтенсивних

робочих навантажень введення-виведення. Будь-який файл SQL Server, поміщений на пристрій РММ, тепер доступний безпосередньо, міняючи стек зберігання операційної системи, використовуючи ефективні операції метасру.

### 1.3.3 Реалізація основних класів та методів

Далі продемонструємо реалізацію класів основних сутностей

```
public class Student{
    public int id { set; get; }
    public string name { set; get; }
    public string dateb { set; get; }
    public string login { set; get; }
    public string parol { set; get; }
}

public class Lesson {
    public int id { set; get; }
    public string topic { set; get; }
    public bool lesonshow { set; get; }
    public string homework { set; get; }
    public int cursid { set; get; }
    public virtual Curs curs { set; get; }
}

public class Group {
    public int id { set; get; }
    public string name { set; get; }
    public int cursid { set; get; }
    public List<Student> students { set; get; }
}

public class Curs{
    public int id { set; get; }
    public string namecurs { set; get; }
```

```
public string descurs { set; get; }  
public string img { set; get; }  
public List<Lesson> lessons { set; get; }  
public List<Group> groups { set; get; } }
```

## **2. ВИКОРИСТАННЯ ПРОГРАМНОЇ СИСТЕМИ**

### **2.1 Розгортання програмної системи та системні вимоги**

Програма складається з трьох компонентів:

- а) Середовища запуску Visual Studio
- б) Власне, програма (Disted.exe)
- в) База даних (БД, файл Disted.db).

Системні вимоги для використання Visual Studio 2019[11]

Операційні системи:

Windows 10 version 1703 or higher: Home, Professional, Education, and Enterprise (LTSC and S are not supported)

Windows Server 2019: Standard and Datacenter

Windows Server 2016: Standard and Datacenter

Windows 8.1 (with Update 2919355): Core, Professional, and Enterprise

Windows Server 2012 R2 (with Update 2919355): Essentials, Standard, Datacenter

Windows 7 SP1 (with latest Windows Updates): Home Premium, Professional, Enterprise, Ultimate

Процесор 1,8 ГГц або швидший. Чотириядерний або краще рекомендований

2 ГБ оперативної пам'яті; Рекомендовано 8 ГБ оперативної пам'яті (мінімум 2,5 ГБ, якщо працює на віртуальній машині)



Місце на жорсткому диску: мінімум 800 МБ до 210 ГБ вільного місця, залежно від встановлених функцій; типові установки вимагають 20-50 ГБ вільного місця.

Швидкість жорсткого диска: для підвищення продуктивності встановіть Windows і Visual Studio на твердотільний накопичувач (SSD).

Відеокарта, що підтримує мінімальну роздільну здатність дисплея 720p (1280 на 720); Visual Studio найкраще працюватиме з роздільною здатністю WXGA (1366 на 768) або вище.

Для роботи з базою даних потрібно встановити локальний сервер

Якщо розмістити сайт з базою даних на хостингу тоді працювати можна з будь якого пристрою на якому є можливість працювати збраузерами та інтернетом.

## 2.2 Опис типових схем використання системи

Як тестування розробленого продукту, було проведена ручна перевірка функцій додатку. Результат перевірки у вигляді інструкції використання додатку наведено в цьому розділі. Багато в чому веб-дизайн стосується побудови веб-інтерфейсу, який дозволяє відвідувачам наказувати сайту діяти корисним чином. Усі хороші власники веб-сайтів надають пріоритет користувачеві. Зрештою, важливо забезпечити, щоб відвідувачі могли легко взаємодіяти з вашим сайтом. Дизайн інтерфейсу веб-сайту має вирішальне значення для успіху сайту та його взаємодії з користувачами.

Щоб працювати в системі студентів потрібно пройти реєстрацію. Для цього потрібно придумати Логін – ім'я для ідентифікації в системі (не менше чотирьох і не більше двадцяти п'яти літер латинського алфавіту, цифр). Ввести своє справжнє ім'я. Придумати пароль, який ніколи нікому не можна показувати та розповідати. Пароль потрібно

повторити (щоб перевірити чи запам'ятав учень свій пароль і як його вводити). Ще потрібно внести електрону пошту.

• Реєстрація користувача

Логін:	<input type="text"/>	Придумайте ім'я для Вашої ідентифікації в системі (не менше 4-х і не більше 25-ти латинських літер, цифр).
Ваше повне ім'я:	<input type="text"/>	Ваше справжнє ім'я українською мовою.
Пароль:	<input type="password"/>	Придумайте пароль (від 6 до 20 латинських літер/символів/цифр). Тримайте його в таємниці.
Підтвердження пароля:	<input type="password"/>	Повторно введіть Ваш пароль.
Діюча електронна адреса:	<input type="text"/>	Ваша електронна скринька.

## Рисунок 2.1 Сторінка реєстрації студентів

Робота з сайтом є досить простою та інтуїтивно зрозумілою. На кожній сторінці сайту розміщене вертикальне меню яке знаходиться в горі вікна робочої області браузера. Меню містить 5 кнопок:

- Disted
- Головна
- Групи
- Курси
- Про нас

Кнопки дозволяють відвідувати сторінки сайту.

