

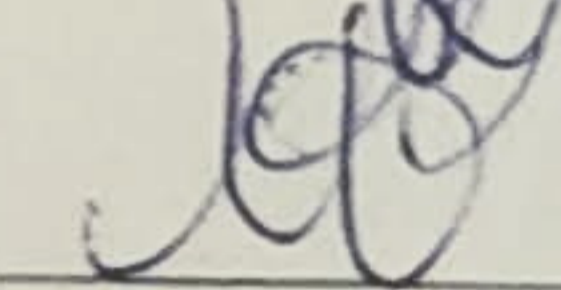
**КВАЛІФІКАЦІЙНА РОБОТА  
БАКАЛАВРА**

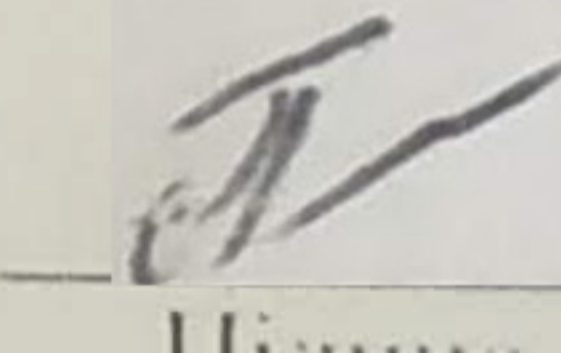
на тему: Інформаційна система для забезпечення дистанційної форми освіти  
учнів загальноосвітніх навчальних закладів

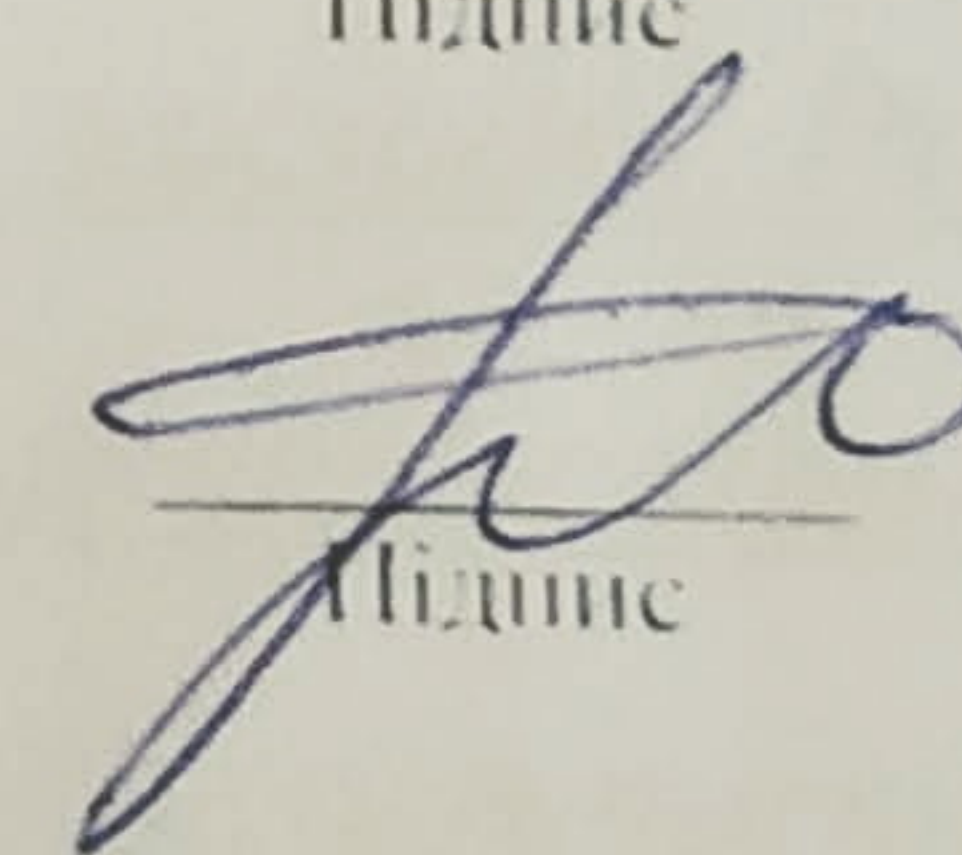
Галузь знань 12 – Інформаційні технології  
Шифр і назва галузі знань

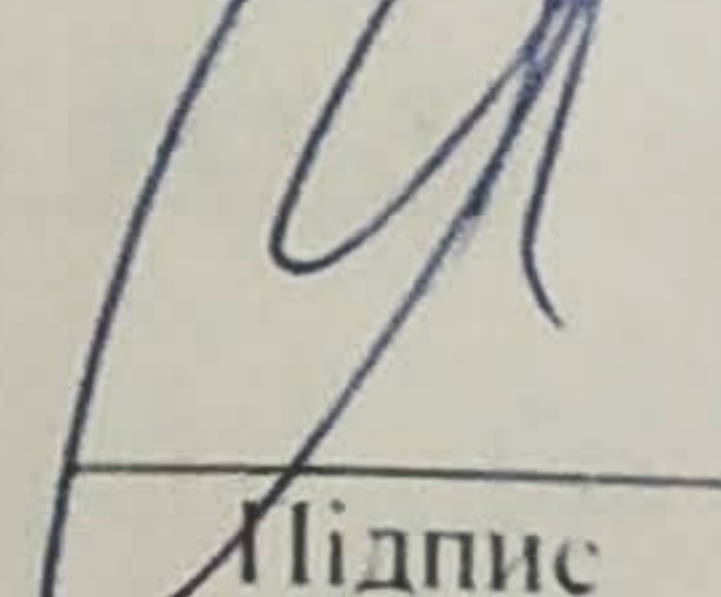
Спеціальність 122 – Комп'ютерні науки  
Шифр і назва спеціальності

Освітня програма Комп'ютерні науки  
Назва освітньої програми

Виконав: студент 4 курсу, група КН-17-1  А.С. Малайко  
Курс, група виконавця Підпис Ініціали, прізвище

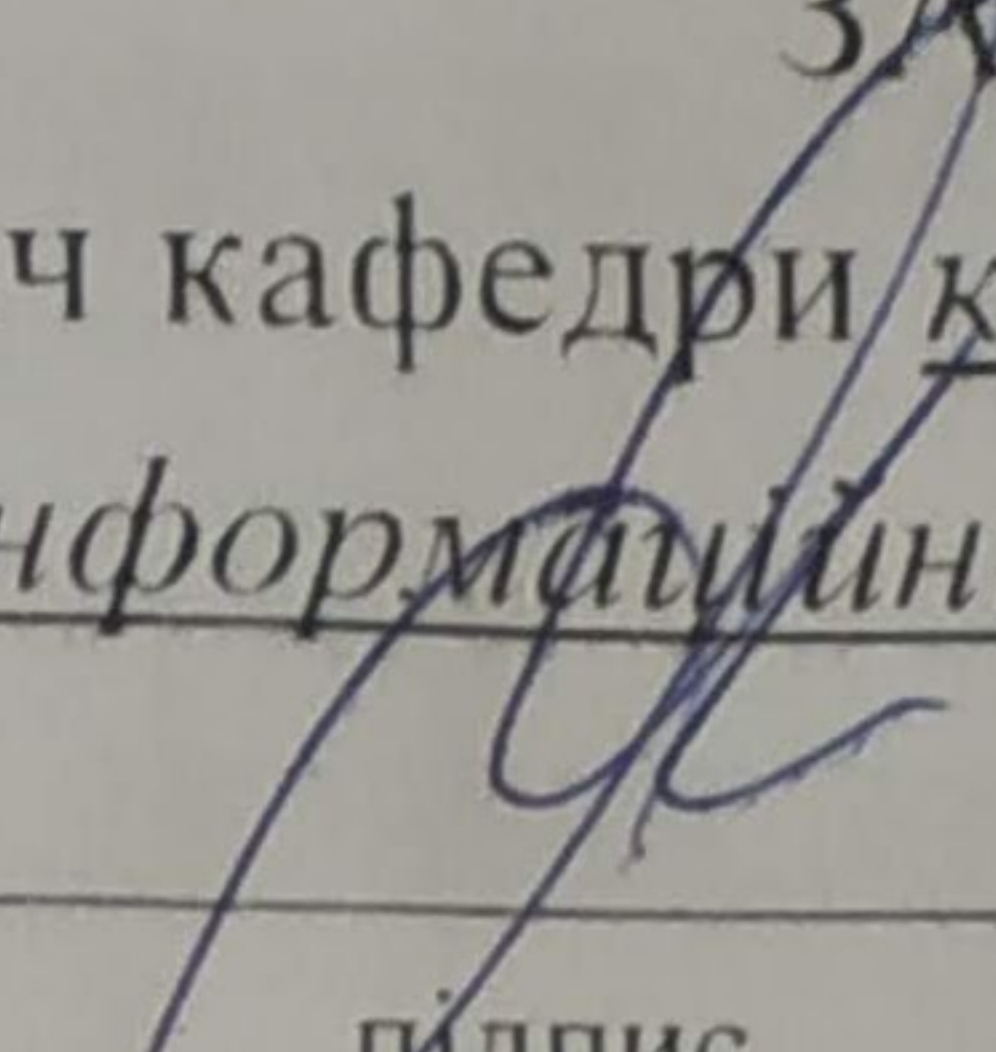
Керівник: к.п.н., доцент кафедри КНІТ  С.С. Петровський  
Науковий ступінь, посада Підпис Ініціали, прізвище

Нормоконтроль: к.т.н., доцент кафедри КНІТ  Р.О. Багрій  
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:  
Зав. кафедри КНІТ, д.т.н, професор  О.В.Бармак  
Підпис Ініціали, прізвище

08 червня 2021 р.

ЗАТВЕРДЖУЮ  
Завідувач кафедри комп'ютерних наук та інформаційних технологій

  
підпис

д.т.н. професор О.В. Бармак

«08» лютого 2021 року

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

- Тема кваліфікаційної роботи бакалавра: «Інформаційна система для забезпечення дистанційної форми освіти учнів загальноосвітніх навчальних закладів»
- Завдання видано студенту Малайку Андрію Сергійовичу  
Прізвище, ім'я, по батькові
- Керівник роботи доцент кафедри КНІТ Петровський Сергій Степанович  
Посада, прізвище, ім'я, по батькові
- Затверджено наказом університету від «05» 02 2021 р. № 11
- Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи – розробка інформаційної системи, що забезпечує дистанційну форму освіти учням загальноосвітніх навчальних закладів. Розроблена система повинна бути наділена функціоналом створювання освітнього матеріалу у форматі електронних підручників, тестів та їх результатів у вигляді оцінок. Також у системі повинен бути присутній профіль користувача задля надання особистих даних компонентам системи, які цього вимагають.

Виконавець:

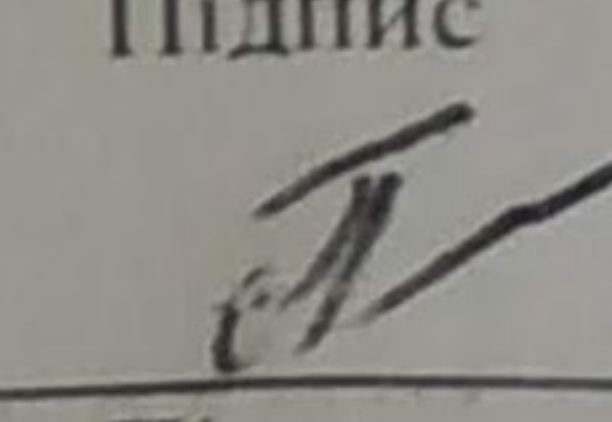
студент 4 курсу, група КН-17-1  
Курс, група виконавця

  
Підпис

А.С. Малайко  
Ініціали, прізвище

Керівник:

к.п.н., доцент кафедри КНІТ  
Науковий ступінь, посада

  
Підпис

С.С. Петровський  
Ініціали, прізвище

## Анотація

Тема кваліфікаційної роботи бакалавра: «Інформаційна система для забезпечення дистанційної форми освіти учнів загальноосвітніх навчальних закладів»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-17-1 Малайко Андрій Сергійович

Керівник кваліфікаційної роботи бакалавра: к.п.н доцент кафедри КНІТ Петровський Сергій Степанович

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
65	42	8	27	3

Метою кваліфікаційної роботи бакалавра є розробка інформаційної системи, що здатна забезпечити дистанційну форму навчання учням загальноосвітніх навчальних закладів. Під час процесу розробки використовувались такі технології як Angular, Node.js, Express.js, система керування базами даних MongoDB та мова програмування JavaScript.

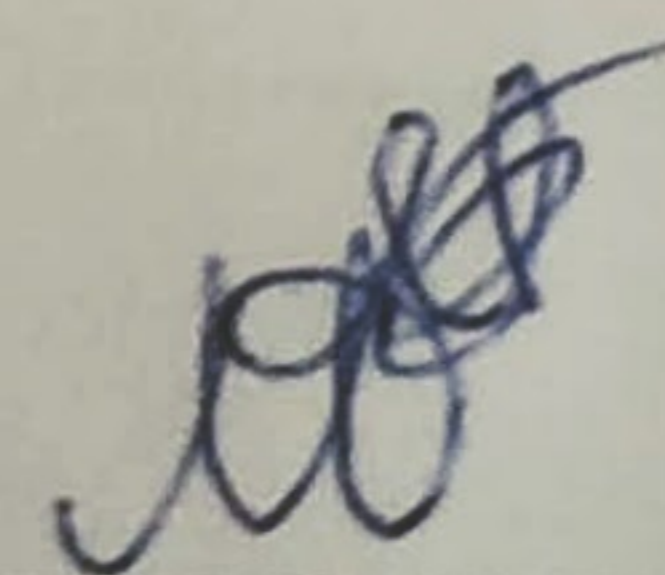
Розроблена система призначена для рішення проблем, що пов'язані з відсутністю очної форми отримання загальної середньої освіти дітьми шкільного віку. Створений вебзастосунок надає змогу учням, отримувати нові знання читаючи електронні підручники, а також перевіряти засвоєний матеріал шляхом проходження відповідних тестів з відображенням результатів.

Ключові слова: адміністратор, учень, електронний підручник, тест, оцінка.

Виконавець:

студент 4 курсу, група КН-17-1

Курс, група виконавця



Підпис

А.С. Малайко

Ініціали, прізвище

## Зміст

Перелік скорочень.....	2
Вступ.....	3
1 Характеристика предметної області та постановка задачі .....	4
1.1 Аналіз предметної області .....	4
1.2 Аналіз існуючого програмного забезпечення предметної області.....	7
1.3 Аналіз сучасних засобів створення програмного забезпечення.....	18
1.4 Постановка задачі і вимоги до розробки інформаційної системи .....	24
2 Проектування структури інформаційної системи .....	25
2.1 Функціональна структура і бізнес-процеси системи.....	25
2.2 Інформаційна структура системи .....	28
2.3 Вибір засобів розробки інформаційної системи.....	35
2.3.1 Вибір мови програмування .....	35
2.3.2 Вибір серверної платформи .....	36
2.3.3 Вибір системи керування базами даних.....	38
2.3.4 Вибір редактору програмного коду.....	41
3 Програмна реалізація інформаційної системи .....	43
3.1 Структура та функціональне призначення складових системи.....	43
3.2 Особливості реалізації складових системи.....	46
3.3 Тестування інформаційної системи.....	50
3.4 Інструкція користувача .....	53
3.5 Вимоги до розгортання інформаційної системи.....	62
Висновки.....	63
Перелік посилань .....	64
Додатки	

## Перелік скорочень

Скорочення, термін, позначення	Пояснення
ПЗ	Програмне забезпечення
ПП	Програмний продукт
ЕОМ	Електронно-обчислювальна машина
ІТ	Інформаційні технології
ІДЗ	Індивідуальне домашнє завдання
ІС	Інформаційна система
НВК	Навчально-виховний комплекс
НЗ	Навчальний заклад
ПІБ	Прізвище, ім'я, по батькові
RoR	Ruby on Rails
CRUD	Create Read Update Delete
ЗНЗ	Загальноосвітні навчальні заклади
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	hypertext Preprocessor
ПКМ	Права кнопка миші
ЛКМ	Ліва кнопка миші
RAM	Random Access Memory

## Вступ

У теперішній час можливість дистанційно працювати чи навчатись з кожним днем набирає оберти популярності. Така ситуація склалася не лише в нашій країні, а в світі загалом. Незважаючи на цей факт, складно висловувати протиріччя на рахунок того, що дистанційна форма роботи необхідна для введення у повсякденне життя людини. Вона може бути застосована або незастосована в залежності від ситуації того чи іншого суб'єкта, але її присутність як варіанту вибору необхідна. Це стосується багатьох сфер діяльності, але найбільш прямолінійний та важливий зв'язок має сфера освіти. Якщо виникає ситуація при якій учень школи, або студент не має змоги відвідувати навчально-виховний комплекс, технікум, університет чи інститут, у нього повинна бути змога здобувати знання за межами учбового закладу, за допомогою дистанційної форми навчання.

Для вирішення даної проблеми можна скористатися існуючим, допоміжним інструментом – інформаційними технологіями. У даний час інформаційні технології відіграють важливу роль в нашому житті. Експлуатація ЕОМ стала повсякденною справою, принаймні ще зовсім нещодавно робоче місце, споряджене комп'ютером, було великою рідкістю. Інформаційні технології надали нові можливості для роботи й відпочинку, у великій мірі полегшили працю і спосіб життя кожної модерної людини. Сьогоднішнє суспільство навряд чи можна уявити без інформаційних технологій. Інформаційні технології миттєво трансформувалися на життєво важливий мотив розвитку не тільки світової економіки, а й інших сфер людської діяльності. На сьогодні практично неможливо знайти сферу, в якій зараз не застосовуються інформаційні технології, і сфера освіти не є виключенням [1].

ІТ є саме тим рішенням, яке може надати студенту навчального закладу чи учню навчально-виховного комплексу можливість здобувати освіту дистанційно за допомогою певного програмного продукту. Характерним прикладом є вебсайт, що містить навчальну інформацію у вигляді електронних книг та іншого матеріалу. Щоб реалізувати подібне до описаного у прикладі, потрібно задіяти таку складову ІТ, як програмування, та широкий спектр її підрозділів.