На головній сторінці ми бачимо інформацію про те для чого призначений даний сайт (Рисунок 2.2)

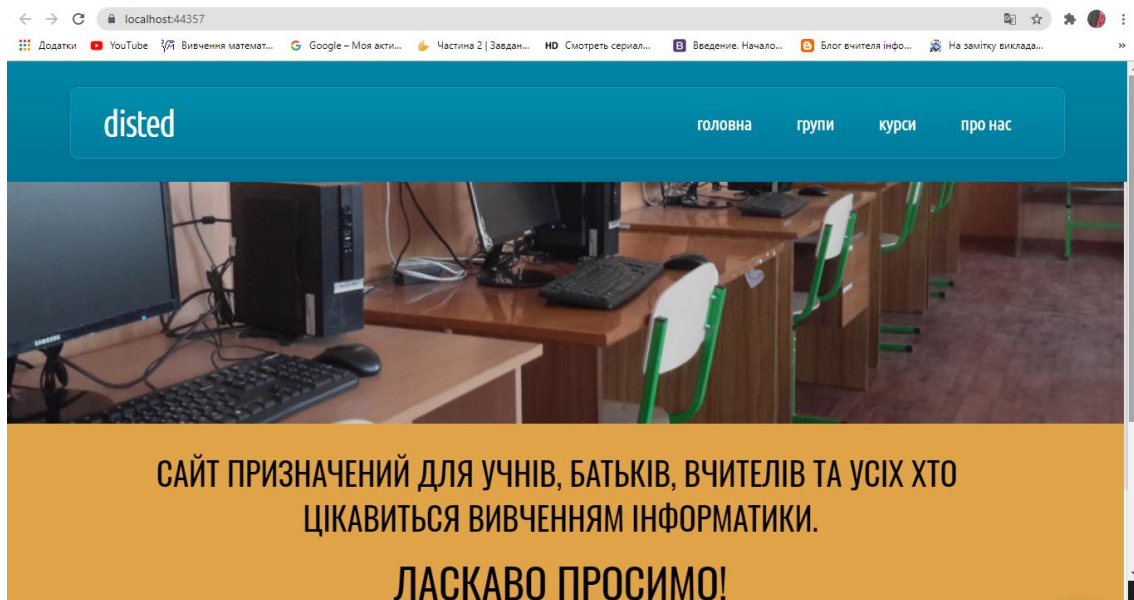


Рисунок 2.2 Головна сторінка сайту

Учень/студент може обирати клас чи групу в якому бажає навчатися натиснувши кнопку Групи Рисунок 2.3

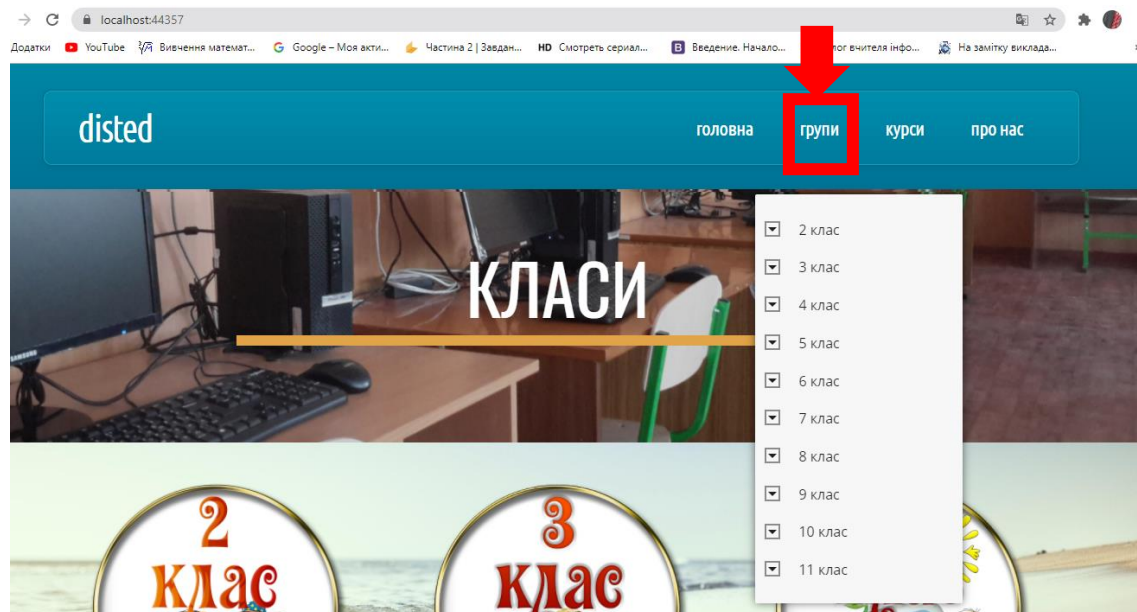


Рисунок 2.3 – Сторінка групи

Коли обрали групу ви зможете натиснувши вкладку курси переглядати інформацію виставлену вчителем на сторінці курси. Крім того, вчитель може використовуючи посилання додати відео з YouTube або інтерактивні вправи з learningapps.org та підключати інші сервіси. (Рисунок 2.4)

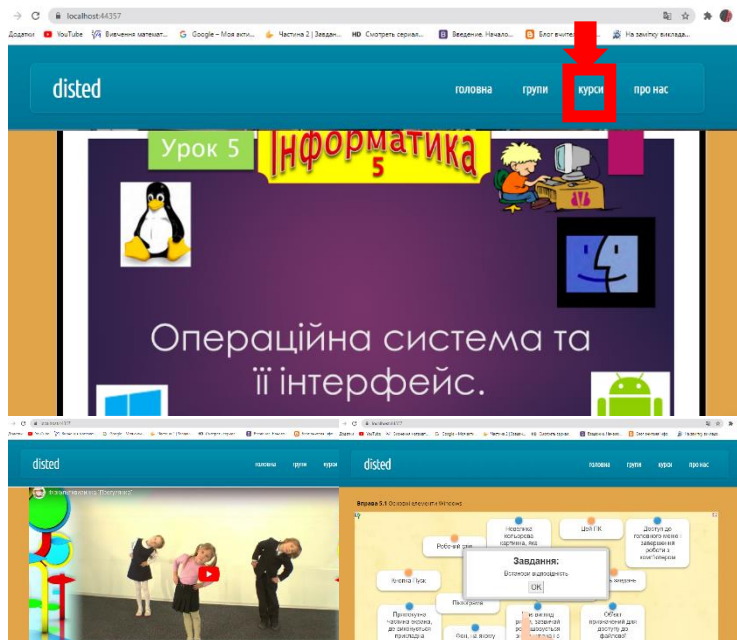


Рисунок 2.4 Сторінка курси

Створений сайт працює в нормальному режимі і дозволяє створювати та переглядати уроки. Далі сайт може бути покращений і додані нові функції. Учень зможе відправляти виконанні завдання. Викладач зможе виставляти оцінки за виконанні завдання.