## **1 Характеристика предметної області та постановка задачі**

### **1.1 Аналіз предметної області**

Освіта сьогодні відіграє дуже впливову роль у житті кожної людини, вона є невід'ємною частиною майбутнього кваліфікованого працівника, чи просто людини, яка прагне стати успішною. Освіченість є важливою, оскільки пришвидшує процес розвитку й становлення людини як особистості, індивідуума, забезпечує формування її світогляду, духовності і моральних принципів. Мета її полягає у допомозі індивідууму увійти в культуру, освоїти її цінності, успішно діяти в культурному середовищі. Її ціллю є формування суб'єкта, здатного зустріти проблеми та суперечності цього світу, розділити його на предметні складові, відшукати комбінації існуючих рішень і своєю діяльністю згенерувати відсутні, для того, щоб творча сторона породила нову культурну реальність, а також діяльність зі створення культурних багатств. Освіта є найбільш ефективним напрямком входження людини в інтелектуальний та культурний всесвіт, тому що саме в процесі здобування освіти людина засвоює як життєві так і культурні цінності [2].

Загальна освіта є відображенням комплексу базових знань про людину та всесвіт в якому вона існує. За класифікацією загальна освіта буває початковою, основною та середньою. Перші дві складові містять у собі фундаментальні знання, що стосуються читання та письма. А загальна середня освіта, являє собою планомірний процес опанування систематизованих знань суб'єктом про природне середовище, людську діяльність, культуру, суспільство за допомогою відповідних, пізнавальних методів. Також участь в опануванні знань бере практична діяльність, разом вони призводять до отримання інтелектуального, соціального й фізичного розвитку індивідууму в якості результату. Підґрунтям такої освіти можна назвати навчальний та виховний процеси, які базуються не лише на принципах науковості, але й на загальнолюдських цінностях. Вони поєднують навчання і виховання і сприяють формуванню цілісної особистості [3].

На теперішній час в Україні загальна середня освіта репрезентована різними загальноосвітніми закладами, що відрізняються типом та формою власності. До окремих складових такої системи можна віднести:

- позашкільні заклади;
- навчально-виробничі комбінати;
- вищі та професійно-технічні навчальні заклади I-II рівнів акредитації.

Отримати таку освіту можна в відповідному загальноосвітньому закладі, який має ціль забезпечити реалізацію прав громадянина на отримання загальної середньої освіти. У випадку, коли загальноосвітній навчальний заклад відноситься до приватної форми власності, його активність повина бути визнана офіційною ліцензією. Процес видачі такої ліцензії проходить відповідно до правил, що були встановлені українським законодавством. Коли загальноосвітній навчальний заклад проводить інноваційну роботу, то йому привласнюється статус експериментального. До загальноосвітніх навчальних закладів відносяться гімназії, школи I-III ступенів, гімназії-інтернати, ліцеї, колегіуми, школи-інтернати I-III ступенів, ліцеї-інтернати, школи соціальної реабілітації, вечірні школи II-III ступенів, а також навчально-реабілітаційні центри, призначені для надання освіти дітям з особливими вимогами та вадами розвитку [4].

Прийомом до загальноосвітніх навчальних закладів займається державна політика, яка націлена на залучення всіх громадян у шестирічному віці до отримання початкової освіти в загальноосвітніх навчальних закладах. Дана діяльність проводиться згідно до права, що гарантується Конституцією України кожному громадянину щодо доступності і безоплатності здобуття повної загальної середньої освіти. Державна політика працює з ціллю створення умов, які приводять до впровадження обов'язкової повної загальної середньої освіти, здобутті учнями належного освітньо-культурного рівня, що забезпечується розвиненою мережею загальноосвітніх навчальних закладів. Державна політика у сфері освіти націлена не лише на здобуття учнями знань чи навичок, а розвитку на їх підставі ключових компетентних якостей, які сприятимуть умінню через спілкування віднаходити

узгодженність з іншими суб'єктами, організовувати власне життя й бути корисним сім'ї та суспільству [5].

Незважаючи на добре розвинуте стаціонарне здобуття освіти, незавжди є змога здобувати освіту за денною формою навчання. Оскільки, у житті кожної людини можуть виникати ситуації, коли потрібно регулярно змінювати місце проживання, або бути у постійних переїздах. А якщо це стосується дитини, то у такому випадку процес здобуття освіти буде складним та болючим. Причини переїзду можуть бути різні, але це не змінює потребу вчасного здобуття якісної освіти. За рішенням даної проблеми можна звернутись до області інформаційних технологій, яка може надати спеціально відведений для цього ряд інструментів. Правильно підібраний інструмент може впровадити рішення даної проблеми шляхом надання дистанційної освіти та автоматизації певних складових процесу її здобуття. Автоматизація може бути задіяна до таких речей як отримання навчальних посібників, зручний контакт з вчителями, також здання ІДЗ на перевірку та отримання оцінок.

Отже, в результаті аналізу предметної області було виявлено ряд основних параметрів, їх перелік наведений у таблиці 1.1.

Таблиця 1.1 – Параметри предметної області

<b>Параметр</b>	<b>Опис</b>
Учні	люди, які прагнуть отримати освіту
Предмети	Різноманітні наукові напрямки
Підручники	Книги, які містять освітню інформацію
Теми	Послідовні етапи вмісту підручників
Оцінки	Результат перевірки засвоєного матеріалу

Після визначення основних параметрів предметної області, з'являється бачення сутностей, які можуть бути застосованими у використанні інформаційної системи. Також це надає змогу сформулювати їх ключові властивості на даному етапі.

## 1.2 Аналіз існуючого програмного забезпечення предметної області

Класифікувати існуючі програмні продукти, що належним чином відносяться до сфери освіти можна за загальними принципами класифікації програмного забезпечення. Загальна класифікація програм містить такі напрямки як веб, програми для робочого столу та мобільні програми. Впровадження дистанційної форми навчання в учбові заклади, у більшості випадках представляється вебсайтами, оскільки вони є більш доступними в плані реалізації, затрат та наданні доступу для користування. Саме даний напрямок є найбільш актуальним для автоматизації певних освітніх процесів. Причиною цього є зручна експлуатація згенерованого за допомогою ІТ функціоналу, для прикладу це може бути отримання доступу до освітнього матеріалу, перевірка знань, отримання результатів перевірки засвоєного матеріалу та інше.

Для отримання детального розуміння як саме може бути реалізована дистанційна форма навчання, були розглянуті приклади існуючих веб-сайтів. Одним із таких прикладів є сайт школи «Optima» (рисунок 1.1) [6].

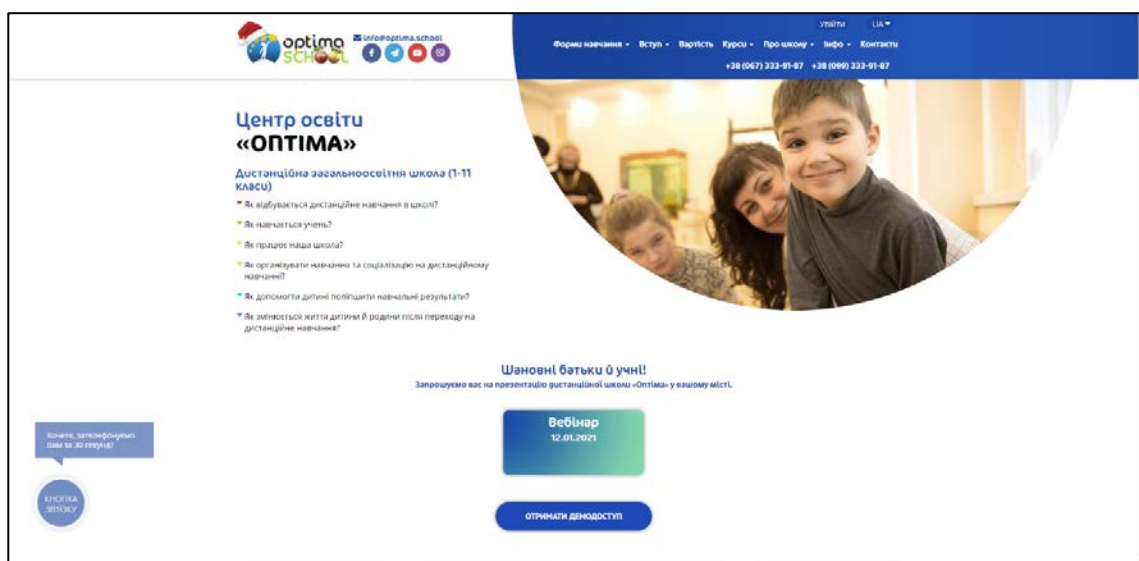


Рисунок 1.1 – Початкова сторінка школи «Optima»

Після здійснення переходу за відповідним посиланням, користувач потрапить на початкову сторінку сайту школи «Optima». Далі в нього з'явиться можливість

ознайомитися з інформацією про ресурс та отримати відповіді на питання, що виникають. Роль гіда для користувача виконує шапка сайту, що містить внутрішні посилання на конкретні сторінки (рисунки 1.2).

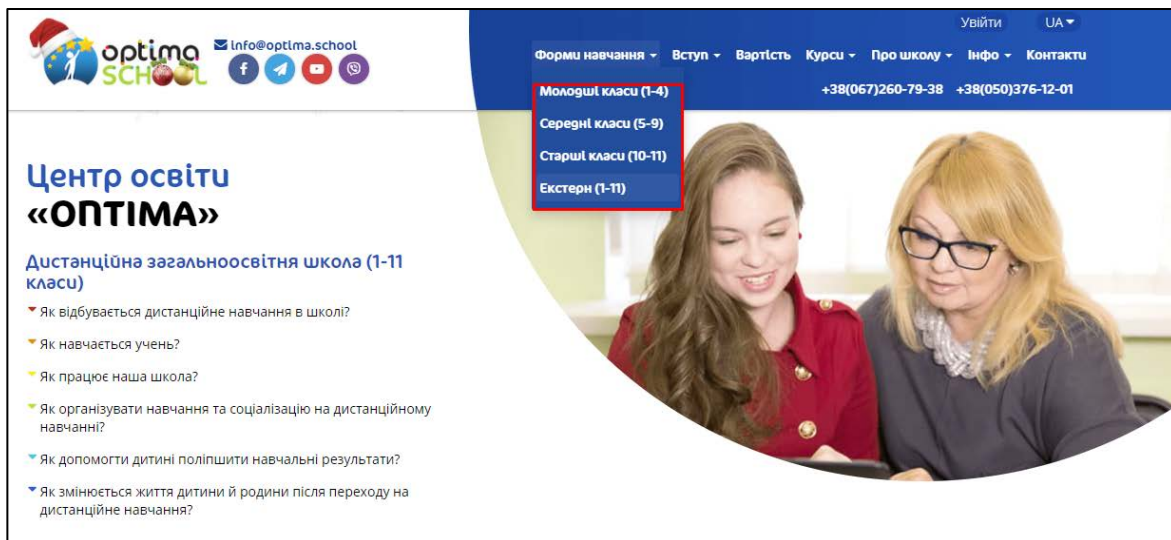


Рисунок 1.2 – Вибір заголовка

Якщо користувач натиснув на елемент випадаючого меню, його автоматично переадресує на відповідну сторінку (рисунки 1.3).

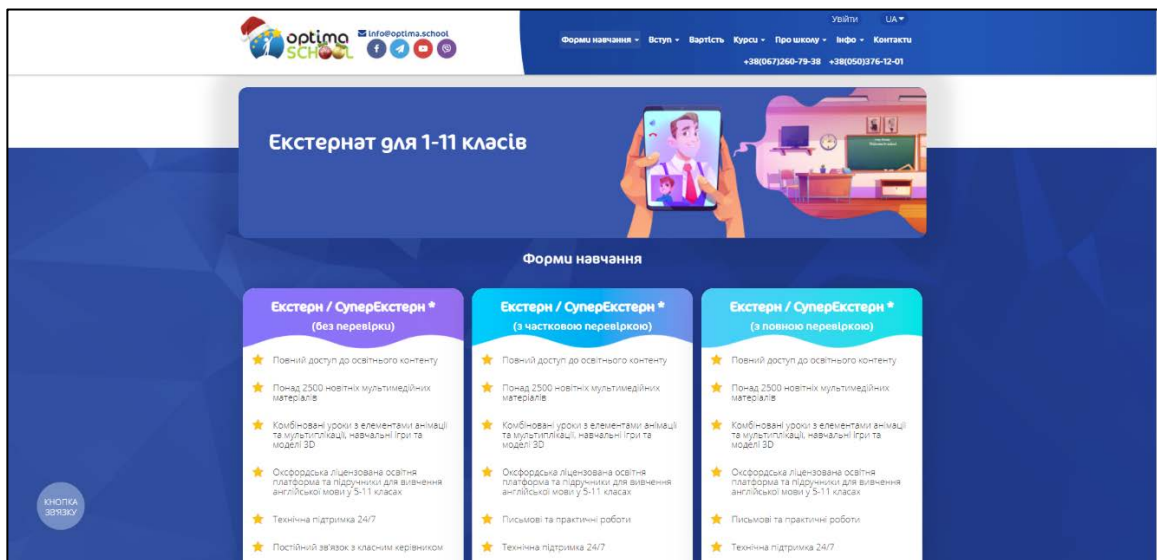


Рисунок 1.3 – Перехід на іншу сторінку

В освітньому ресурсі – «Optima» також є можливість отримати демо-доступ до навчального сайту, що являється інформаційної системою дистанційного навчання для користувачів даного сервісу (рисунок 1.4).

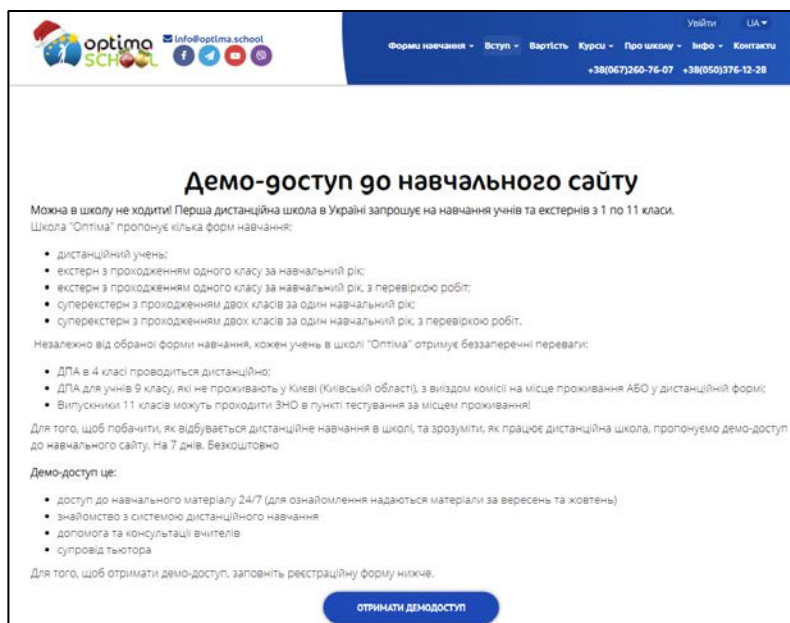


Рисунок 1.4 – Інформаційна сторінка демо-доступу

Для отримання можливості навчатись дистанційно в ІС «Optima» потрібно натиснути на кнопку «Отримати демодоступ» після чого відбудеться переадресація на сторінку реєстрації (рисунок 1.5).

**Отримати демо-доступ**

Для отримання безкоштовного демо-доступу до електронної системи навчання, заповніть будь ласка форму реєстрації нижче. Демо-доступ буде надано на 7 днів з моменту реєстрації. Всі поля обов'язкові для заповнення.

**Дані учня**

Прізвище учня:  Ім'я учня:

По-батькові учня:  Клас:

Електронна пошта:  Номер телефону для зв'язку:

**Дані одного з батьків**

Тато:  Ім'я одного з батьків:

Прізвище одного з батьків:  Ім'я одного з батьків:

По-батькові одного з батьків:

**Отримати демо-доступ**

Рисунок 1.5 – Реєстрація на ресурсі «optima.school»

На даній сторінці необхідно ввести особисті дані учня та одного з його батьків на вибір. До особистих даних учня відноситься:

- прізвище;
- ім'я;
- по-батькові;
- клас;
- загальна діюча електронна адреса;
- номер телефону.

В особисті дані одного з батьків входить:

- ім'я;
- прізвище;
- по-батькові.

Всі поля є обов'язковими для введення і після надання всієї інформації з'являється можливість натиснути на кнопку «Отримати демо-доступ». Наступне, що побачить користувач це впливаюче вікно з інформацією про прийняття заявки, посилання на ресурс дистанційного навчання, а також дані для входу в систему, що складаються з логіну та паролю (рисунок 1.6).

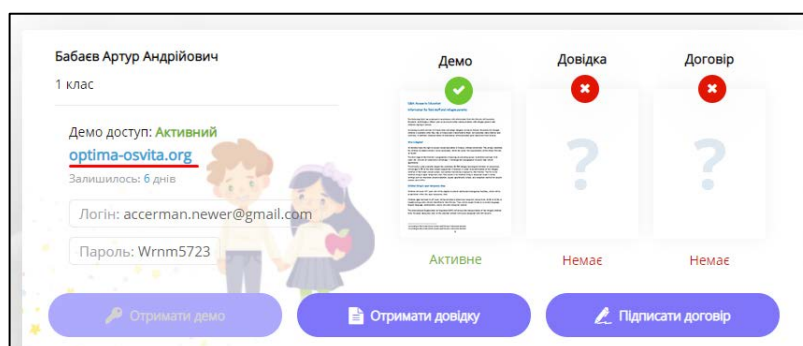


Рисунок 1.6 – Впливаюче вікно з даними для авторизації

Наступною дією для отримання доступу до ІС є перехід на відповідну сторінку авторизації на ресурсі «optima-osvita.org», після переходу необхідно ввести вже існуючі логін та пароль і натиснути на кнопку «Вхід» (рисунок 1.7).