### **3. ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ**

Під час створення будь якого проекту , є грошові витрати. Важливо правильно зробити оцінку фінансових затрат, корисність, продуктивність, економічну ефективність проекту Наукові дослідження та розробки, на відміну від корпоративних, не завжди мають за мету отримання прибутку або ж іншої матеріальної вигоди. Дуже часто, проекти наукового спрямування є економічно не вигідними. Однак саме завдяки їм відбувається прогрес у світі. Дуже багато наукових відкриттів робилися не загроші. Багато галузей в Україні не можуть похвалитися гарним фінансуванням. Особливо в галузі освіти очікувати вливання коштів не приходиться. Викладачі та вчителі велику частку роботи виконують самостійно, купуючи розхідні матеріали за власний кошт. Добре що хоч наукова і освітня діяльність може розраховувати на підтримку зовнішніх інвестицій та підтримку держаних інститутів, міжнародних та українських меценатів.

Проектування та створення веб-сайту вимагає зваженого планування, коли мова йде про кошторисні витрати та доставку. І терміни, і остаточний рахунок повинні бути встановлені з самого початку, щоб гарантувати, що проект вважається успішним як для клієнта, так і для дизайн-студії. Хороші оцінки можуть означати різницю між прибутковим проектом та завищеною ціною.

Зі зростанням важливості веб-додатків загалом з питань бізнесу багато практиків вважають вимірювання веб-додатків особливо цінною сферою постійних комерційних досліджень. Раніше успіх веб-додатків визначався тим, що вони вийшли на ринок. У наш час якість програмного забезпечення стала важливим фактором підвищення конкурентоспроможності. Якісні програмні додатки вимагають відповідної розробки програмного забезпечення та процесів управління проектами. Для керівників проектів оцінка витрат є одним із найважливіших кроків на початку будь-якого нового програмного

проекту. Точні кошториси витрат є важливим елементом для того, щоб забезпечити конкурентоспроможні заявки та залишатися успішними на ринку. Завищення та заниження очікуваних витрат мають значний вплив на репутацію та конкурентоспроможність компанії. Промисловість все ще повідомляє про труднощі з визначенням точних кошторисів витрат, навіть незважаючи на те, що дослідження методів оцінки вартості програмного забезпечення розпочато вже на початку 1960-х.

Протягом останніх десятиліть багато досліджень було зосереджено на розробці точних моделей оцінки витрат. Однак опитування показують, що насправді ними користується лише дуже мало організацій [16].

Існує ряд практично релевантних причин невикористання моделей оцінки витрат: Багато моделей оцінки витрат не дуже точні [25]. Досвід показує, що оцінки на основі моделей працюють не набагато краще, ніж оцінки виключно на основі експертного судження. Ще однією причиною рідкісного використання моделей оцінки витрат на практиці є відсутність достатньої кількості явних даних минулих проектів для систематичного побудови моделі оцінки витрат. Багато організацій не мають достатньо ресурсів для необхідного вимірювання методи та засоби. Збір даних - це трудомісткий процес, який потрібно ретельно планувати, оскільки послідовність і повнота є двома ключовими рушіями для корисності даних для оцінки зусиль. Лише дані про нещодавні проекти дозволяють проводити дослідження оцінки витрат, які мають значення для поточних проектів.

Ще однією причиною скромного використання моделей оцінки витрат є відповідна оцінка розміру програмного продукту.

Розмір програмного продукту визнаний ключовим фактором витрат для розробки, але він вимагає точних методів вимірювання та адекватних заходів, яких часто бракує у промисловості. Нарешті, багато моделей оцінки витрат базуються на стандартних та / або

методах чорного ящика для отримання кошторису витрат. Відсутність прозорості у багатьох методах оцінки витрат не дозволяє достатньо калібрувати конкретні потреби організації

Для економічного оцінювання нематеріальних засобів можна використати міжнародні стандарти оцінки розрахунку вартості об'єктів інтелектуальної власності, розроблені TIAVIS (The International Assets Valuation Standards Committee). Рада з міжнародних стандартів оцінки (IVSC) - це незалежна некомерційна організація, яка прагне підвищити якість професії оцінки. Наша головна мета - створити довіру та довіру громадськості до оцінки шляхом вироблення стандартів та забезпечення їх загального прийняття та впровадження для оцінки активів у всьому світі. Ми вважаємо, що Міжнародні стандарти оцінки (MCB) є фундаментальною частиною фінансової системи, поряд з високим рівнем професіоналізму у їх застосуванні.

Оцінки широко використовуються та покладаються на них на фінансових та інших ринках, будь то для включення до фінансової звітності, для дотримання нормативних вимог чи для підтримки забезпеченого кредитування та операційної діяльності. Міжнародні стандарти оцінки (IVS) - це стандарти для проведення оціночних завдань із використанням загальновизнаних концепцій та принципів, що сприяють прозорості та послідовності в практиці оцінки. IVSC також просуває провідні практичні підходи до поведінки та компетенції професійних оцінювачів.

Рада зі стандартів IVSC є органом, відповідальним за встановлення IVS. Рада має самостійність у розробці порядку денного та затвердженні своїх публікацій. Розробляючи IVS, Правління:

- дотримується встановленої належної процедури при розробці будь-якого нового стандарту, включаючи консультації із зацікавленими сторонами (оцінювачами, користувачами послуг з оцінки,

регуляторами, професійними організаціями з оцінки тощо) та публічне викриття всіх нових стандартів чи істотних змін існуючих стандартів,

- підтримує зв'язок з іншими органами, які виконують функції встановлення стандартів на фінансових ринках,
- проводить інформаційно-роз'яснювальні роботи, включаючи круглі столи із запрошеними виборцями та цілеспрямовані дискусії з конкретними користувачами або групами користувачів. Метою IVS є підвищення довіри та довіри користувачів оціночних послуг шляхом встановлення прозорості та послідовної практики оцінки. Стандарт виконує одне або кілька з наступного:
- визначити або розробити загальновизнані принципи та визначення,
- визначити та оприлюднити міркування щодо проведення оціночних завдань та складання звітності про оцінки,
- визначити конкретні питання, які потребують розгляду, та методи, які зазвичай використовуються для оцінки різних видів активів або зобов'язань.

Розроблення ПЗ можна виконати кількома способами: об'єктноорієнтованим і процедурним. [17]. У цій роботі розглядаються сильні та слабкі сторони існуючих методів оцінки розміру, розглядається суть оцінки розміру програмного забезпечення та представлена модель оцінки розміру програмного забезпечення, яка може забезпечити більш точні оцінки розмірів. Запропонований методи використовує характеристики об'єктно-орієнтованих систем, природну відповідність між специфікацією та реалізацією, щоб дозволити користувачам складати кращі оцінки розмірів на ранніх стадіях циклу розробки програмного забезпечення. За допомогою процедурного підходу метод також забезпечує довірчий інтервал для отриманих оцінок розміру. Також розглянуто співвідношення між представленою



моделлю розміру програмного забезпечення та оцінкою вартості проекту

Спочатку роботи потрібно скласти технічне завдання, воно є першочерговим. Ціна складання технічного завдання складає до десяти відсотків від запланованої вартості розробки. Технічне завдання надає виклад про передумови, цілі та мету пропонованого проекту. Шаблон технічного завдання включає ряд критеріїв, необхідних для прийняття стратегічних проектних рішень. Цей документ визначає заходи, які потрібно здійснити, та вказує питання, бюджет та експертні знання, пов'язані з проектом.

У таблиці 3.1 наведення вартість технологічного процесу з урахуванням усіх зарплат працівників. Усі суми наведені в українській національній валюті.

Таблиця 3.1 – Розрахункова вартість технологічного процесу розробки

Посада	Місячний оклад, грн.	Денний оклад, грн.	Об'єктно-орієнтований підхід		Процедурний підхід	
			Днів	Сума, грн	Днів	Сума, грн.
Керівник	19800	900	20	18000	25	22500
Програміст	19140	870	32	27840	35	30450
Розробник тестів	16940	770	5	3850	6	6620
Тестування						
Дизайнер	17600	800	7	5600	7	5600
Додаткова зар. плата 20%			44	1158	48	12634
Фонд оплати праці 36,77%			20330,14		23227,60	
Всього витрат на зар. плату			86678,14		99031,60	

Військовий збір 1,5%	1300,18	885,48
Єдиний соціальний внесок 3,6%	3120,41	3565,14
ПДВ, 15%	1301,72	1494,74
Всього	104100,44	118936,96

Для об'єктно-орієнтованого підходу тривалість роботи над веб сайтом становить 44 робочих дня, що містить роботу програміста, керівника, роботу тестувальника та дизайнера. Сума витрат на заробітню плату становить 104100,44 гривні. Для процедурного підходу до розробки з часом розробки - 48 днів, витрати у вигляді зарплати становлять 118936,96 гривень.

З цих сум обов'язково утримуються відрахування на зарплату:

- Військовий збір 1,5%
- Єдиний соціальний внесок 3,6%
- ПДВ, 15%

До окремих витрат також відносяться витрати на куповані вироби (матеріальне забезпечення таке як: папір, заправка принтера, дошка, фломастер) та спец обладнання(комп'ютер, принтер, операційна система, програмне забезпечення), накладні витрати.

Матеріальні витрати визначаються як добуток кількості витрачених матеріалів та їх ціни (формула 3.1).

$$M_{Bi} = q_i \cdot p_i \quad (3.1)$$

де  $p_i$  – ціна матеріалі  $i$ -го виду;  $q_i$  – кількість витрачених матеріалів  $i$ -го виду. Матеріальні витрати в рамках проекту наведені в таблиці 3.2. Загальна сума матеріальних витрат становить 2188 гривень.

Таблиця 3.2 – Матеріальні витрати

Найменування ресурсу	Кількість, шт	Ціна одиниці, грн	Загальна сума, грн
Пачка паперу А4	2	200	400
Заправка принтера	2	160	320
Дошка для записів	1	1000	1000
Фломастер	10	20	200
Всього			1920

Розрахунок витрат на електроенергію одиниці обладнання визначаються за формулою 3.2:

$$З_{\text{в}} = W \cdot T \cdot S, \quad (3.2)$$

де  $S$  – вартість кіловат-години електроенергії,  $T$  – кількість годин роботи обладнання;  $W$  – необхідна потужність, кВт;

$S = 2,50$  грн/кВт·год.

$З_{\text{в1}} = 1,2 \cdot 352 \cdot 2,50 = 1056,00$  грн;

$З_{\text{в2}} = 1,2 \cdot 384 \cdot 2,50 = 1152,00$  грн;

Таблиця 3.3 – Перелік необхідного обладнання

Найменування	Кількість, шт	Ціна, грн	Сума, грн
ПК	3	22400	67200
Принтер	1	8400	8400
Середовища розробки	3	безкоштовно	безкоштовно
Операційна система	3	безкоштовно	безкоштовно

Всього більше 1000 грн.	75600	
Всього витрат на амортизацію	7531,48	8216,16
Всього	83131,48	83816,16

Розрахунок суми амортизаційних відрахувань. Будь яка оргтехніка чи персональні комп'ютери відносяться до четвертої групи основних фондів. Для цієї групи річна норма амортизації дорівнює 60 %, вартість яких перевищує 1000 грн. і визначається:

$$A = \frac{C_B \cdot N_A \cdot T_{\text{ФАК}}}{T_{\text{РІК}}} \quad (3.3)$$

де  $T_{\text{РІК}}$  – річний робочий фонд часу, год;  $T_{\text{ФАК}}$  – фактичний час роботи обладнання, год; грн;  $N_A$  – норма амортизаційних відрахувань в рік, %;  $C_B$  – збалансова вартість обладнання;

В цій формулі норма відрахувань на амортизацію рівна  $N_A = 0,6$ .