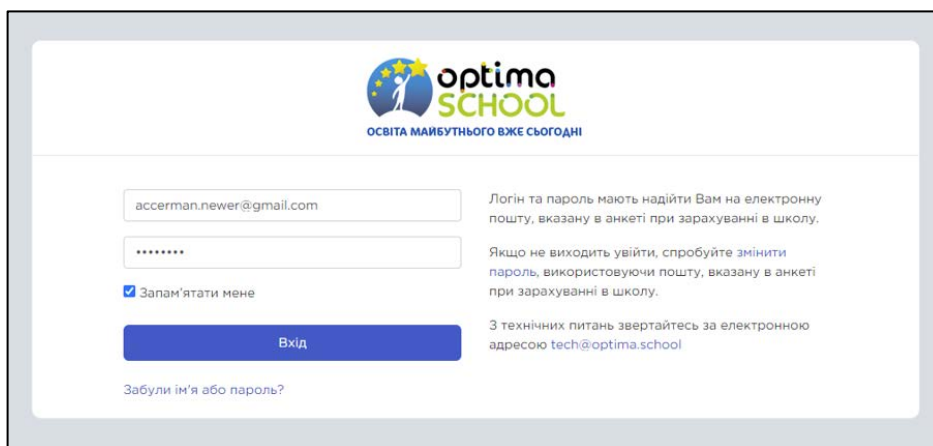


Рисунок 1.7 – Авторизація до ІС «Optima»

Після виконання всіх, вище описаних дій користувач опиниться на початковій сторінці «optima SCHOOL» з можливістю подальшого ознайомлення з ІС (рисунок 1.8).

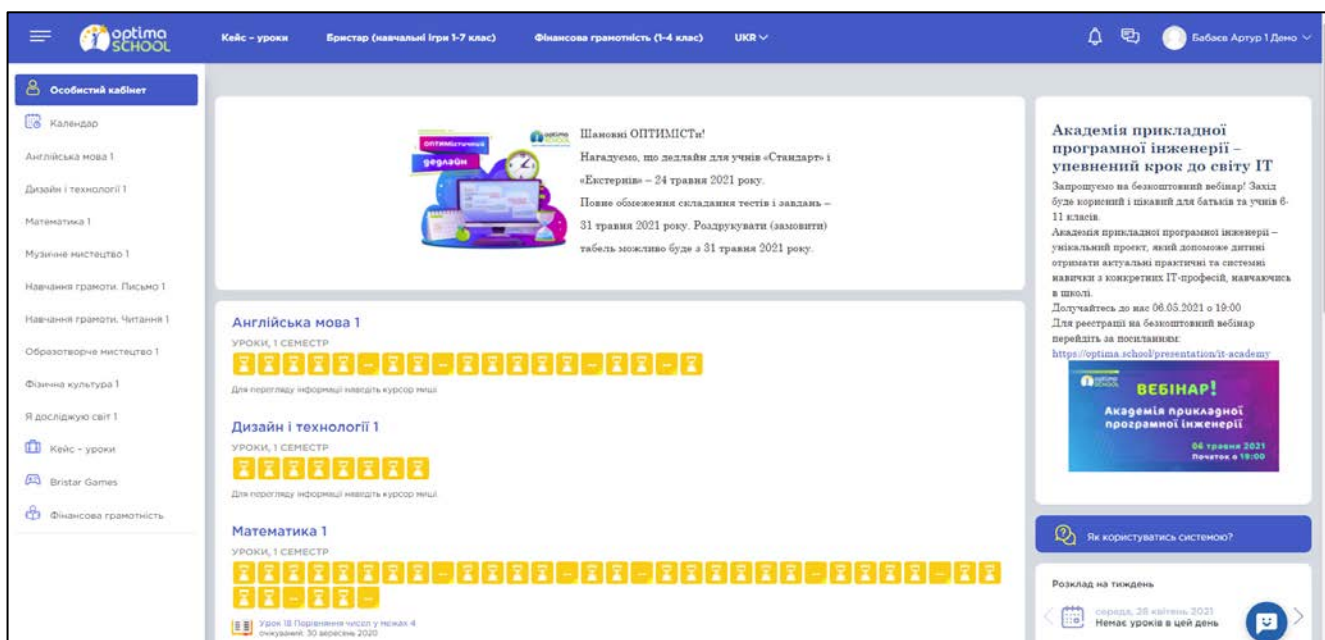


Рисунок 1.8 – Початкова сторінка ІС дистанційного навчання

Під час першого візиту до сайту «optima SCHOOL» складається приємне враження про ресурс за причиною присутності навігаційних панелей, пам'яток, меню, контактів, сповіщення та іншого. Також початкова сторінка містить список предметів відповідно до класу зареєстрованого учня, в даному випадку базові предмети першого

класу, до яких відноситься математика, англійська мова, навчання грамоти, образотворче мистецтво, фізична культура, музичне мистецтво та інші. Також існують підказки як користуватися системою для менш досвідчених користувачів подібних ресурсів, що можна рахувати як дуже значимий плюс. Всі перераховані раніше інформаційні елементи свідчать про розумне проектування ІС, ціль якого полягала у наданні будь-якої потрібної для користувача інформації стосовно ресурсу. Сайт не містить помилок стосовно механіки, на кшталт невідповідності вмісту тої чи іншої сторінки при здійсненні на неї переходу, кожен елемент містить ту інформацію для якої був створений, теж саме стосується модулів, які виконують ті функції для яких вони були призначені. Для отримання більшої кількості інформації стосовно даної системи було здійснено перехід на сторінку предмету англійської мови (рисунок 1.9).

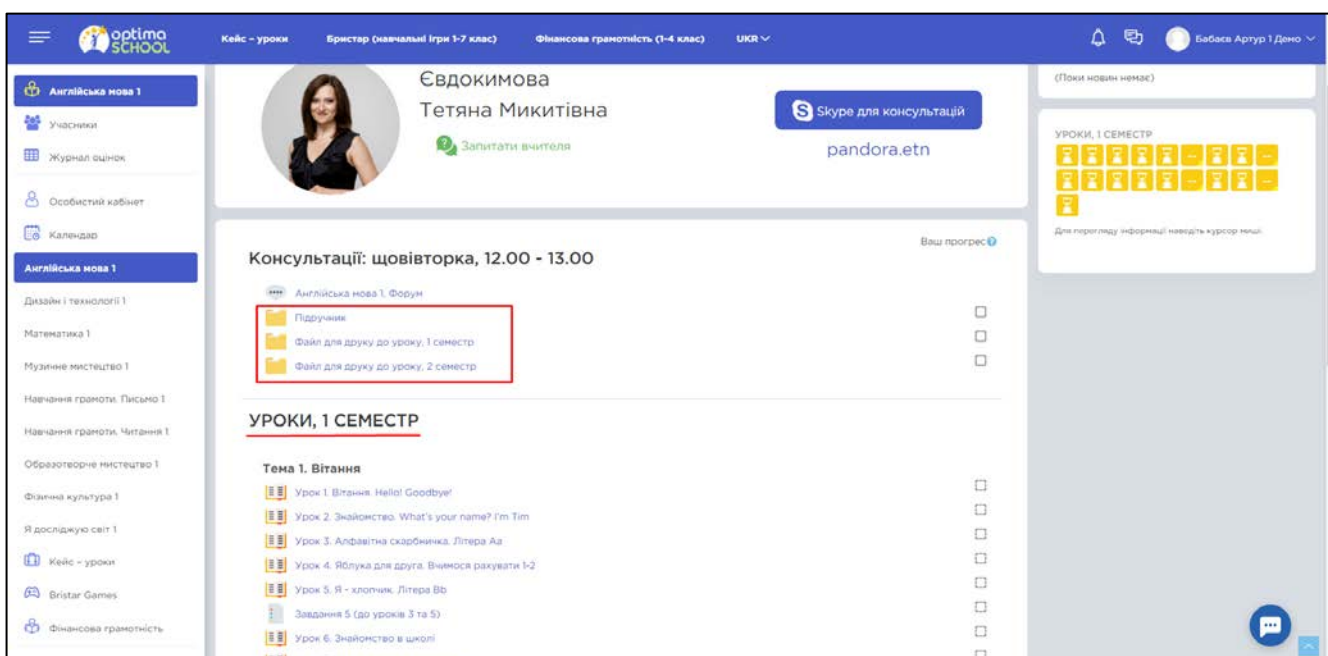


Рисунок 1.9 – Інформаційна сторінка предмету з освітніми файлами

Відповідно до рисунку 1.9 сторінка предмету містить теми та їх уроки, також освітня програма ділиться на семестри, що є аналогічним до стаціонарних шкіл. Також на сторінці предмету присутній контакт для зв'язку з вчителем, файли що

містять освітній матеріал, форум на якому можна отримати відповідні на задані питання, і підказки щодо прогресу користувача.

Підсумовуючи розглянутий приклад можна сказати, що він був побудований з використанням найсучасніших технологій, спроектований у тому напрямку, який надасть користувачеві все необхідне для отримання освіти дистанційно. Він не містить слабких сторін і його можна вважати гідним для орієнтиру побудови власної інформаційної системи.

Наступним прикладом є вебсайт навчально-виховного комплексу №2 міста Хмельницького [7]. При першому візиті до даного ресурсу можна отримати детальну інформацію про заклад, а саме його історію, місце розташування, актуальні новини, оголошення, контакти та інше (рисунок 1.10).



Рисунок 1.10 – Початкова сторінка вебсайту НВК№2

У користувача даної інформаційної системи також є можливість отримати додаткову інформацію стосовно початкової, основної та старшої школи, за допомогою бічної навігаційної панелі (рисунок 1.11). В цілому даний вебсайт надає всю необхідну початкову інформацію, а також містить посилання на джерело дистанційного навчання.

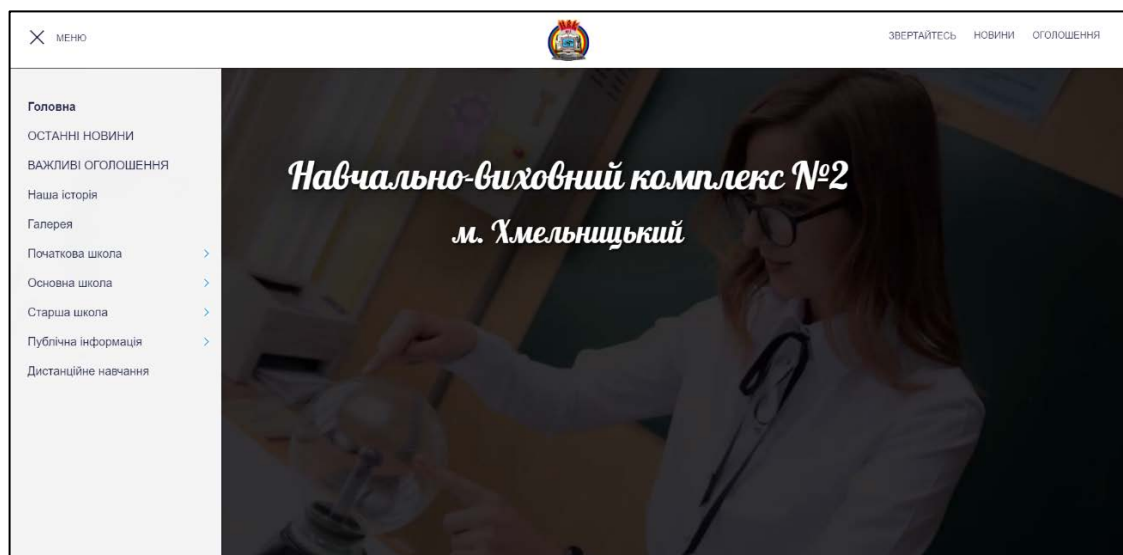


Рисунок 1.11 – Бічне навігаційне меню з додатковою інформацією про заклад

Якщо користувач хоче дізнатись як саме впроваджене дистанційне навчання в НВК №2, він може натиснути на елемент «Дистанційне навчання» бічного меню. Після виконання вище описаної дії користувача перенаправить на інший ресурс, що надає можливість дистанційно отримувати освіту (рисунок 1.12).

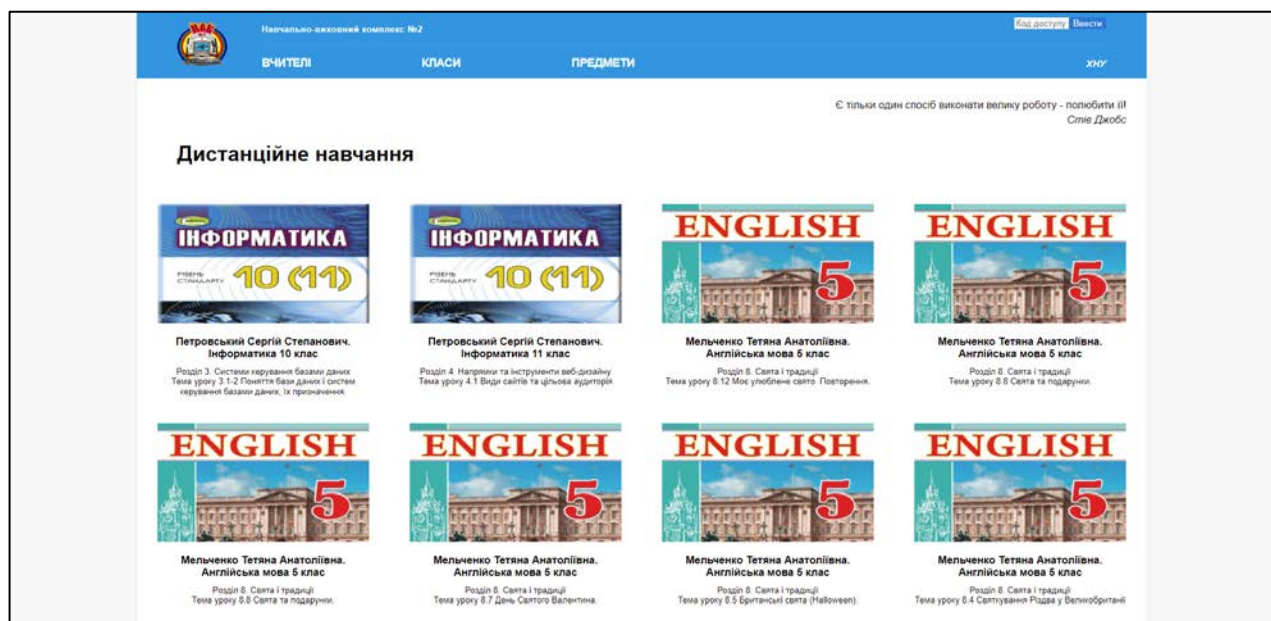


Рисунок 1.12 – Інформаційна система дистанційного навчання

Дивлячись на рисунок 1.12, сайт містить освітню інформацію у вигляді уроків,

кожен урок містить такі дані як назва, розділ книги, тема, предмет, клас та ПІБ вчителя, що його проводить (рисунок 1.13).



Рисунок 1.13 – Інформація про урок

Також є можливість швидкого пошуку необхідного уроку за допомогою сортування, для прикладу необхідно відкрити урок по геометрії будь-якого класу. Для виконання цього завдання переходимо у верхню область веб-додатку та наводимо курсором на предмети, після чого відобразиться випадаючий список всіх існуючих предметів на платформі. Обирається необхідний – геометрія, далі на сторінці з'являються всі відповідні уроки (рисунок 1.14).

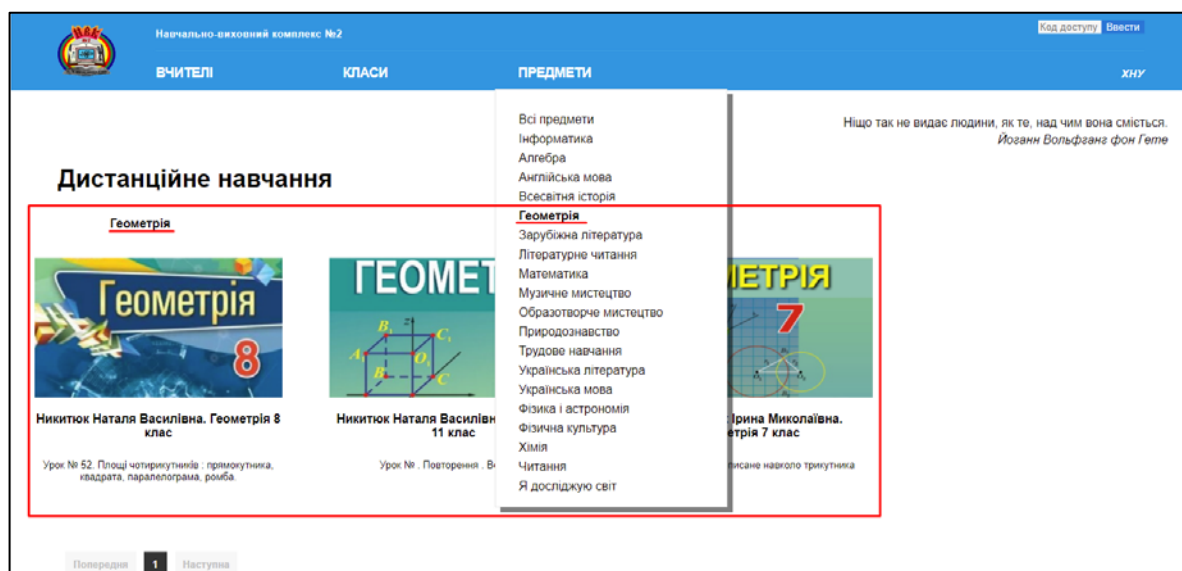


Рисунок 1.14 – Використання сортування

Хоч у користувача і є змога переглядати зовнішню інформацію про уроки, та знаходити необхідні, проте самого доступу до інформації, що знаходиться в середині у нього немає. Щоб отримати доступ потрібно пройти авторизацію шляхом введення відповідного коду доступу (рисунок 1.15).

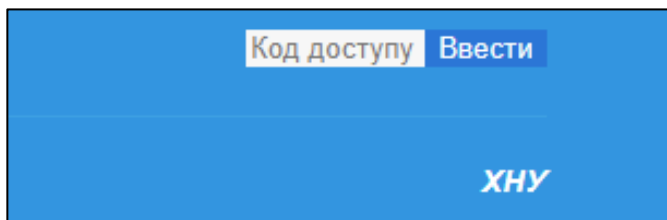


Рисунок 1.15 – Проходження авторизації для отримання вмісту уроків

Одним із способів отримання коду для користувача є звернення до адміністрації школи з відповідним проханням, і якщо код нададуть користувачу то при натиску на один з уроків він побачить його вміст (рисунок 1.16).