Балансова вартість обладнання вказана в таблиці 3.3 і рівна  $C_B = 74600$  гривень.

Річний робочий фонд часу візьмемо за  $T_{\text{ГОД}} = 2002$  годин.

З них реальний фактичний робочий час становить  $T_{\text{ФАК}} = 44$  години згідно об'єктно-орієнтованого підходу та  $T_{\text{ФАК}} = 48$  годин згідно процедурного підходу.

Згідно вищезгаданої формули витрати на амортизацію становлять 983,73 гривень та 1073,16 гривень для кожного підходу відповідно.

Накладні витрати стосуються поточних комерційних витрат, які безпосередньо не пов'язані зі створенням продукту чи послуги. Це важливо для цілей складання бюджету, а також для визначення того, скільки компанія повинна брати за свою продукцію чи послуги, щоб отримати прибуток. Коротше кажучи, накладні витрати - це будь-які витрати, понесені на підтримку бізнесу, не пов'язані безпосередньо з

певним продуктом чи послугою. Наші накладні витрати наведені в таблиці. Залежно виду діяльності господарюючого об'єкта від організаційно-правового, накладні витрати становлять від двадцяти до 60 відсотків суми зарплати працівників. Нехай вона буде дорівнювати 40%, що становить 41640,18 грн для об'єктно-орієнтованого і 47594,78 грн для процедурного підходу розробки.

Повна вартість програмного продукту дорівнює сумі витрат на його виробництво: 235342 грн застосовуючи об'єктно-орієнтований підхід, 256988 грн при процедурному підході розробки.

Визначення "продуктивності" з точки зору виробництва є відношенням виробництва до вкладеного у виробництво і є показником ефективності.

Економічна ефективність (Е) можна порахувати за формулою 3.4 яка показує відношення прибутку до собівартості продукту:

$$E = \frac{\Pi}{C_B} \quad (3.4)$$

де  $\Pi$  – прибуток,  $\Pi = B - C_B$ ;  $C_B$  – собівартість.

В випадку нашої розробки інформаційної системи для забезпечення ефективності дистанційного навчального процесу з використанням мови С#, маючи некомерційний проект без економічно корисного результату, можна прогнозувати, що економічна ефективність прямує до 0 у обох випадках.

Але за умови надходження фінансів з будь яких джерел на реалізацію нашого проекту можна буде вважати ефективність проекту рівною 1. Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень ( $T_{OK}$ ):

$$T_{OK} = \frac{1}{E} \quad (3.5)$$

В нашому випадку прямого прибутку немає. Окупність для розробки програмного забезпечення за ефективності дорівнює одиниці можна вважати теж рівною одиниці.

Додавши усі витрати нашого проекту. Загальна вартість робіт для процедурного підходу розробки становить 259230,76 грн. А для для об'єктно-орієнтованого підходу розробки – 314110,15 грн. В даному випадку реалізації проекту зрозуміло що варто вибрати об'єктно орієнтований підхід для розробки інформаційної системи для забезпечення ефективності дистанційного навчального процесу з використанням мови C#, адже це фінансово більш вигідно.

#### **4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ**

Люди дедалі частіше отримують зв'язок із цифровою інформацією, і це впливає на безпеку та здоров'я на виробництві. Одним із ключових впливів на безпеку та здоров'я на виробництві є те, що технологічні розробки за певних обставин змогли взяти на себе брудні, небезпечні та принизливі роботи, що виконувались раніше. робітниками. З цим пов'язаним розвитком стає все більше використання ІТ - тобто використання комп'ютерів для спроби відтворити мислення людини. ІТ все частіше використовується для підтримки безпеки та здоров'я працівників іншими способами, наприклад, при медичній діагностиці. Однак у деяких випадках використання технологій може призвести до різних негативних наслідків, хвороб пов'язаних з довгим сидінням за ПК. Це впливає на безпеку та добробут працівників. Ключовою зміною у світі праці стала «віртуалізація» роботи, що призвела до її збільшення попит на "доцільність" стосовно організації роботи, організації робочого часу та роботи на дистанції. Окрім зміни способу взаємодії людей на роботі (і справді вдома), розвиток і поширення цифровізації та ІТ дедалі більше стирає межу між роботою та рештою життя. Зростає розповсюдження таких практик, як робота на дистанції / мобільна робота, заснована на ІКТ, та час їх використання. Це може забезпечити нові можливості для людей та підприємств, в тому числі з точки зору охорони праці. Наприклад, робота на дистанції може зменшити час їзди на роботу та пов'язаний з цим стрес та ризик нещасних випадків на виробництві, а також може сприяти кращому балансу роботи та життя. Однак це також може спричинити проблеми з охороною праці, такі як необхідність управління психосоціальними ризиками, пов'язаними з самотністю, та можливим розмиванням меж між роботою та особистим життям, а також забезпечення ергономіки робочих станцій.

Тим часом, ймовірно, що працівники все частіше працюватимуть поза традиційними робочими місцями або працюватимуть віддалено. Незважаючи на те, що це може відвести працівників від шкідливих робочих місць, це може також познайомити їх з новими ризиками. Психосоціальні та організаційні фактори стануть дедалі важливішими у міру того, як змінюються тип і темп роботи, включаючи спосіб управління нею. Інші нові ризики можуть виникнути внаслідок збільшення інтерфейсів людина-машина, у тому числі пов'язаних з ергономічними ризиками (наприклад, через збільшення використання мобільних пристроїв та сидячої роботи) та когнітивного навантаження.

Задоволені розробники ефективніші та пишуть кращий код. Хороші робочі дні підвищують задоволеність роботою розробника. Розуміння того, що відрізняє хороші будні від інших днів, особливо нетипових, допоможе нам зробити гарні дні типовими. Ця робота шукає саме цього розуміння. Розуміння типових та нетипових робочих днів дозволить нам встановити базову лінію для порівняння з іншими робочими днями розробників та прийняти більш обґрунтовані рішення щодо вдосконалення процесу. Розробка - це багатоступінчастий процес зі складною взаємодією між етапами. Ці взаємодії означають, що ми не можемо розглядати кожен етап окремо, але потрібно розглядати процес як єдине ціле. Нам потрібно цілісне розуміння того, як розробники програмного забезпечення проводять свій час на роботі. Без цілісного розуміння можна подумати, що розробники, оскільки вони «розвиваються», проводять більшу частину часу на написання коду. Однак розробники витрачають на диво мало часу на кодування, від 9% до 61% залежно від дослідження. Натомість вони проводять більшу частину свого часу, збираючи інформацію, необхідну для виконання завдань розвитку, проводячи зустрічі, читаючи документацію чи веб-пошук, допомагаючи колегам та виконуючи адміністративні обов'язки. Загальноприйнята думка полягає в тому, що електронна пошта є великим джерелом відволікання та розчарування. Ми показуємо, що,



навпаки, діяльність електронною поштою мало впливає на сприйняття доброти робочого дня. Отже, зосередившись лише на одній розробковій діяльності, можна втратити важливі можливості для підвищення продуктивності.

Робота пов'язана з довгим перебуванням за комп'ютером призводить до різноманітних хвороб. Нижче наведемо приклади таких хвороб які можуть виникнути та способи їх не допущення:

Проблеми оператора ПК з опорно-руховим апаратом можна не допустити правильно організовуючи робоче місце та регулярно проводячи руханку.

Проблеми із зором. Ризик погіршити зір дуже високий. Щоб знизити імовірність таких проблем рекомендуємо працювати з великими моніторами, які знаходяться на нормальній відстані (50-90см) від очей, з нормальною яскравістю. Дуже ефективною профілактикою буде періодичне проведення розминки для очей.

Повторювані стресові травми. Використання комп'ютера може призвести до того, що м'язи будуть використовуватися якось дивно, що може спричинити підвищену скутість, біль або набряк у шії, плечах або в будь-якому місці від плечей до пальців. Щоб знизити ризик цих травм намагайтесь часто змінювати вид діяльності за ПК та потягувати м'язи рук, ніг, шії та інші.

Головні болі можуть бути наслідком перевтоми. Щоб знизити ризик утворення болю в голові, рекомендуємо слідкувати за станом повітря, та вологості. Періодично випивати водичку та підживлювати організм.

Внаслідок мало-рухливого виду діяльності та постійного підживлювання організму люди можуть зіштовхнутися з ожирінням. Щоб цього не сталося радимо періодично відвідувати спортзал, або робити пробіжки та слідкувати за збалансованістю раціону.

Стресові розлади можуть виникнути в будь, якої професії де вид діяльності є монотоним та часто людина стикається з вирішенням проблем. Для уникнення стресу рекомендуємо виділяти час для зустрічі з друзями, коханими і проведення вільного часу цікаво та різноманітно.

Травми від використання ноутбука. У наш час люди вибирають частіше використовувати ноутбуки, ніж комп'ютери, через зручність. Ноутбуки призначені для коротких періодів використання для тих, хто не має доступу до настільного комп'ютера. Але довга робота за ноутбуком може швидше призвести до усіх вище перелічених проблем. Якщо не має змоги придбати для роботи стаціонарного ПК рекомендують хоча б користуватись додатковою клавіатурою.

Штучне освітлення, яке виділяється з екранів комп'ютерів, насправді може обдурити ваш мозок і придушити його виділення мелатоніну що призведе до порушення сну. Утримайтеся від використання комп'ютера та інших гаджетів безпосередньо перед сном. Зверніться до читання книги чи чогось подібного до того, як лягати спати, тому заснути вам може легше.

Втрата слуху з навушників може виникнути через постійне та не правильне їх використання. Тому рекомендуємо не слухати постійно голосно музику чи інші звуки.

Отже, щоб уникнути ризиків пошкодити своє здоров'я загальні рекомендації це дотримуватись загальних правил користування ПК. Та вести здоровий спосіб життя, що корисно для усіх людей.

Від надзвичайних ситуацій ніхто не застрахований і вона може статися з будь ким і будь де. Якщо розробники компанії працюють не віддалено, то зазвичай це офіс де вони проводять весь день. Щоб знизити наслідки від будь якої надзвичайної ситуації рекомендуємо керівникам організувати роботу дотримуючись наступних правил:

Мати діючу та ефективну систему оповіщення

Призначити відповідальну особу за готовність до надзвичайних ситуацій.

Забезпесити наявність аварійного обладнання та аптечки.

Забезпесити наявність аварійних планів, планів евакуації та інструкцій і тому подібне.

Проводити постійні навчання персоналу щодо дій у випадку надзвичайних ситуаціях.

Якщо людина працює вдома то рекомендують мати так звану «тривожну валізу». Наявність такої валізи не дасть вам гарантій що не станеться ніякої надзвичайної ситуації, але може значно полегшити негативні наслідки від неї.

В загальному безпека праці в багатьох аспектах залежить від самого робітника. Якщо дотримуватись усіх вимог законодавства України, слідувати усім правилам та інструкціям тоді вашому здоров'ю та життю нічого не загрожує.