	ADVERB	TRANSLATION	EXAMPLE
100%	ALWAYS	завжди	<i>I always drive to work.</i>
90%	USUALLY	зазвичай	<i>I usually go out for dinner.</i>
70%	OFTEN	часто	<i>I often see my friends.</i>
50%	SOMETIMES	іноколи	<i>I sometimes go to the gym.</i>
30%	OCCASIONALLY	час від часу	<i>I occasionally play tennis.</i>
10%	SELDOM	рідко	<i>I seldom watch TV.</i>
5%	HARDLY EVER	майже ніколи	<i>I hardly ever drink coffee.</i>
0%	NEVER	ніколи	<i>I never smoke.</i>

3. Впр.6 ст102 доповніть речення необхідними словами (письмово)  
4. Підготуйте проєкт на тему: «Моя улюблена свята» «My favourite holiday» (зразок ст103)

Рисунок 1.16 – Вміст уроку

Згідно з зображенням 1.16 до вмісту уроку можуть входити відео, картинки і текст. Також можуть підкріплятися «pdf» та інші файли, додаватись посилання на корисні, зовнішні ресурси, що мають пряму лінійний зв'язок з темою уроку (рисунок 1.17).

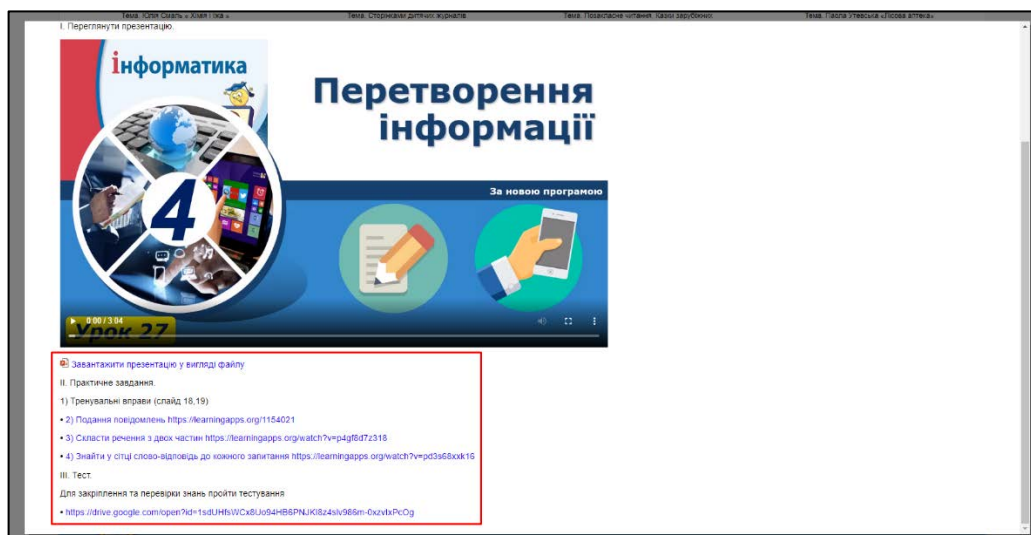


Рисунок 1.17 – Додаткові ресурси, що стосуються теми уроку

Після процесу розгляду прикладів існуючих вебсайтів дистанційних шкіл, можна визначити позитивні та негативні сторони кожного з них. До позитивних сторін першого прикладу потрібно віднести сучасний дизайн, який складає гарне враження про веб-додаток та відокремлення інформаційного веб-додатку від системи, що надає дистанційне навчання, цей підхід є дуже корисним, оскільки полегшує процес нововведень та редагування існуючих компонентів. Також потрібно відзначити надання кожній складовій освітнього матеріалу окремої сторінки, що запобігає нагромадженню інформації та сприяє її легкому засвоєнню. Підсумовуючи даний приклад можна сказати те, що він створений за допомогою сучасних технологій такої складової ІТ як програмування, не містить помилок, що стосуються працездатності. Отже його можна рахувати близьким до ідеалу й гідним прикладом по якому можна орієнтуватись при побудові власного веб-додатку.

Певні подібні плюси є також в другому прикладі, а саме:

1. Розділення інформаційного веб-додатку від програми, що надає дистанційне навчання.
2. Простий та зрозумілий інтерфейс для користувача.
3. Наповненість сайту великою кількістю інформації щодо предметів та уроків.
4. Працездатність всіх навігаційних елементів.

5. Авторизація для отримання вмісту уроків.

До плюсів також можна віднести:

1. Можливість швидко знайти необхідний освітній матеріал за допомогою сортування інформації.

2. Вміст уроків має картинки та відео, що сприяє легкому їх засвоєнню.

3. Присутність корисних посилань на додаткові ресурси, наприклад сайт з тестами для перевірки вивченого матеріалу учнями.

Проте є і слабкі сторони такі як, примітивний дизайн, а також відсутність вбудованого модуля для перевірки вивченого матеріалу учнями.

Таким чином виникає розуміння як саме має бути створена система, що надасть змогу дистанційно здобувати освіту, яку інформацію вона має містити в собі, який функціонал повинен бути реалізований для користувачів. Згідно проведеного аналізу можна дійти до висновку, що розробка програмного забезпечення для дистанційної освіти є перспективним у напрямку веб-технологій, оскільки даний напрямок є найбільш популярним для додатків такого типу, а також оптимальним з точки зору витрат.

### **1.3 Аналіз сучасних засобів створення програмного забезпечення**

У сучасному світі більшість сфер людської діяльності починають використовувати інформаційні технології. Приватні підприємства та державні установи ознайомлюються з новинками програмного ринку, організовують мітинги для вирішення питань побудови необхідного програмного забезпечення та наймання команди спеціалістів. Сфера освіти теж приймає участь у такому роді зайнятості, такі навчальні заклади як школи, університети, інститути, технікуми не так давно почали користуватися послугами ринку ІТ. Відповідно їх може зацікавити вирішення найбільш актуальних питань, які можуть стосуватися визначення фреймворку, стеку, типу застосування чи графічного рушія, на основі якого буде створюватись система. Зазвичай навчальні заклади обирають продукти, які розробляються веб напрямком та

його фреймворками, оскільки процес створення ПП не займе багато часу та відповідно не буде коштувати занадто багато для освітньої організації.

До найбільш актуальних фреймворків веброзробки відноситься Angular.JS, Ruby on Rails, Yii, Laravel, ASP.NET (ASP.NET MVC5).

Angular – це JavaScript фреймворк розроблений компанією Google, його мета полягала у створенні динамічних веб-додатків. Він є неймовірно популярним вже майже на протязі десяти років. За допомогою Angular, можна створювати інтерфейсні додатки без необхідності застосовування додаткових фреймворків або плагінів. До його структури входить цілий ряд корисних функцій. Ось деякі з них:

1. При використанні шаблонів, є можливість показувати інформацію моделі та контролера.

2. Angular підтримує одну з найпопулярніших архітектур – MVC. В результаті її застосування інженер може просто розмежувати програму на MVC складові. Все інше, що залишилось буде керуватися засобами фреймворка.

3. Всі вирази, які є схожими тим чи іншим чином на фрагменти коду, вкладені у фігурні дужки. Вони не застосовують умовні вирази або цикли. Скоріш навпаки – вони використовуються для відображення і форматування даних [8].

Проте даний фреймворк також містить і недоліки, ось деякі з них:

1. Присутність різноманітних структур (Components, Modules, Pipes, Injectables та інших), що ускладнює процес вивчення в порівнянні з подібними фреймворками (Vue.js, React), які містять лише «Component».

2. Відносно повільна продуктивність враховуючи різноманітні показники. Але з іншої сторони ця проблема має рішення у вигляді використання так званого «Change Detection Strategy», який допомагає ручним способом контролювати процес рендерингу компонентів [9].

3. В основі лежить складна мова програмування – JavaScript.

4. Виникнення помилок під час процесу міграції між версіями [10].

Всі вище перераховані функції є частиною платформи, яка надає змогу побудувати чудовий вебсайт. Про актуальність даного фреймворку свідчить список популярних вебсайтів, розроблених за його допомогою, до такого списку відносяться:

- Netflix;
- Freelancer.com;
- GoodFilms та інші [8].

Фреймворк Ruby on Rails згідно назви побудований на доволі таки відомій мові програмування – Ruby. Одною з його головних особливостей є те, що при повторному використанні коду, радикально спрощується і прискорюється розробка веб-додатків. Крім того, при використанні такого підходу виникає можливість додавати певні додаткові функції. Серед існуючих популярних веб-додатків, написаних за допомогою Ruby on Rails, можна виділити GitHub, Ask.fm, Basecamp, 500px та інші. Ruby on Rails – це величезне співтовариство веб-розробників, де можна за мінімальний час знайти допомогу для свого проекту. Хоч RoR не має ніякого відношення до ряду дешевих технологій. Проте фреймворк забезпечує значимий ряд можливостей. І в першу чергу мова йде про швидку можливість розробки з меншою кількістю написаного коду та відповідно помилок. Також до спектру переваг можна віднести:

1. Гнучкість. Фреймворк підходить для всіх галузей бізнес сфери, це може бути як управління проектами так і будівництво.

2. Швидкість. Про це вже було наголошено і повторюється ще раз: Ruby on Rails скорочує час розробки веб-додатків на значимий відсоток.

3. Можливість вносити зміни в код абсолютно не викликає будь-яких труднощів. По даній причині RoR ідеально підходить для проектів, які потребують довгострокової підтримки. Даний фреймворк також запобігає виникненню проблем при різкій зміні вимог або зміні команди [8].

Фреймворк Ruby on Rails також містить і слабкі сторони:

1. Виникнення труднощів з вивченням та початковим налаштуванням серверу.
2. Проблеми з хостингом, що пов'язані з відсутністю доступних цін спеціалізованого Rails хостингу. Оптимальним варіантом є VPS/VDS що коштує близько трьох доларів, проти PHP хостингів, що розпочинаються від ціни одного долару.

3. Відносно мала кількість документації або покрокового керівництва на російській мові [11].

YII – платформа, що містить відкритий вихідний код, а також є вбудованою в PHP5. Завдяки присутності зрозумілого дизайну і зручного інтерфейсу, даний фреймворк надає надзвичайно простий і швидкий процес веб-розробки. Yii оптимізований у напрямку продуктивності, тому його можна застосовувати для будь-яких проектів, від малих до великих. Крім того, платформа містить ряд чудових інструментів, які допоможуть з процесом налагодження і тестування додатку. Ще одним приємним плюсом є добре написана документація.

До ключових ефективних особливостей Yii відносяться:

1. Фреймворк спершу не включає класи і об'єкти, до того моменту поки не виникне потреба в їх використанні. Дана механіка прискорює завантаження веб-додатків.

2. Використання ООП стандартів, що свідчить про надійність та безпеку.

3. Абсолютна безкоштовність всіх професійних ресурсів [8].

До негативних особливостей Yii належать:

1. Не достатньо гнучке формування роутів (система маршрутів на сайті).

2. Відсутність регулярних оновлень, що вказує на поганий розвиток фреймворку.

3. Занадто пов'язані між собою бібліотеки frontend'а та backend'а [12].

Наступним популярним фреймворком є Laravel, який рахується одним з найкращих PHP фреймворків. Він надає можливість легко створювати веб-додатки і додатки для мобільних пристроїв. Реалізація таких вебзастосунків може бути у вигляді невеличких сайтів, а також у вигляді масивних веб-програм, що орієнтовані на бізнес. В арсеналі Laravel існує ряд корисних та цікавих особливостей, таких як об'єктно-орієнтовані бібліотеки, техніка авторизації, підтримка MVC, міжсайтові запити, міграції баз даних, та інші. Основними перевагами цього фреймворка є:

1. Присутність можливості збільшення трафіку на сайті. Причиною актуальності даної технології є те, що вона може бути застосована до будь-якого браузера та пристрою.

2. Гнучкість. За гнучкість даного фреймворку відповідає модульна структура, що допомагає спростити процеси розробки самого вебсайту.

3. PHP не є вимогливим до спеціальних способів обслуговування. Причиною даної особливості є автоматичне завантаження об'єкта, який знаходиться в програмному забезпеченні.

4. Фреймворк Laravel здатен генерувати унікальні URL-адреси, за рахунок реалізації різних маршрутів з однаковою назвою [8].

Laravel не є ідеальним за причиною присутності ряду недоліків, до якого відноситься:

1. Доволі маленька кількість статей, прикладів коду, перекладів офіційної документації для російськомовних користувачів, у яких слабкий рівень англійської мови або взагалі відсутнє її знання.

2. Синтаксичний цукор в даному фреймворку, що може бути як плюсом так і мінусом. Мова йде про здатність легкого вивчення, а також забуття процесу написання простих запитів та функцій.

3. Порушення зворотної сумістності між версіями фреймворку.

4. Не логічне розміщення каталогів та файлів. Для прикладу, по замовчуванню в каталозі `app` розміщена модель `User.php`, хоча для моделей існує окремий каталог `Models`, що свідчить про більш логічне розміщення раніше описаної моделі всередині. Каталог з назвою `resources`, що містить файли уявлень розміщений в корні проекту, хоча більш логічним було б його розташувати в `app/resources` [13].

ASP.NET – технологія створена компанією Microsoft, її мета полягає у наданні функціоналу, за допомогою якого можна створювати веб-додатки та веб-сервіси. Вона є частиною більш загальної платформи Microsoft.NET і продовженням її предка – технології Microsoft ASP. На даний момент часу останньою версією цієї технології є ASP.NET Core 2.0. Звернувши увагу на назву ASP.NET стає зрозумілим, що вона її успадковує від більш ранішої технології Microsoft ASP, проте вона значимо відрізняється від неї. Microsoft цілком перебудувала ASP.NET таким чином, що в даний момент платформа базується на Common Language Runtime (CLR), що є підґрунтям всіх теперішніх додатків створених за допомогою Microsoft .NET.

Інженери мають можливість писати код для ASP.NET, разом з використанням усіх мов програмування, що входять до комплекту .NET Framework, а саме: Visual Basic.NET, C#, F#, JScript.NET та інші. Важливим моментом є те, що ASP.NET має перевагу в швидкості при її порівнянні зі скриптовими технологіями, причиною цього є компіляція коду при першому зверненні. Продовженням даної активності є його переміщення в спеціально відведений кеш, і згодом виконання без витрат часу на парсинг, оптимізацію, та подібні процеси. До переваг ASP.NET відноситься:

1. Перевага в швидкодії при порівнянні з технологіями, що були засновані на скриптах таких як PHP та інших.

2. Гнучкий набір бібліотек класів та елементів управління, що надає змогу швидше розробляти веб-додатки.

3. ASP.NET вміщує у собі багатомовні можливості .NET, які дозволяють писати код на VB, C, C++, C#, та інших.

4. Розділення візуальної частини від бізнес-логіки.

5. Модель обробки запитів з можливістю розширення [14].

До слабких сторін платформи ASP.NET належить:

1. Початок роботи. У сфері IT існує безліч технологій, що стосуються веб-доменів, таких як PHP, Python, Ruby on Rails та інших. Найбільшим недоліком ASP.NET є його великий обсяг, в рамках якого існують багато модулів, вбудованих функцій, класів та бібліотек. Отже, це в сукупності створює величезну проблему для початківців, яка полягає у вивченні всього раніше перерахованого. З іншої сторони такі технології як Ruby on Rails та PHP дуже легко вивчити та відповідно писати нові коди в IDE.

2. Комплексні сторінки з проблемою продуктивності. Оскільки ASP.NET був створений, щоб кинути виклик Java JSP, тому очевидно знайти ефективні модулі в ASP.NET для розробки корпоративних стандартних веб-додатків. Він часто використовується при розробці великомасштабних веб-додатків, але важливі веб-сторінки, розроблені в ASP.NET, зазвичай працюють повільніше. Завантаження у браузері займає більше часу, що створює дразливий досвід для кінцевого користувача.

3. Додатки ASP.NET створюють проблему тестування. Тестувальники часто скаржаться на те, що їм здається нудним процес тестування невеликих модулів програм ASP.NET. Методології тестування, такі як модульне тестування та інтеграційне тестування, зазвичай створюють проблему для програм ASP.NET за багатьох обставин [15].

Відповідно до розглянутих існуючих фреймворків, платформ та їх переваг і недоліків було визначено найбільш оптимальний варіант – фреймворк Angular.

#### **1.4 Постановка задачі і вимоги до розробки інформаційної системи**

Метою кваліфікаційної роботи бакалавра є розробка інформаційної системи, що впроваджує дистанційну форму освіти учням загальноосвітніх навчальних закладів із використанням фреймворку Angular. Інформаційною системою будуть користуватись адміністратор та учень, відповідно у ній повинен бути присутній наступний функціонал:

- реєстрація та авторизація для користувачів системи;
- надання можливості редагування БД адміністратору;
- відображення вмісту БД учню у вигляді навчального матеріалу;
- надання можливості проходження тестів учню;
- відображення результатів проходження тестів у вигляді оцінок;
- надання можливості пошуку необхідної інформації учню.

Також обов'язковою умовою при розробці ІС є використання раніше спроектованої БД, що повинна бути створена за всіма необхідними критеріями.

## 2 Проектування структури інформаційної системи

### 2.1 Функціональна структура і бізнес-процеси системи

Програмне забезпечення націлене насамперед на учня, що робить його головним користувачем ІС. Оскільки, саме учень буде здобувати освіту в дистанційній формі, то програмний продукт повинен надавати йому можливість перегляду освітньої інформації, яка буде знаходитися у відповідних електронних підручниках. Також ПП має надавати можливість проходження тестів по вивченому матеріалу та показувати результати у вигляді оцінок. Отже для учня є сформований ряд відповідних функцій системи, які є об'єднані у спеціальну групу (рисунок 2.1).