## ВИСНОВКИ

У кваліфікаційній роботі було розроблено інформаційну систему для забезпечення ефективності дистанційного навчального процесу з використанням мови C#. З самого початку роботи, було проведено аналіз завдання та вибір методу його рішення. У результаті аналізу предметної області та ситуації що склалася у світі було виявлено, що в сучасній освітній сфері існує потреба інформаційної системи для забезпечення ефективності дистанційного навчального процесу. Я користувався трьома сервісами дистанційного навчання: Електронне навчання від ТНТУ, Claasroom, та дистанційним навчанням створеним вчителями міста Вінниці. Двома останніми користувався в двох ролях студента і викладача. Кожен з них має як переваги так і недоліки у зв'язку з тим що заклади освіти мають індивідуальні особливості, а вчителі своє бачення щодо роботи дистанційної освіти. Тому було прийняте рішення створити власну систему, яку можна постійно вдосконалювати. Після завершення аналізу було проведено проектування структури інформаційної системи. Була розроблена логічна та фізична структури системи. Був спроектований попередній інтерфес та базу даних, враховуючи можливу подальшу модифікацію сервісу та можливість додаткового створення мобільного додатку хоча дизайн веб-сайту адаптивний. На наступному етапі був аналіз існуючих технологій та середовищ розробки подібних систем і обґрунтування їх вибору для процесу розробки. Для реалізації завдання було обрано мову програмування C# та технологію ASP.NET MVC, в якості програмного середовища розробки веб-сайту було обрано Microsoft Visual Studio 2019. В результаті було розроблено інформаційну систему у вигляді веб-сайта. На останньому етапі було проведено тестування розробки в ручному режимі у найпопулярніших браузерах. Щоб сайтом можна було користуватися потрібно придбати хостинг та викласти наш проект в мережу Internet. В подальшому робота над проектом не

завершується і сервіс дистанційного навчання буде постійно вдосконалюватися для якісного навчання студентів та учнів.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Новини Міністерства освіти і науки України: «Відсьогодні набувають чинності оновлені умови дистанційного навчання у школах»  
Опубліковано 16 жовтня 2020 року о 17:29 [Електронний ресурс] –  
Режим доступу: <https://mon.gov.ua/ua/news/>
2. ЕЛЕКТРОННЕ НАВЧАННЯ В ТНТУ. [Електронний ресурс] –Режим доступу: <https://dl.tntu.edu.ua/login.php>
3. Камер Дуглас Компьютерные сети и Internet / Дуглас Камер – М.: Вильямс, 2007. – 640 с 4. Сікорський П. І. Дидактичні поняття кредиту і модуля в контексті Болонського процесу // Шлях освіти. – 2004. – №2. – с. 19.
3. Методичні вказівки до виконання магістерської роботи освітнього рівня —магістр студентами усіх форм навчання для напряму підготовки —Інженерія програмного забезпечення// Укладачі : Петрик М.Р., МихаликД.М., Кінах Я.І., Гладько С.В., Цуприк Г.Б. – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2016 – 26 с.
4. What is Agile? – Неприбуткова організація Agile Allianceю  
[Електронний ресурс] – Режим доступу:  
<https://www.agilealliance.org/agile101/>.
5. Джамса К. Эффективный самоучитель по креативному Web – дизайну. Перевод с англ. / Крис Джамса, Конрад Кинг, Энди Андерсон – М.: ДияСофтЮП, 2005. – 672 с. 8. Кент Бек. Экстремальное программирование. — СПб : Питер, 2002. — ISBN 5-94723-032-1.
6. Розенсон И. А. Основы теории дизайна: учебник для вузов / И. А. Розенсон – СПб.: Издательство «Питер», 2010. – 219 с.

7. Адам Фрімен. ASP.NET Core MVC с примерами на C# для профессионалов // Вильямс — 2017 — 992 с
8. Top 10 Most Popular Programming Languages Північно-Східного університету [Електронний ресурс] – Режим доступу: <https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/> Дата доступу: 07.12.2020
9. Офіційна документація Microsoft з ASP.Net Core: Introduction to ASP.NET Core — Режим доступу: <https://docs.microsoft.com/en-us/aspnet/core/>. Дата доступу: 07.12.2020
10. Top 10 Websites Written Using ASP.NET MVC – Режим доступу: <https://dzone.com/articles/top-10-websites-written-using-aspnet-mvc>
11. «Visual Studio» [Електронний ресурс] – Режим: <https://visualstudio.microsoft.com/ru/>
12. Date, Chris. Database in depth : relational theory for practitioners. O'Reilly. ISBN 0-596-10012-4..
13. Офіційна сторінка MySQL. [Електронний ресурс] – Режим доступу: <https://dev.mysql.com/doc/relnotes/mysql/8.0/en/> .
14. Офіційна сторінка Microsoft SQL Server. [Електронний ресурс] – Режим доступу: <https://www.microsoft.com/en-us/sql-server/default.aspx> .
15. Беллиньясо М. Разработка Web-приложений в среде ASP .NET 2.0. . Перевод с англ. / М. Беллиньясо – М.: Вильямс, 2007. – 640 с. 22. Макдональд М. Microsoft ASP .NET 3.5 с примерами на C# 2008 для профессионалов, 2–е изд.: Перевод с англ. / Мэтью Макдональд, Марио Шпуста – М.: Вильямс, 2008. – 1424 с
16. F.J. Heemstra. Software cost estimation. Information Software Technology, 1992, vol. 34, no. 10, pp. 627-639
17. Brooks, Fred (1986). Kugler, H.J. (ed.). No Silver Bullet Essence and

Accidents of Software Engineering. ІНФормation Processing ' Elsevier Science Publishers B.V (North-Holland). ISBN 0-444-70077-3.