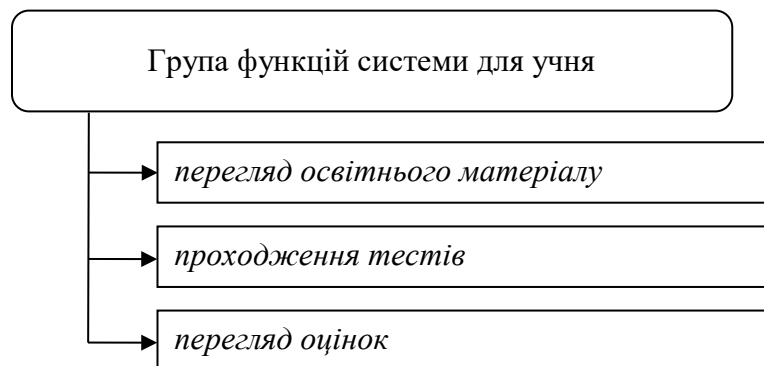


Рисунок 2.1 – Функції системи для учня

Відповідно до існуючої групи функцій системи учня, необхідно автоматизувати виконання відповідних бізнес-процесів. Для автоматизації необхідно реалізувати такі процеси як перегляд освітнього матеріалу, проходження тестів та перегляд оцінок.

*Бізнес-процес «перегляд освітнього матеріалу»* призначений для надання учням освітньої інформації, яка буде представлена в електронному вигляді. Тобто автоматизації підлягає наступний спектр його складових:

- загальний перегляд освітнього матеріалу;
- пошук необхідного матеріалу;
- пошук необхідної частини електронного підручника.

*Бізнес-процес «проходження тестів»* надає можливість учню перевірити засвоєні знання, які він здобув за допомогою електронних підручників. Тому автоматизації підлягають наступні його складові:

- перегляд існуючих тестів;
- пошук необхідних тестів;
- проходження потрібного тесту.

*Бізнес-процес «перегляд оцінок»* призначений для отримання результатів перевірки засвоєного матеріалу, які відповідно можуть бути переглянуті самим учнем та іншими користувачами ІС. Таким чином виникають складові даного бізнес-процесу, які необхідно автоматизувати:

- перегляд існуючих оцінок відповідних тестів;
- пошук оцінок конкретних тестів;
- пошук оцінок того чи іншого учня.

Також приймає участь у використанні програмного забезпечення і адміністратор, який буде мати змогу виконувати маніпуляційні дії в системі з користувачами, освітнім матеріалом, тестами та оцінками. Тому для адміністратора є сформований відповідний ряд функцій системи, що можуть бути об'єднані в спеціальну групу (рисунок 2.2).

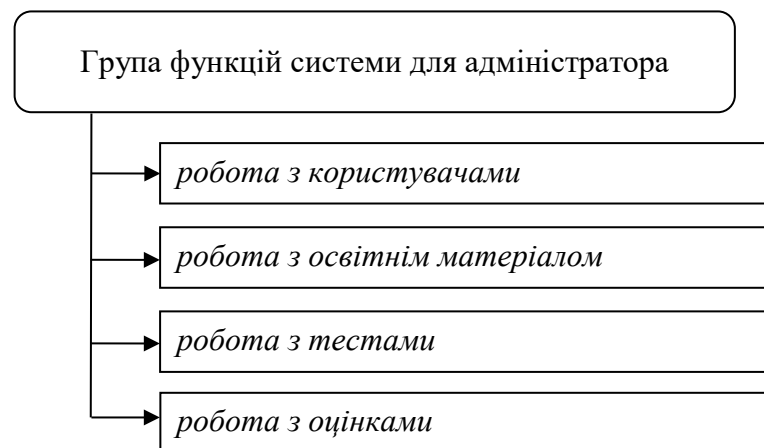


Рисунок 2.2 – Функції системи для адміністратора

Згідно до встановленої групи функцій адміністратора, необхідно автоматизувати відповідні бізнес-процеси. Під автоматизацію підлягають такі процеси як робота з користувачами, робота з освітнім матеріалом, робота з тестами та робота з оцінками.

*Бізнес-процес «робота з користувачами»* призначений для надання можливості адміністратору виконання маніпуляційних дій з користувачами системи. До дій, які підлягають автоматизації, вноситься:

- перегляд існуючих користувачів;
- видалення конкретних користувачів;
- додавання нових користувачів;
- надання певних прав конкретним користувачам.

*Бізнес-процес «робота з освітнім матеріалом»* призначений для надання адміністратору ІС функціоналу, за допомогою якого він може виконувати різноманітні операції з освітнім матеріалом, тобто з електронними підручниками. Тому автоматизації підлягають наступні операції:

- створення електронних підручників;
- перегляд існуючих електронних підручників;
- видалення конкретного освітнього матеріалу;
- редагування освітнього матеріалу.

*Бізнес-процес «робота з тестами»* призначений для надання функціоналу адміністратору, який здатен забезпечити перевірку засвоєних знань кожного користувача даної ІС. Відповідно автоматизації підлягають наступні функції:

- створення нових тестів;
- редагування існуючих тестів;
- видалення конкретних тестів.

*Бізнес-процес «робота з оцінками»* призначений для надання адміністратору змоги виконувати різні операції з результатами перевірки засвоєного матеріалу учнями у вигляді оцінок. Таким чином автоматизації підлягає наступне:

- перегляд існуючих оцінок;

- редагування необхідних оцінок;
- видалення конкретних оцінок.

## **2.2 Інформаційна структура системи**

Відповідно до постановки завдання, необхідно спроектувати програмний продукт належним чином, а також його структуру в оптимальному варіанті. Щоб знайти рішення для даної задачі, слід повторно передивитись раніше описані приклади, та на основі їх сильних сторін спроектувати вебзастосунок. Таким чином виникає розуміння, які потрібно зазначити моменти, що стосуються впровадження коректної роботи системи для різних груп користувачів майбутнього програмного продукту. До такої роботи можна віднести надання конкретного функціоналу системи відповідно конкретній групі користувачів.

Наприклад, щоб учні стали користувачами системи, вони повинні пройти спершу реєстрацію, а згодом при повторному використанні програми, проходити авторизацію до системи. Після чого у них з'являється можливість працювати з функціоналом, що надає ІС. До додаткового функціоналу, який може бути доданий в систему, крім того що вимагають бізнес-процеси є пошук необхідної сторінки електронного підручника. Дана функція впровадить приємний досвід перегляду електронних підручників для користувачів, оскільки знайти необхідний матеріал в підручнику з чотирмастами сторінками є не легким завданням.

Схожим чином можна налагодити роботу системи і для її адміністраторів. Вони теж повинні проходити початкову реєстрацію, і також при повторному використанні веб-застосування проходити авторизацію, якщо відповідно був раніше здійснений вихід із системи. Щоб стати початковим адміністратором даної системи потрібно ввести конкретний домен пошти у процесі реєстрації. Який саме домен повинен бути введений, вирішує розробник, а точніше він отримує відповідні вказівки від замовника.

Для більш кращого розуміння як саме повинна бути організована система, яка впроваджує дистанційну форму навчання для учнів ЗНЗ, слід звернути увагу на діаграму варіантів використання, що зображена на рисунку 2.3.



Рисунок 2.3 – Діаграма варіантів використання ІС

Після отримання розуміння, яке надає ілюстрація функцій програмного продукту, слід спроектувати БД, яка безпосередньо є необхідною для його реалізації.

Для створення БД, потрібно визначити ключові сутності, які приймають роль у використанні програмного забезпечення та їх властивості. До таких сутностей відносяться:

- учні;
- адміністратори;
- навчальні посібники;
- тести;
- оцінки.

Відповідно до предметної області було встановлено кожній сутності необхідні властивості. Для прикладу, до атрибутів учнів входить ПІБ, також як і для адміністраторів. В свою чергу навчальні посібники містять назву, авторів та клас. Інша сутність – тести, містить назву для їх ідентифікації, крім назви присутні також питання та відповіді до них. А оцінки мають лише числове значення, яке ідентифікує результат перевірки засвоєних знань учнів. Отже, загальну структуру сутностей та їх всіх, ключових властивостей можна побачити на рисунках 2.4 – 2.6.



Рисунок 2.4 – Властивості користувача

Відповідно до того факту, що користувачами програмного забезпечення є адміністратор та учень, то у них ідентичний ряд властивостей. Тому сенсу будувати діаграми варіантів використання ключових властивостей для кожного з них немає.

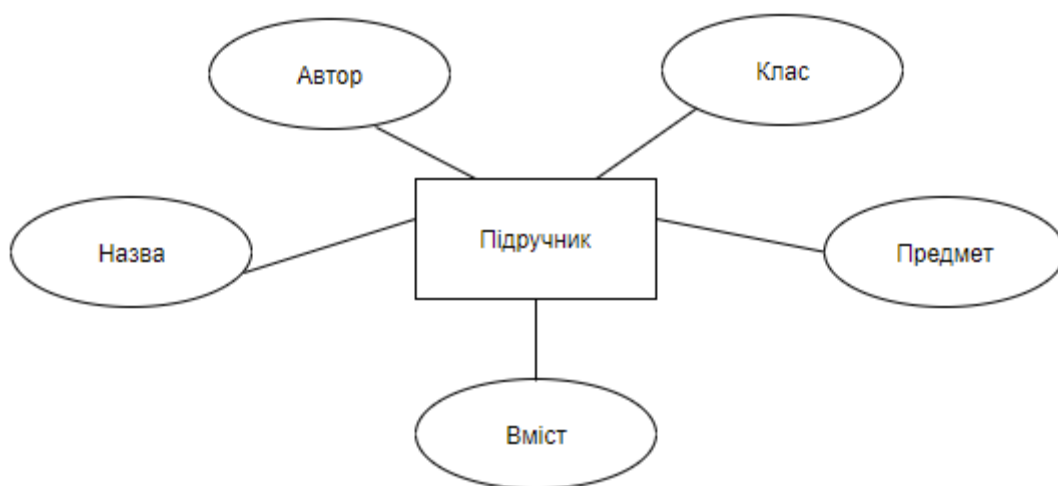


Рисунок 2.5 – Властивості підручника

На основі даних предметної області, підручник володіє атрибутами, що наведені на рисунку 2.5. Такі атрибути як предмет, автор, та клас є додатковими, а точніше такими які у майбутньому нададуть можливість розробнику впровадити можливість пошуку. Це означає що підручнику достатньо мати назву, яка буде вміщувати всі вище перераховані додаткові параметри. Відповідно також необхідний його вміст з точки зору достатності, але не потрібно забувати про частину програмування.

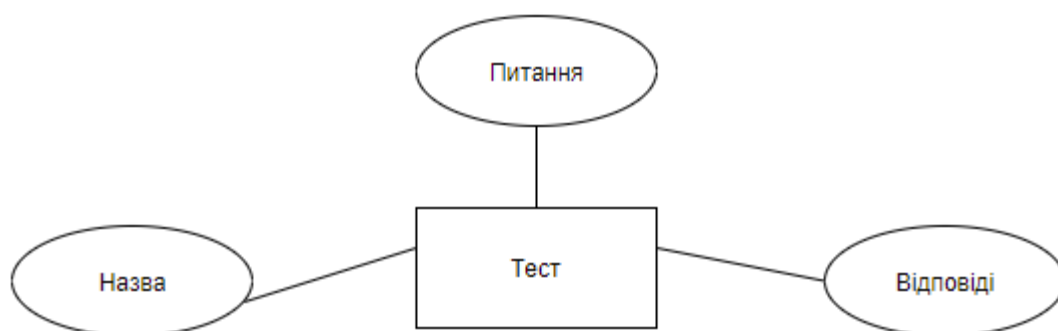


Рисунок 2.6 – Властивості тесту

Для даної сутності необов'язково звертатись до предметної області, достатньо звернутись до логіки та отримати відповіді. Проте, кожен був у школі та інших подібних навчальних закладах та відповідно має уявлення якими властивостями може володіти тест, мова йде саме про назву, питання та відповіді до них.

В свою чергу, оцінка володіє такою ключовою властивістю як числове значення, тому сенсу ілюструвати лише її немає. В підсумку можна сказати, що були розглянути всі ключові властивості сутностей, проте можуть виникати зайві питання. Для уникнення таких питань, був створений ряд таблиць, який детально пояснює кожен сутність та її всі атрибути (таблиці 2.1 – 2.4).

Таблиця 2.1 – Атрибути користувача

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Числовий ідентифікатор
2.	name	{}	Дані користувача, такі як ПІБ
3.	data	{}	Певна біографічна інформація про користувача
4.	avatar	{}	Картинка профілю користувача
5.	email	{}	Електронна адреса за допомогою якої користувач зможе зареєструватись та авторизуватись до сайту
6.	password	{}	Пароль який користувач створює при реєстрації, та може використовувати для авторизації у майбутньому
7.	is	{}	Роль користувача, яка поділяє його на такі групи як студент та адміністратор

Після всіх розглянутих атрибутів, які відносяться до користувача системи, виникає чітке розуміння того, як саме повинні бути впроваджені у вебзастосунок учень та адміністратор. Також виникає обізнаність щодо коректності та достатності атрибутів, якими повинен володіти кожен з них для логічної взаємодії з системою, яка у майбутньому буде виконувати різноманітні дії з їх даними.

Таблиця 2.2 – Атрибути підручника

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Числовий ідентифікатор
2.	name	String	Назва підручника
3.	tb_author	String	Автор, який працював над написанням підручника
4.	class	String	Клас, за яким відбувається класифікація підручника
5.	subject	String	Предмет, що представляє ту чи іншу науку, також може виступати у ролі елемента класифікації
6.	content	String	Вміст підручника, що являє собою текст та картинки
7.	image	String	Картинка, що представляє титульну частину підручника

Процес формування атрибутів підручника безпосередньо базувався на предметній області та логіці, яку потребує програмна реалізація системи. Таким чином виникає запобігання незрозумілих моментів, які можуть виникати при розробці вебзастосунку, а точніше його інформаційної структури.

Таблиця 2.3 – Атрибути тесту

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Числовий ідентифікатор
2.	name	String	Ім'я тесту
3.	question	[]	Масив питань, який буде вміщувати відповіді до кожного з них

У випадку формування атрибутів тесту, також була звернута увага на предметну область, в якій існує даний спосіб перевірки знань учнів ЗНЗ. Тому, таблиця атрибутів тесту володіє всіма необхідними елементами для його програмної реалізації.

Таблиця 2.4 – Атрибути оцінки

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Числовий ідентифікатор
2.	mark	int	Числове значення оцінки

Відповідно до раніше наведеної таблиці 2.4, сформовано лише два атрибути оцінки, і дана кількість є цілком достатньою. Причиною даного твердження є те, що логіка інформаційної системи базується на предметній області, і вимагає певну ідентифікацію оцінки того, чи іншого учня за конкретний тест. Відповідно сама оцінка, до якої прив'язана ідентифікація має бути представлена у числовій формі.

Після розглянутих таблиць та їх вмісту, можна дійти до висновку, що створювана БД буде зберігати всю необхідну інформацію для впровадження і функціонування системи, що забезпечить дистанційну форму навчання. Також можна зазначити, що наведені таблиці та діаграми своєрідним чином реалізують поставлені раніше завдання у пункті 1.4. Наприклад, діаграма варіантів використання показує загальну картину експлуатації складових ІС користувачами, а також паралельно демонструє присутність всіх необхідних модулів для логічного та коректного функціонування вебзастосунку.

В свою чергу діаграми ключових властивостей сутностей, надають змогу сформувати відповідні документи у БД, які будуть їх представляти. Вони виділяють лише ключові атрибути, оскільки саме вони є необхідними для початкової роботи програмного продукту. Таблиці виступають в якості доповнення, а точніше містять додаткові атрибути, які можуть бути застосовані в необхідний момент при використанні складових вебзастосунку, що безпосередньо мають зв'язок з БД.

Тому, для створення інформаційної системи, яка буде наділена раніше наведеними функціями, слід обрати відповідні засоби розробки.

## 2.3 Вибір засобів розробки інформаційної системи

### 2.3.1 Вибір мови програмування

Оскільки, в якості фреймворку, на якому буде базуватися інформаційна система було обрано Angular, то відповідно мовою програмування буде JavaScript. Дана мова відноситься до динамічних, об'єктно-орієнтованих мов програмування з елементами прототипності. Найбільше її використовують для створення скриптів вебсторінок, що відповідно забезпечує можливість на пристроях кінцевого користувача працювати з споживачем, браузером та асинхронно обмінюватись даними з сервером. Також з її допомогою можна модифікувати структуру та зовнішній вигляд сторінки веб-додатку.

JavaScript може застосовуватись для наступного:

1. Створення односторінкових та прогресивних вебзастосунків, в яких також приймають участь такі фреймворки як Angular JS, React, Vue.js.
2. Написання скриптів вебсторінок для надання їм інтерактивності.
3. Програмування на стороні серверної платформи Node.js з допомогою його фреймворку Express.js.
4. Створення програм для мобільних пристроїв за допомогою Cordova, React Native та інших.
5. JS містить незначну кількість властивостей, характерних функціональним мовам, до таких властивостей відносяться функції, які подані у формі об'єктів першого класу, також об'єкти у формі списків. Присутній каррінгу, анонімні функції, замикання, що у сукупності дає мові ефективну гнучкість та причину її використання [6].

До переваг даної мови можна віднести:

- швидкість, яка демонструється при запуску клієнтського браузера;
- відсутність перешкодження мережевих викликів до серверу;
- простота, яка полягає у простому процесі вивчення та застосування JS;
- сумісність з іншими мовами програмування;

- запобігання навантаження сервера;
- можливість створення розширених інтерфейсів.

Дивлячись на той факт, що JS вважається лідером серед скриптових мов, у ній також присутні недоліки, якими можна вважати:

1. Ненадійність захисту клієнтської частини. Оскільки код виконується на комп'ютерах користувачів, то у деяких випадках його можна змінити для зловмисних цілей. Це є однією з причин, яка відвертає увагу від JavaScript.

2. Некоректна підтримка. Інколи JavaScript інтерпретується різними браузерами по-різному. Це в певній мірі ускладнює написання коду між браузерами [19].

Хоча у JavaScript присутня певна кількість недоліків, вона не змістить JS з першої позиції мов програмування, які застосовуються при розробці як веб-додатків так і вебсайтів. Головною і єдиною причиною обрання даної мови є те, що вона в сумісності з такими фреймворками як Angular JS, React, Vue.js здатна побудувати необхідний програмний продукт.