18. Методичні вказівки для виконання розділу дипломної роботи щодо техніко-економічного обґрунтування вибору проектного рішення розробки та оцінки якості програмного забезпечення / Упор. Петрик М.Р., Кінах Я.І., Головатий А.І., Рогатинська Л.Р. – Тернопіль: Вид-во ТНТУ ім. І.Пулюя. – 2013. – 34 с.

19. Санітарні норми мікроклімату виробничих приміщень ДСН

3.3.6.042-99. [Електронний ресурс]. – Режим доступу:

<https://zakon.rada.gov.ua/rada/show/va042282-99> .

20. Державні санітарні правила і норми роботи з візуальними

дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98. [Електронний ресурс]. – Режим доступу:

<https://zakon.rada.gov.ua/rada/show/v0007282-98> .

21. Наказ Міністерство охорони здоров'я України «Про затвердження Порядку проведення медичних оглядів працівників певних категорій».

[Електронний ресурс]. – Режим доступу:

<https://zakon.rada.gov.ua/laws/show/z0846-07> .

22. Why Is C# Among The Most Popular Programming Languages in The World? – Онлайн журнал Medium. [Електронний ресурс] – Режим доступу: <https://medium.com/sololearn/why-is-c-among-the-most-popular-programminglanguages-in-the-world-ccf26824ffcb> . Дата доступу: 07.12.2020

23. Про затвердження Порядку проведення медичних оглядів працівників певних категорій [Електронний ресурс] – Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0846-07#Text>



24. Форма №2 “Звіт про фінансові результати”: методика підготовки.  
[Електронний ресурс]. – Режим доступу:  
<http://osvita.ua/vnz/reports/accountant/17368/> .
25. L.C. Briand, I. Wiecezorek. Software Resource Estimation. Encyclopedia of Software Engineering. Volume 2, P-Z (2nd ed., 2002), Marciniak, John J. (ed.) New York: John Wiley & Sons, pp. 1160-1196.

## **ДОДАТОК А**











## ДОДАТОК Б

### Код класу Startup

```
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Disted.Domain;
using Disted.Domain.Repositories.Abstract;
using Disted.Domain.Repositories.EntityFramework;
using Disted.Service;
Код класу Startup
namespace Disted
{
    public class Startup
    {
        public IConfiguration Configuration { get; }

        public Startup(IConfiguration configuration) =>
        Configuration = configuration;

        public void ConfigureServices(IServiceCollection
        services)
        {
            //Підключаємо конфігурацію
            Configuration.Bind("Project", new CoHfig());
            //підключаємо функціонал
            services.AddTransient<ITextFieldsRepository,
            EFTextFieldsRepository>();
            services.AddTransient<IServiceItemsRepository,
            EFServiceItemsRepository>();
            services.AddTransient<DataManager>();
            //підключаємо контекст БД
            services.AddDbContext<AppDbContext>(x =>
            x.UseSqlServer(Configuration.ConnectionString));
        }
    }
}
```



```

        //налаштовуємо identity систему
        services.AddIdentity<IdentityUser,
IdentityRole>(opts =>
            {
                opts.User.RequireUniqueEmail = true;
                opts.Password.RequiredLength = 6;
                opts.Password.RequireNonAlphanumeric =
false;

                opts.Password.RequireLowercase = false;
                opts.Password.RequireUppercase = false;
                opts.Password.RequireDigit = false;

            }).AddEntityFrameworkStores<AppDbContext>().AddDefaultToken
Providers();

        //налаштовуємо authentication cookie
        services.ConfigureApplicationCookie(options =>
            {
                options.Cookie.Name = "myCompanyAuth";
                options.Cookie.HttpOnly = true;
                options.LoginPath = "/account/login";
                options.AccessDeniedPath =
"/account/accessdenied";
                options.SlidingExpiration = true;
            });

        //налаштовуємо політику авторизації для Admin
area
        services.AddAuthorization(x =>
            {
                x.AddPolicy("AdminArea", policy => {
policy.RequireRole("admin"); });
            });

        //Додаємо підтримку контролерів і подань (MVC)
        services.AddControllersWithViews()

        // Сумісність. asp.net.core 3.0

```

```

        .SetCompatibilityVersion(CompatibilityVersion.Version_3_0).
        AddSessionStateTempDataProvider();

    }

    // This method gets called by the runtime. Use this
    method to configure the HTTP request pipeline.

    public void Configure(IApplicationBuilder app,
        IWebHostEnvironment env)
    { // Помилки

        if (env.IsDevelopment())
        {

            app.UseDeveloperExceptionPage();

        }

        app.UseRouting();
        // підтримка статичних файлів
        app.UseStaticFiles();
        //підключаємо систему маршрутизації
        app.UseRouting();

        //підключаємо аутентифікацію и авторизацію
        app.UseCookiePolicy();
        app.UseAuthentication();
        app.UseAuthorization();
        // реєструємо маршрути
        app.UseEndpoints(endpoints =>
        {

            endpoints.MapControllerRoute("default",
                "{controller=Home}/{action=Index}/{id?}");

        });

    }

}

```

## ДОДАТОК В

### Код Шаблону документа Layout

```
<!DOCTYPE HTML>

<html>

<head>

    @await Html.PartialAsync("MetatagsPartial")

    @await Html.PartialAsync("CssPartial")

</head>

<body>

    <div id="page-wrapper">

        @await Html.PartialAsync("HeaderPartial")

        <div id="main">

            <div class="container">

                <div class="row main-row">

                    <div class="col-8 col-12-medium">

                        @RenderBody()

                    </div>

                </div>

                @await
Html.PartialAsync("SidebarPartial")

            </div>

        </div>

        @await Html.PartialAsync("FooterPartial")

    </div>

    @await Html.PartialAsync("ScriptsPartial")

</body>

</html>
```

## **ДОДАТОК Г**

Диск з програмою