### **2.3.2 Вибір серверної платформи**

В даний момент часу найбільш популярними серверними платформами є Node.js та PHP. Хоча PHP рахується скриптовою мовою, вона також може відігравати роль серверної платформи аналогічно до того, як це робить Node.js. PHP створений у вигляді скриптів, які можуть містити JavaScript, HTML, CSS, або вміст звичайного тексту, що є необхідним для веб-розробки. PHP-скрипти здійснюються на стороні серверу, згодом їх результати повертаються до веб-браузера користувача у вигляді звичайного HTML файлу, зображення, PDF файлів, або навіть у відео чи аудіо форматах.

До функціоналу PHP відноситься:

- створення динамічного вмісту сторінки;
- виконання CRUD операцій з файлами на сервері;

- зберігання та використання форм з даними;
- надсилання та отримання файлів cookie;
- додавання, видалення, зміна даних у БД;
- контроль доступу користувачів;
- шифрування даних.

Крім функціоналу, який може використовуватися для реалізації раніше поставленого завдання, слід зазначити переваги використання РНР, до яких відноситься:

- кросплатформенність;
- підтримка великої кількості баз даних;
- безкоштовність;
- легкість при вивченні [16].

У РНР також присутні недоліки, ось деякі з них:

- Несумісність з веб-додатками, що мають гігантський вміст.
- Присутність слабкого типу, що може спричинити неправильні дані.
- Використання великої кількості фреймворків та інструментів РНР, що спричиняє низьку продуктивність веб-додатків [17].

Іншою серверною платформою, яка користується великою популярністю на протязі останніх років є Node.js. Дана платформа надає можливість виконувати JavaScript сценарії на сервері, після чого надсилати результат їхнього процесу виконання розробнику. Якщо раніше Javascript використовувався з цілю обробки даних в браузері користувача, то тепер платформа Node.js разом з її фреймворком Express.js перетворила JavaScript на мову загального використання з великою спільнотою розробників [18].

До функціонала, який може створити необхідний програмний продукт, відноситься:

- генерація динамічного вмісту сторінок;
- виконання CRUD операцій з файлами на сервері;
- зберігання даних форми;

- додавання, видалення та зміна у БД.

До переваг даної платформи можна віднести:

- ефективність мови;
- чудову взаємодію з фреймворком Angular;
- обмін та багаторазове використання коду;
- швидкість та продуктивність;
- присутність менеджера пакетів (NPM);
- велика кількість доступних та безкоштовних інструментів [16].

Аналогічно до всіх раніше розглянутих прикладів вебзастосунків чи клієнтських фреймворків, дана серверна платформа також містить певний ряд недоліків, до якого відноситься:

- зниження продуктивності при обробці важких обчислювальних задач;
- вимагання великої кількості змін у коді завдяки нестабільному API;
- наявність асинхронного програмування, що ускладнює підтримку коду [20].

Після процесу розгляду обох серверних платформ, з врахуванням їх слабких та сильних сторін, а також орієнтуючись на поставлене завдання, було обрано Node.js. Ключовим фактором, який сприяв обранню Node.js, є його сумісність з фреймворком клієнтської частини під назвою Angular.

### **2.3.3 Вибір системи керування базами даних**

Оскільки розробка програмного забезпечення включає в свій ряд компонентів БД, то є необхідним обрання системи керування базами даних.

Однією з ряду актуальних СКБД є MongoDB, яка користується великою популярністю з моменту її виходу. MongoDB відноситься до документо-орієнтованих систем керування базами даних, при цьому вона не потребує опису схеми таблиць. MongoDB займає позицію між швидкими та масштабованими системами, які виконують певні операційні дії з даними у форматі ключ/значення, і реляційними

СКБД, які в свою чергу є зручними у створенні запитів та володіють функціональною характеристикою.

MongoDB може використовуватись для рішення таких проблем, які можуть виникнути при розробці інформаційної системи:

1. Інтеграція великих обсягів різноманітних даних. Якщо необхідно об'єднати десятки або сотні джерел даних, такі властивості як гнучкість та потужність MongoDB можуть створити єдиний вигляд таким чином, що інші бази даних не можуть.

2. Опис складних структур даних, що розвиваються. MongoDB дозволяє вбудувати документи для опису вкладених структур і легко переносить зміни даних у різних поколіннях документів.

3. Передача даних у високопродуктивних додатках. Розширена архітектура MongoDB може підтримувати величезну кількість транзакцій у величезних базах даних [21].

MongoDB містить і ряд сильних сторін, до яких відноситься:

1. Безсхемний тип БД NoSQL. При використанні даної СКБД не потрібно розробляти схему бази даних, що відповідно значно економить час. Таким чином, код який ми пишемо, визначає схему.

2. MongoDB не потребує складних об'єднань, у ній відсутній зв'язок між даними.

3. MongoDB можна легко масштабувати.

4. Легка при встановленні та налаштуванні.

5. Мова запитів документів, що підтримується MongoDB, дуже проста у порівнянні із запитами SQL.

У MongoDB також присутні і слабкі сторони, до яких відноситься:

- використання великої пам'яті для зберігання даних;
- існує обмеження на розмір документа, у значенні 16 МВ;
- відсутність підтримки транзакцій [22].

Іншою розповсюдженою СКБД, та прямим конкурентом MongoDB є PostgreSQL. PostgreSQL рахується об'єктно-реляційною системою керування базами даних. До основних можливостей даної СКБД можна віднести:

1. Функції, які дозволяють виконувати певний код за допомогою серверу бази даних.
2. Індеси, які можуть бути різних типів, починаючи від В-дерева, закінчуючи R-деревом з можливістю створення додаткових.
3. Багатоверсійність, що вказує на підтримку одночасної модифікації БД кількома користувачами [23].
4. PostgreSQL надзвичайно стабільна СКБД, вона підтримується більш ніж 20-річним розвитком суспільства, що сприяло її високому рівню стійкості, цілісності та коректності. PostgreSQL використовується як головне сховище даних для багатьох веб-програм, мобільних, геопросторових та аналітичних програм [24].

До сильних сторін PostgreSQL відноситься:

- легкість у використанні;
- присутність визначеного користувачем типу даних;
- використовує збережені процедури;
- містить атомність, консистенцію, ізоляцію та довговічність.

Дана СКБД також містить ряд недоліків, до яких належить:

1. Велике використання пам'яті при створенні окремих служб для кожного клієнта.
2. PostgreSQL у порівнянні з MongoDB програє у такому параметрі як продуктивність.
3. Відсутня швидкодія при порівнянні з аналогічними інструментами.
4. Присутність складної реплікації.
5. Складний процес встановлення для початкового розробника [25].

Отже, після аналізу найбільш актуальних СКБД в даний момент часу, які використовуються у напрямку веб-технологій, із врахуванням їх сильних та слабких сторін, було обрано СКБД під назвою MongoDB. Ключовими причинами, які сприяли

даному вибору є те, що MongoDB користується більшою увагою, що відповідно вказує на її величезну підтримку, також вона здатна взаємодіяти з такими технологіями як Angular, Express.js та Node.js.

### 2.3.4 Вибір редактору програмного коду

В останній час найбільшою популярністю серед редакторів програмного коду користується VS Code та Sublime Text.

Sublime Text є чудовим редактором програмного коду за причиною присутності таких характеристик як швидкість та кросплатформенність. Крім цього він підтримує різноманітні плагіни, і не містить велетенського функціоналу, який міг би залякати початкового розробника. Sublime Text надає ряд варіативних тем для інтерфейсу користувача, з можливістю завантаження інших. Присутня навігація у вигляді міні-карти у верхньому, правому куті екрану, що відповідно забезпечує швидкий та приємний перехід до необхідного фрагменту. Також вбудована зміна режиму екрану, її можна задіяти натискаючи на відповідні комбінації клавіш. Дана функціональність містить чотири режими, починаючи з першого, який показує лише вміст одного файлу, закінчуючи четвертим, що здатен показати одразу чотири файли, відповідно це може пришвидшити процес розробки інформаційної системи. Sublime Text наділений множинною правкою, яка полягає у виділенні необхідного фрагменту коду на всіх рядках за допомогою відповідної комбінації клавіш, після чого можна його змінити одночасно у всьому файлі. При написанні коду користувачем на тій чи іншій мові програмування, редактор надає різні варіанти для завершення запису, крім цього він здатен самотужки додати закінчення тому чи іншому фрагменту коду. За допомогою навігаційного інструменту можна швидко здійснити перехід по файлам, а також всередині них. Це відбувається при введенні того чи іншого нечіткого запиту у пошукове вікно, що відкривається натисканням відповідної комбінації клавіш [26].

Visual Studio Code рахується засобом створення, редагування та налагодження сучасних вебзастосунків і програм, що характеризуються як хмарні системи. Даний редактор коду є безкоштовним та кросплатформенним. До функціоналу Visual Studio

Code відноситься вбудований процес налагодження програм, ряд інструментів для роботи з Git і засобами рефакторингу. Присутня навігація по коду, також вбудоване автоматичне закінчення найбільш часто використовуваних та типових конструкцій або фрагментів коду. Присутня підтримка розробки на таких платформах як ASP.NET та Node.js, що вказує на позиціонування VS Code легковагового рішення, що надає змогу обійтися без повного інтегрованого середовища розробки [27].

Підсумовуючи потрібно зазначити, що в якості засобів розробки інформаційної системи вибрано стек технологій під назвою MEAN, який означає, що в ролі СКБД виступає MongoDB, роль серверного фреймворку відіграє Express.js, за клієнтську частину відповідає Angular, і серверною платформою є Node.js. А в якості редактору програмного коду, було обрано Sublime Text 3.

### 3 Програмна реалізація інформаційної системи

#### 3.1 Структура та функціональне призначення складових системи

Відповідно до поставленого завдання, процес створення інформаційної системи включає в себе певну кількість необхідних модулів для коректної та логічної роботи. Проте, слід виділити головні модулі, які можуть забезпечити мінімально-необхідний функціонал програми. До таких модулів відноситься авторизація та пошук. Кожен з них містить певне функціональне призначення, яке націлене на загальне покращення вебзастосунку або приведення його до оптимальної форми перед використанням. Отже, метою авторизації є перевірка належності даних користувача до системи, даний процес містить певну структуру послідовних дій, яка зображена на рисунку 3.1.

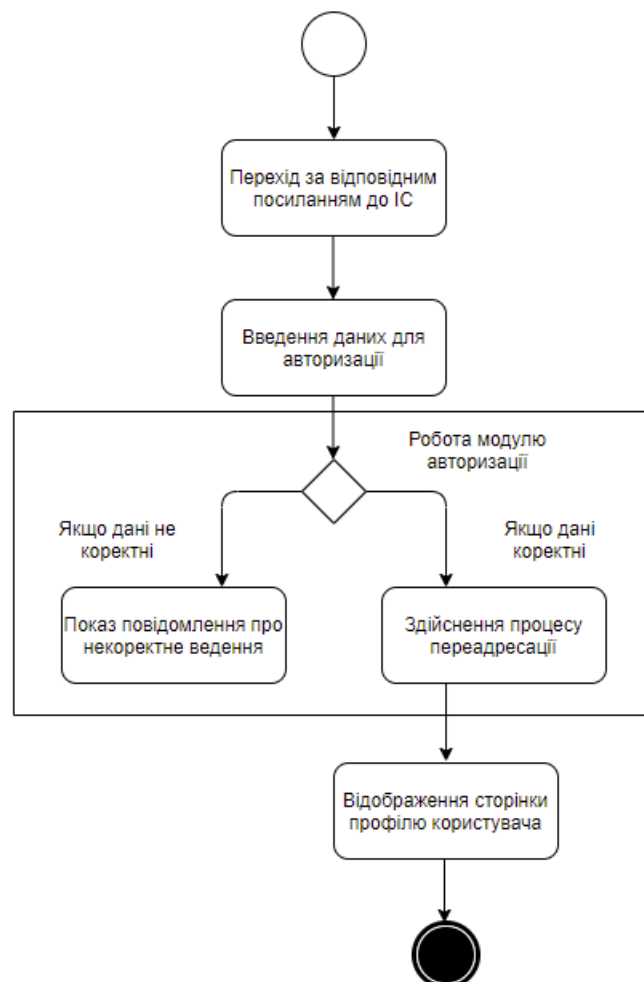


Рисунок 3.1 – Діаграма діяльності модуля авторизації

Після переадресації на профіль, у користувача є можливість перейти на іншу сторінку, яка може містити спеціально відведене вікно для здійснення операції пошуку. Таким чином виникає послідовність роботи модулів системи.

У свою чергу, функціональним призначенням модуля пошуку є надання можливості користувачеві знаходити потрібну для нього інформацію. Пошук інформації буде здійснюватися шляхом введення того, чи іншого запиту у відповідне поле пошуку. Проте, необхідно враховувати різноманітні сценарії розвитку подій. Наприклад, у випадку введення некоректної інформації або даних що не містяться у системі, результатом виконання даної операції буде порожнина. У випадку, якщо дані введено коректно, або введені дані містяться у системі, користувачу вони відобразяться у відповідному форматі. Детальну роботу модулю пошуку можна побачити на рисунку 3.2.

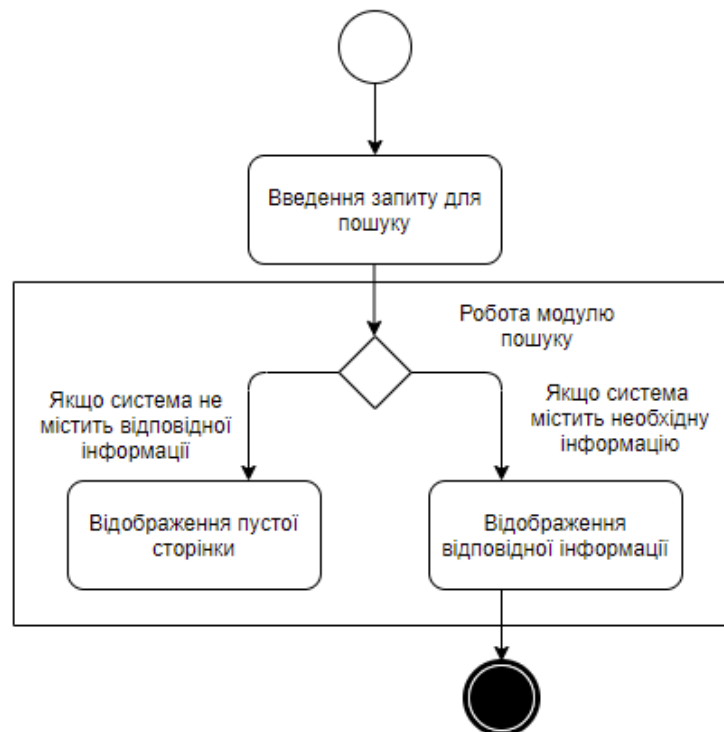


Рисунок 3.2 – Діаграма діяльності модуля пошуку

Також, у модулі авторизації присутня додаткова функція, яка може бути задіяна на етапі входу до системи. А саме у випадку, коли користувач забув власний пароль. Тобто, ціллю даної функції є відновлення паролю користувача системи. Вона

розпочинає свою діяльність з того моменту, як користувач натисне на відповідне посилання, після чого йому необхідно буде ввести свою електронну адресу. Електронна адреса використовується для отримання коду відновлення паролю. Після того, як користувач отримав такий код, він може повернутись до системи та ввести його у відповідне поле і створити новий пароль.

У системі окрім модулів та їх допоміжних складових, присутній унікальний ряд функцій, який надає можливість адміністратору виконувати операції CRUD з даними відповідної БД. Робота однієї з таких функцій розпочинається з натиску на певну кнопку, або подібний графічний елемент. Далі всередину функції передаються в якості параметру відповідні дані, і в залежності від призначення функції відбудеться конкретна операція. Приклад виконання такого процесу зображено на рисунку 3.3.



Рисунок 3.3 – Діаграма діяльності виконання однієї з операцій CRUD

Отже, після отримання детального опису роботи модулів, а також функцій, які вони повинні виконувати для коректної роботи інформаційної системи, необхідно розпочати їх розробку.

### 3.2 Особливості реалізації складових системи

До найбільш важливих та цікавих алгоритмів складових системи, можна віднести модуль авторизації та модуль пошуку. Кожен з них потребує коректної та логічної реалізації, оскільки в сукупності вони повинні забезпечувати мінімально-необхідний функціонал для користувача системи.

Розробка модуля авторизації розпочинається з зовнішнього вигляду, а точніше з форми, в яку вводяться особисті дані для входу у систему. Така форма створюється за допомогою ряду технологій, до якого входить HTML, CSS та Angular. Сукупність даних технологій забезпечує можливість написання коду, який реалізує зовнішній вигляд модуля авторизації:

```
<div class="auth-wrapper">
  <div class="auth">
    <div class="auth-title">Вхід</div>
    <form class="auth-form" (ngSubmit)="login()">
      <div class="waw-input mb15">
        <span>Email:</span>
        <input (keydown.enter)="login()" type="text"
placeholder="Email" name="email" [(ngModel)]="user.email">
      </div>
      <div class="waw-input mb15">
        <span>Пароль:</span>
        <input (keydown.enter)="login()" type="password"
placeholder="Password" name="password" [(ngModel)]="user.password">
      </div>
      <div class="auth-form_btn"><button class="waw-btn
_primary" type="submit">Увійти</button></div>
    </form>

    <div class="auth-link">
      <a routerLink="/reset">Забули пароль?</a>
      <a routerLink="/sign">Немає акаунту?</a>
    </div>
  </div>
</div>
```

Дивлячись на лістинг попереду, можна помітити присутність всіх, необхідних елементів для вводу особистих даних та відповідно їх процесу обробки. Для більш детального розуміння можна поглянути на сьомий рядок, у ньому присутній HTML елемент для введення тексту, всередині якого зберігається виклик функції для виконання відповідної операції. В якості доповнення йде модель Angular, яка пов'язує значення даного поля з властивістю відповідної сутності БД, після чого здійснює її

зберігання. Також доданий графічний елемент у вигляді кнопки, що підтверджує виклик функції входу. Реалізація у вигляді програмного коду такої функції розташована в іншому файлі, задля уникнення нагромодження коду та його комфортного процесу редагування:

```
login() {
  console.log(this.alert.show);
  if(!this.user.email) {
    return this.alert.show({
      type: "error",
      text: 'Enter your email',
      position: "center"
    });
  }
  this.hash.set('email', this.user.email);
  if(!this.user.password) {
    return this.alert.show({
      type: "error",
      text: 'Enter your password'
    });
  }
  this.http.post('/api/user/status', this.user, (resp:any) =>
{
  if(resp.email && resp.pass) {
    this.http.post('/api/user/login', this.user,
(user:any) => {
      if(!user){
        return this.alert.show({
          type: "error",
          text: "Something went wrong",
        });
      }
      localStorage.setItem('waw_user',
JSON.stringify(user));
      this.router.navigate(['/profile'])
    })
  } else {
    this.alert.show({
      type: "error",
      text: "Wrong email or password"
    });
  }
})
}
```

Верхній лістинг демонструє, що вміст функції логіну складається з кількох умов та дій. Метою умов є перевірка введених користувачем особистих даних (електронної пошти та паролю), які повинні бути аналогічними до тих, що вводилися на етапі реєстрації. В свою чергу, дії є простим та логічним завершенням перевірки. У випадку некоректного введення даних, в умові буде спрацьовувати функція з відповідним повідомленням. Далі відбувається ідентифікація користувача, іншими

словами перевіряється його належність до системи, якщо все гаразд здійснюється переадресація на сторінку власного профілю.

Створення пошуку, з точки зору структури файлів є аналогічним до авторизації, причиною даного твердження є використання раніше визначеного ряду технологій для побудови вебзастосунку. Проте, даний модуль має власну мету, відповідно кожна його складова може мати унікальний об'єм у порівнянні з авторизацією. У даному випадку представлення обмежується лише двома рядками коду, які представляють пошукове вікно для надання можливості введення необхідної інформації:

```
<input type="text" placeholder="Для пошуку натисніть тут.." name="search"
[(ngModel)]="search" class="search">
```

Лістинг вище демонструє, що зовнішня частина містить елемент введення тексту, всередині якого розташований ряд атрибутів. Найбільш важливим атрибутом, який потребує уваги є модель Angular, яка впроваджує зв'язок даного елемента з файлом, в якому написана функціональна частина модуля. Інші атрибути відіграють допоміжну роль, наприклад атрибут «placeholder» показує в пошуковому вікні текст-підказку, атрибут «name» використовується для ідентифікації, а «class» для використання стилів.

Файл, що містить функціональну частину пошуку, володіє рядом параметрів, які перевіряються відповіднимим умовами задля отримання необхідного результату. Наприклад, функція отримує в якості одного з параметрів масив документів, над яким буде здійснюватися сам процес пошуку. Іншим параметром, що надходить до функції є змінна пошуку, яка буде відігравати роль ідентифікатора завершення процесу.

Отже, в цілому робота алгоритму пошуку розпочинається з процесу перевірки присутності символів у пошуковому вікні, якщо процес завершився невдало, то повертається масив документів без змін. При успішному завершенні процесу визначається довжина масиву, далі береться значення з пошукового поля та перевіряється на наявність збігів зі всіма елементами. У випадку збігу, користувач

побачить лише конкретне значення, всі інші будуть видалені з масиву. Реалізацію функції, яка забезпечує знаходження необхідної інформації, демонструє лістинг нижче:

```

if(!s){
    return given;
}
if(typeof f == 'number'){
    l = f;
    f = null;
}
if(i || !s){
    if(l && Array.isArray(given)) return given.slice(0, 1);
    else return given||[];
}
let _arr = [], _check = {};
if(!Array.isArray(s)&&typeof s == 'object'){
    let _s = [];
    for(let key in s){
        if(s[key]) _s.push(key);
    }
    s = _s;
}
if(typeof s == 'string'){
    s = [s];
}
if(!f) f = ['name'];
if(typeof f == 'string') f = f.split(' ');
let sub_test = function(obj, _f, initObj, check){
    if(!obj) return;
    if(_f.indexOf('.')>-1){
        let sub = _f.split('.');
        let nsub = sub.shift();
        if(Array.isArray(obj[nsub])){
            for(let s = 0; s < obj[nsub].length; s++){
                sub_test(obj[nsub][s], sub.join('.'), initObj, check);
            }
            return;
        }else{
            return sub_test(obj[nsub], sub.join('.'), initObj, check);
        }
    }
    for (let j = 0; j < s.length; j++) {
        let b = false;
        if((typeof obj[_f] == 'string' || typeof obj[_f] == 'number') &&
typeof s[j] == 'string' &&
1 ||
1)){
            (obj[_f].toString().toLowerCase().indexOf(s[j].toLowerCase())>-
s[j].toLowerCase().indexOf(obj[_f].toString().toLowerCase())>-
1)){
                if(!_check[check]) _arr.push(initObj);
                _check[check] = true;
                b = true;
                break;
            }
            if(b) break;
        }
    }
}
}

```

У підсумку потрібно зазначити важливий момент, що стосується використання допоміжних інструментів технології Angular. Мова йде саме про фільтри або «pipes» в оригінальній назві, які застосовуються для перетворення даних прямо в HTML-шаблоні. Крім існуючого ряду фільтрів, є можливість створення власних, що значно може прискорити процес розробки системи.

### 3.3 Тестування інформаційної системи

Після створення компонентів інформаційної системи, необхідно здійснити їх перевірку працездатності шляхом тестування. Цей крок є необхідним, оскільки може надати розробнику розуміння, що створюваний ним програмний продукт буде поводитися надійно в період експлуатації користувачем.

Дана інформаційна система побудована у вигляді вебзастосунку, й відповідно для такого типу додатків потрібно обрати правильний вид тестування. Види тестувань розподіляються на функціональні, які відповідають за перевірку зовнішньої поведінки системи та номінальні, що визначають або вимірюють показники роботи системи. В даному випадку було обрано функціональне тестування у форматі тест-кейсів модулів та інших складових системи, результат виконаного процесу продемонстрований у таблицях нижче.

Таблиця 3.1 – Тестування модуля авторизації

<b>Author:</b> Malayko A.S.	<b>Spec ID:</b> 1	<b>Priority:</b> 1	<b>Producer:</b> Malayko A.S.	<b>Developer:</b> Malayko A.S.
<b>OVERVIEW:</b> поточний тест-кейс перевіряє працездатність модуля авторизації				
<b>GLOBAL SETUP and ADDITIONAL INFO:</b> необхідно відкрити будь-який сучасний браузер і перейти за посиланням <a href="http://localhost:4200/">http://localhost:4200/</a>				
<b>TC ID/Priority</b>	<b>TCID1</b>	<b>1</b>		
<b>IDEA:</b> після процесу введення особистих даних у відповідні поля форми авторизації та натиску на кнопку «Увійти», відбудеться перехід на сторінку власного профіля користувача				
<b>Revision History</b>				
<b>Created on:</b> 11.05.2021 by Malayko A.S.		Новий тест-кейс		
<b>Modified on:</b> 12.05.2021 by Malayko A.S.		Кроки були змінені, щоб зробити тест-кейс більш зручним для виконання.		
<b>Execution part</b>				

PROCEDURE	EXPECTED RESULT
1. Ввести у поле біля мітки «Email:» електронну адресу – user1@webart.work	Користувача повинно переадресувати на сторінку власного профілю.
2. Ввести у поле «Пароль» значення «user1»	
3. Ввести у поле «Пароль» значення «user1»	
<b>Result</b>	SUCCESS

Таблиця 3.2 – Тестування модуля пошуку

<b>Author:</b> Malayko A.S.	<b>Spec ID:</b> 2	<b>Priority:</b> 2	<b>Producer:</b> Malayko A.S.	<b>Developer:</b> Malayko A.S.
<b>OVERVIEW:</b> поточний тест-кейс перевіряє працездатність модуля пошуку				
<b>GLOBAL SETUP and ADDITIONAL INFO:</b> необхідно відкрити будь-який сучасний браузер та перейти за посиланням <a href="http://localhost:4200/">http://localhost:4200/</a> . Далі пройти авторизацію та натиснути на елемент «Підручники» навігаційного меню, після чого ввести будь-який логічний запит у пошуковому вікні				
<b>TC ID/Priority</b>		<b>TCID2</b>	2	
<b>IDEA:</b> при введенні коректної інформації в пошукове вікно, користувачу на даній сторінці відобразиться відповідний результат у вигляді електронних підручників				
<b>Revision History</b>				
<b>Created on:</b> 11.05.2021 by Malayko A.S.			Новий тест-кейс	
<b>Modified on:</b> 12.05.2021 by Malayko A.S.			Кроки були змінені, щоб зробити тест-кейс більш зручним для виконання.	
<b>Execution part</b>				
<b>PROCEDURE</b>			<b>EXPECTED RESULT</b>	
1. Перейти на вкладку «Підручники» навігаційного меню			Користувача повинно переадресувати на сторінку власного профілю.	

2. Активувати пошукове вікно, шляхом натискання ЛКМ	
3. Ввести у поле коректне значення, яке може бути присутнім всередині ІС	
<b>Result</b>	SUCCESS

Таблиця 3.3 – Тестування ряду функцій CRUD

<b>Author:</b> Malayko A.S.	<b>Spec ID:</b> 3	<b>Priority:</b> 3	<b>Producer:</b> Malayko A.S.	<b>Developer:</b> Malayko A.S.
<b>OVERVIEW:</b> поточний тест-кейс перевіряє працездатність ряду функцій, який забезпечує виконання операцій CRUD на адміністративній стороні системи				
<b>GLOBAL SETUP and ADDITIONAL INFO:</b> потрібно перейти за посиланням <a href="http://localhost:4200/">http://localhost:4200/</a> , далі пройти авторизацію на адміністративну сторону шляхом введення відповідних даних. Після переадресації на профіль користувача натиснути на елемент «Підручники» навігаційного меню та виконати будь-яку маніпуляційну дію з існуючим набором даних				
<b>TC ID/Priority</b>		<b>TCID3</b>	3	
<b>IDEA:</b> після натиску на будь-який графічний елемент управління, повинна відобразитись відповідна зміна даних у системі				
<b>Revision History</b>				
<b>Created on:</b> 11.05.2021 by Malayko A.S.		Новий тест-кейс		
<b>Modified on:</b> 12.05.2021 by Malayko A.S.		Кроки були модифіковані для отримання більш зручного алгоритму виконання тест-кейсу.		
<b>Execution part</b>				
<b>PROCEDURE</b>		<b>EXPECTED RESULT</b>		
1. Перейти на вкладку «Підручники» на адміністративній частині вебзастосунку		Після натискання на графічні елементи управління, адміністратор повинен		

	побачити відповідний результат виконаної операції на поточній сторінці.
2. Натиснути на кнопку «Створити підручник»	
<b>Result</b>	SUCCESS

Після отримання результату успішного тестування складових інформаційної системи, її можна вважати готовою до використання. Тому виникає актуальність створення інструкції по експлуатації вебзастосунку.

### 3.4 Інструкція користувача

Для отримання можливості використання готового програмного продукту, необхідно мати встановлений пакет технологій та додатків, до якого входить Node.js, Angular, Git та відповідно браузер. Якщо ряд таких технологій встановлено на комп'ютері користувача, він може без особливих проблем розпочати використовувати інформаційну систему.

Незважаючи на той факт, що в системі присутні два типа користувачів, це не змінює загальний алгоритм запуску вебзастосунку. Першим кроком даного процесу є відкриття терміналів для серверної та клієнтської частин у відповідних каталогах проекту. У першому випадку термінал повинен бути відкритим у кореневому каталозі «DistanceLearning», далі всередині нього необхідно прописати команду «waw». У другому випадку потрібно перейти до каталогу «DistanceLearning/client» та запустити термінал, після його відкриття прописуємо команду «go» (рисунок 3.4). Після виконання вище описаних дій відбудеться процес запуску обох частин вебзастосунку, проте сам вебзастосунок не з'явиться одразу. Причиною цього є генерація пакетів додатку браузера, цей процес може займати від тридцяти секунд до п'яти хвилин, дивлячись скільки разів був здійснений запуск.

```

MINGW64:/d/work/DistanceLearning
Andrew@DESKTOP-7ACFKBP MINGW64 /d/work/DistanceLearning (dev)
$ waw
===== App has started =====
App listening on port 8080

MINGW64:/d/work/DistanceLearning/client
Andrew@DESKTOP-7ACFKBP MINGW64 /d/work/DistanceLearning/client (dev)
$ go
> app@0.0.0 start D:\work\DistanceLearning\client
> ng serve -o --proxy-config proxy.conf.json

- Generating browser application bundles...
✓ Browser application bundle generation complete.

Initial Chunk Files | Names
vendor.js           | vendor
polyfills.js       | polyfills
styles.css, styles.js | styles
main.js            | main
runtime.js         | runtime

| Initial Total | 5.62 MB
Lazy Chunk Files | Names

```

Рисунок 3.4 – Запуск вебдодатку через термінали Git

Після завершення процесу генерації пакетів додатку, при умові якщо браузер відкритий, автоматично з'являється сторінка з формою авторизації інформаційної системи (рисунок 3.5).

Рисунок 3.5 – Форма авторизації інформаційної системи

Спочатку буде описане використання програмного продукту учнем, тому в полі електронної пошти необхідно ввести значення «user1@webart.work», а в полі паролю значення «user1» та натиснути на кнопку входу. Далі учень потрапить на сторінку власного профілю, на якій зможе заповнити інформацію про себе, ці дані можуть відображатись на певних сторінках інформаційної системи (рисунок 3.6).

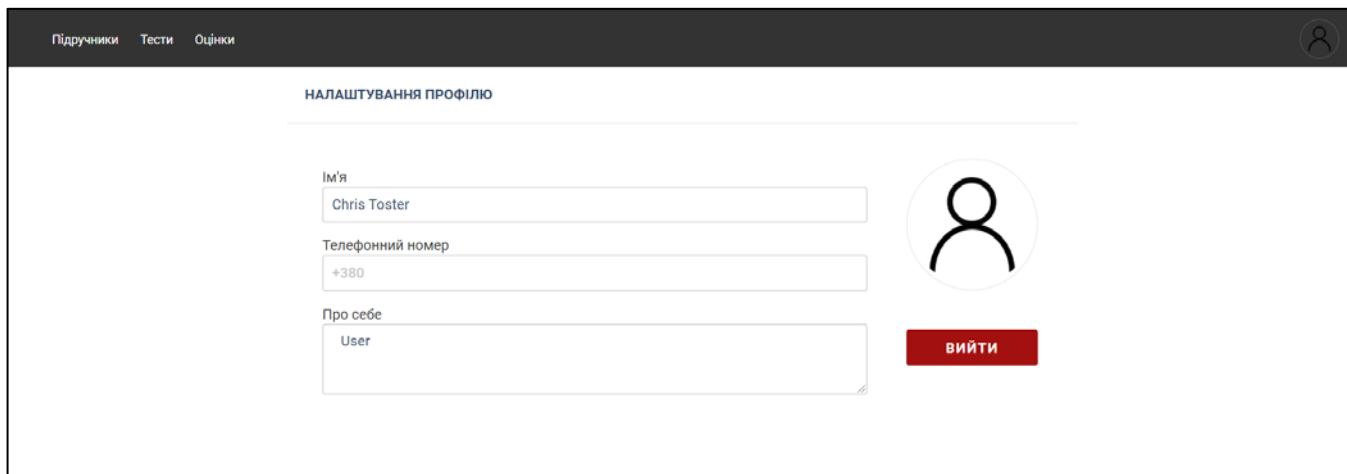


Рисунок 3.6 – Сторінка профілю користувача

Після заповнення власного профілю, учень може перейти до головної інформативної частини додатку, а саме до електронних підручників, натиснувши у верхньому навігаційному меню на відповідне посилання (рисунок 3.7).

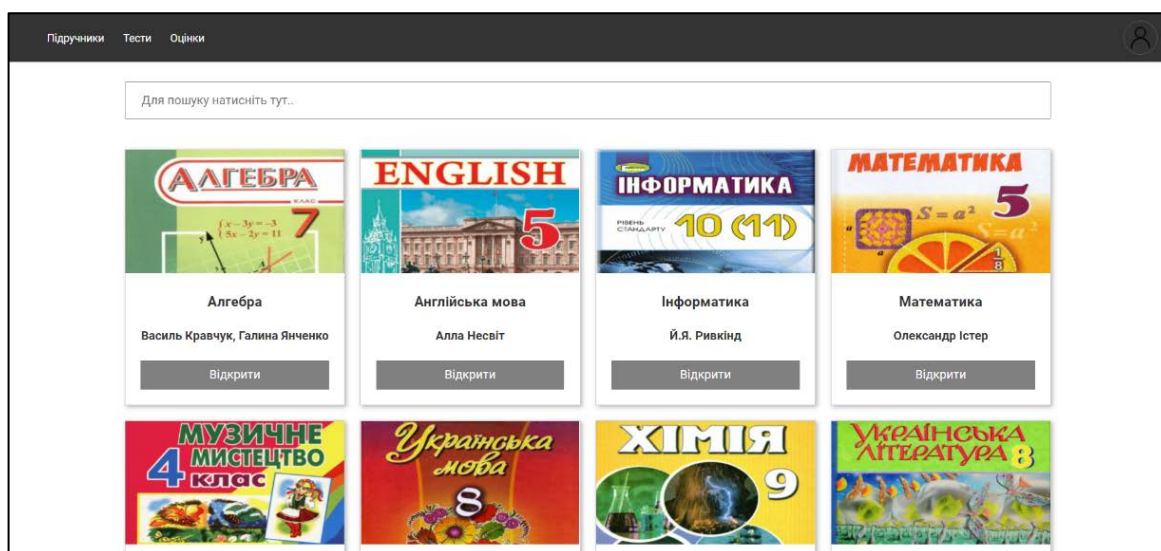


Рисунок 3.7 – Сторінка з електронними підручниками

На сторінці освітнього матеріалу, учень може скористатись глобальним пошуком для знаходження необхідних електронних підручників. У нього також присутня можливість перегляду вмісту підручника після натиску на кнопку «Відкрити». У випадку виникнення необхідності пошуку конкретної частини, він може скористатись вбудованою навігацією (рисунок 3.8).

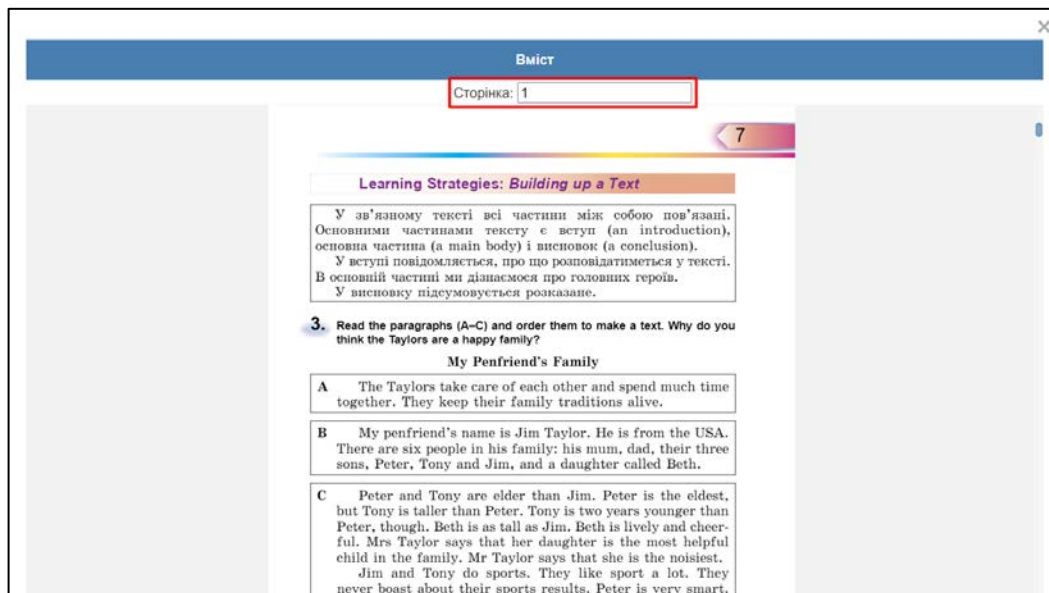


Рисунок 3.8 – Вміст електронного підручника

Після прочитання певної частини електронного підручника, учень може перевірити засвоєнні знання пройшовши відповідний тест. Щоб це здійснити, потрібно натиснути на елемент навігаційного меню «Тести», далі відбудеться переадресація на сторінку з існуючими тестами інформаційної системи (рисунок 3.9).

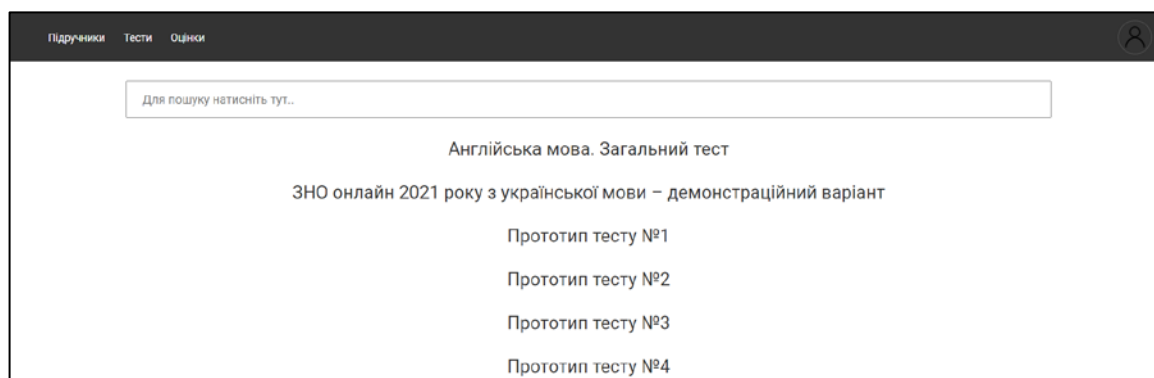


Рисунок 3.9 – Сторінка з тестами

Після обрання необхідного тесту, відкривається сторінка з його вмістом, до якого входять питання та відповіді. Якщо учень прочитав питання та знайшов правильну відповідь на його думку, він повинен обрати її шляхом натискання на відповідну радіокнопку. Після опрацювання всіх питань користувач закінчує тест натискаючи на кнопку «Завершити» (рисунок 3.10).

The screenshot shows a test interface with a dark header containing 'Підручники', 'Тести', and 'Оцінки'. The main content area contains four questions, each with three radio button options:

- Question 1: Options are 'b. complain' and 'c. argue'.
- Question 2: 'I really don't like this meal. ... money in the world wouldn't get me to eat it.' Options: 'a. Whatever', 'b. Enough', 'c. All the' (selected).
- Question 3: 'Last year, Joanna bought two ... coats in New York.' Options: 'a. long, black, leather' (selected), 'b. black, long, leather', 'c. leather, black, long'.
- Question 4: 'I must report to the meeting that Cyrus completed his first piece of work well ahead of schedule. ..., however, his work has been handed in late.' Options: 'a. Sequentially', 'b. Subsequently', 'c. Consequesntly' (selected).

A blue button labeled 'Завершити' is located at the bottom of the test area.

Рисунок 3.10 – Сторінка вмісту тесту

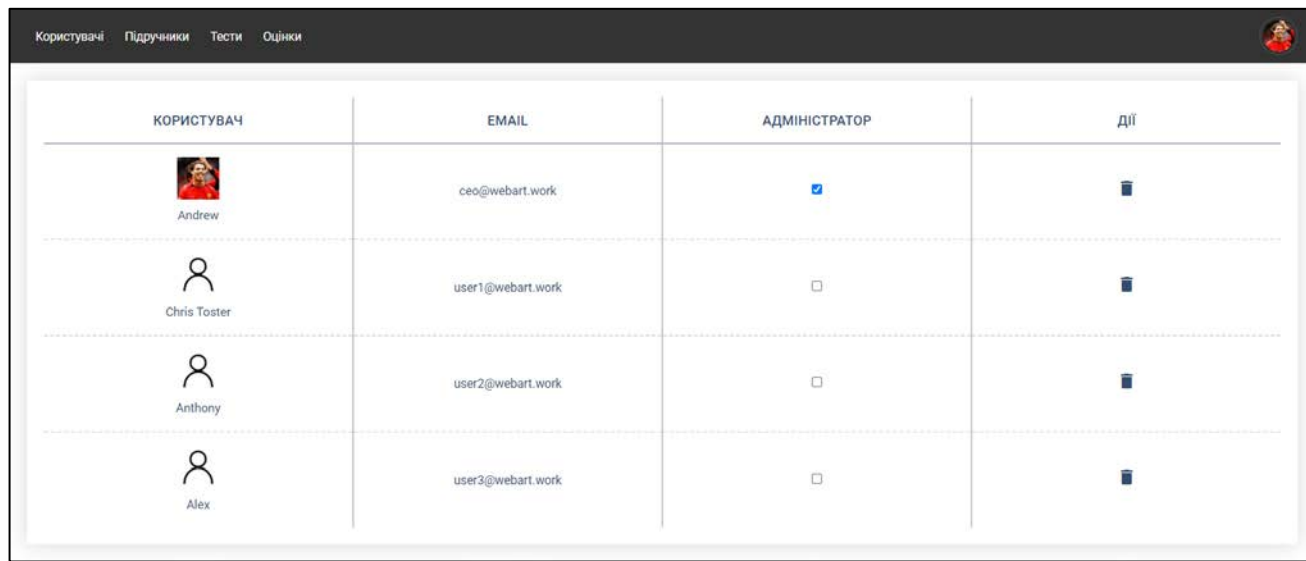
Після завершення тесту, система направляє учня на сторінку власних оцінок (рисунок 3.11).

The screenshot shows a test results page with a dark header containing 'Підручники', 'Тести', and 'Оцінки'. Below the header is a search bar with the placeholder text 'Для пошуку натисніть тут..'. Below the search bar is a table with the following data:

Тест	Дата	Оцінка
Англійська мова. Загальний тест	6/4/21, 1:26 AM	10
ЗНО онлайн 2021 року з української мови – демонстраційний варіант	5/28/21, 4:42 PM	12

Рисунок 3.11 – Сторінка результатів проходження тестів

Щоб використовувати інформаційну систему на адміністративній сторінці, потрібно у формі авторизації ввести електронну пошту «ceo@webart.work» та пароль «asdasdasdasd». Далі адміністратор потрапляє на сторінку власного профілю аналогічно з випадком учня. На даній сторінці він може переглянути, а також видалити користувачів системи (рисунок 3.12).











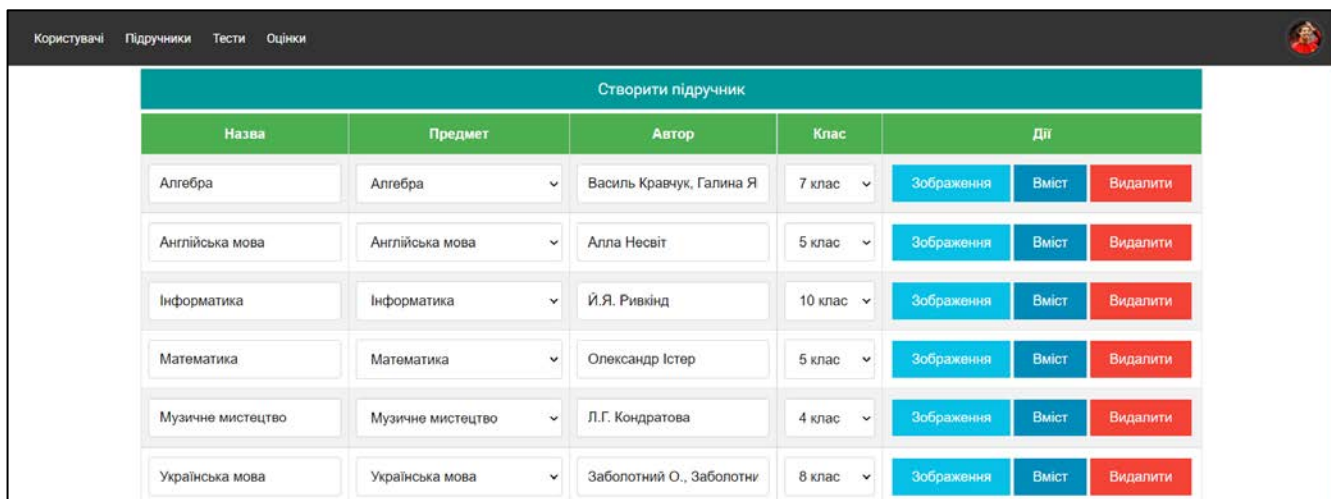
КОРИСТУВАЧ	EMAIL	АДМІНІСТРАТОР	Дії
 Andrew	ceo@webart.work	<input checked="" type="checkbox"/>	
 Chris Toster	user1@webart.work	<input type="checkbox"/>	
 Anthony	user2@webart.work	<input type="checkbox"/>	
 Alex	user3@webart.work	<input type="checkbox"/>	

Рисунок 3.12 – Сторінка з користувачами системи

До функціоналу адміністратора належить і виконання операцій CRUD з електронними підручниками (рисунок 3.13).





















Створити підручник				
Назва	Предмет	Автор	Клас	Дії
Алгебра	Алгебра	Василь Кравчук, Галина Я	7 клас	  
Англійська мова	Англійська мова	Алла Несвіт	5 клас	  
Інформатика	Інформатика	Й.Я. Ривкінд	10 клас	  
Математика	Математика	Олександр Істер	5 клас	  
Музичне мистецтво	Музичне мистецтво	Л.Г. Кондратова	4 клас	  
Українська мова	Українська мова	Заболотний О., Заболотни	8 клас	  

Рисунок 3.13 – Сторінка електронних підручників адміністратора

Згідно з рисунком 3.13, адміністратор здатен написати назву підручника та його авторів, обрати предмет та клас для класифікації. Проте даних дій не достатньо для надання освітнього матеріалу учням в оптимальному форматі. Рішенням даної проблеми є відвантаження конкретних файлів за допомогою відповідних функцій, що знаходяться в модальному вікні, яке можна відкрити натискаючи на кнопку «Вміст» (рисунок 3.14).

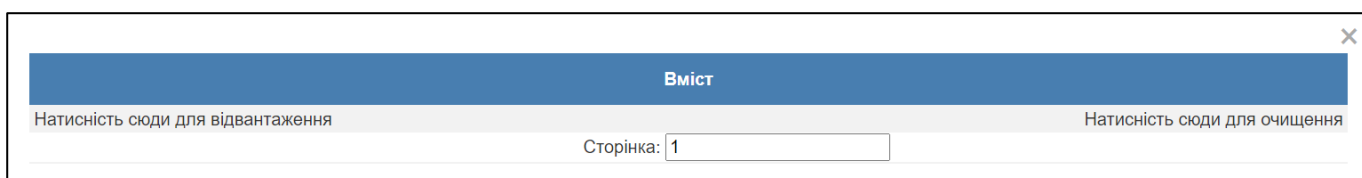


Рисунок 3.14 – Модальне вікно для завантаження вмісту підручника

Дивлячись на рисунок 3.14 можна помітити присутність функцій для відвантаження файлу, а також його очищення. Щоб адміністратор був впевнений у відповідності вмісту файлу згідно з його назвою, існує навігація для перегляду сторінок підручника. При натиску на функцію відвантаження відкриється каталог для обрання файлу, обраний файл одразу відобразиться у модальному вікні (рисунок 3.15).

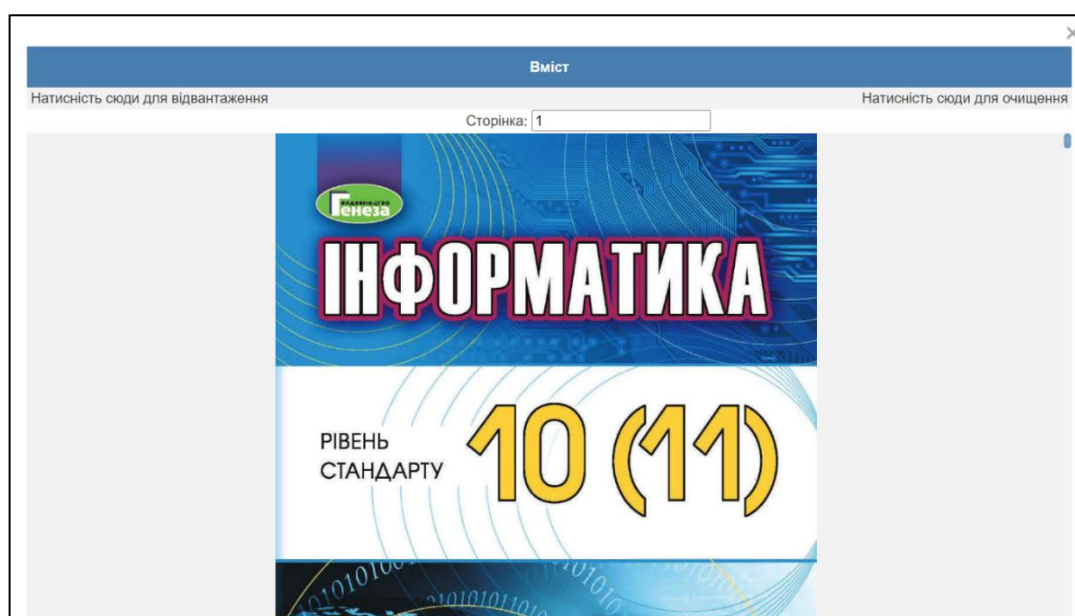


Рисунок 3.15 – Модальне вікно з відвантаженим файлом

В свою чергу натискання на функцію очищення, приведе модальне вікно у початковий стан. У підсумку даного етапу виконання маніпуляційних дій з електронними підручниками та їх складовими, адміністратор отримує загальну інформацію про підручник та його вміст. Бракує лише однієї деталі, а саме титульного зображення, яке може бути додане аналогічним чином, як у випадку вмісту, потрібно лише перейти у відповідне модальне вікно натискаючи на потрібну кнопку.

Після формування підручників, слід створити певний ряд тестів, щоб учні мали можливість перевірити засвоєні знання. Для цього адміністратор повинен перейти на сторінку «Тести» й натиснути на кнопку «Створити тест» (рисунок 3.16).

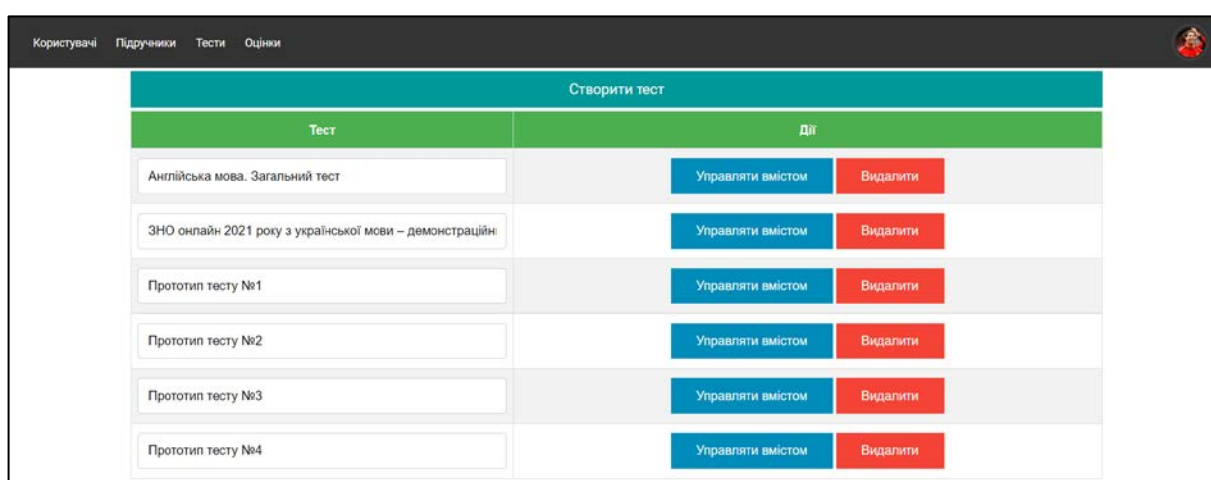


Рисунок 3.16 – Сторінка управління тестами

Рисунок вище демонструє, що адміністратор має змогу створювати і видаляти тест, а також задавати йому ім'я. Щоб додати вміст тесту, необхідно натиснути на кнопку «Управляти вмістом», після чого з'явиться нова сторінка з можливістю додавання питань та відповідей (рисунок 3.17).

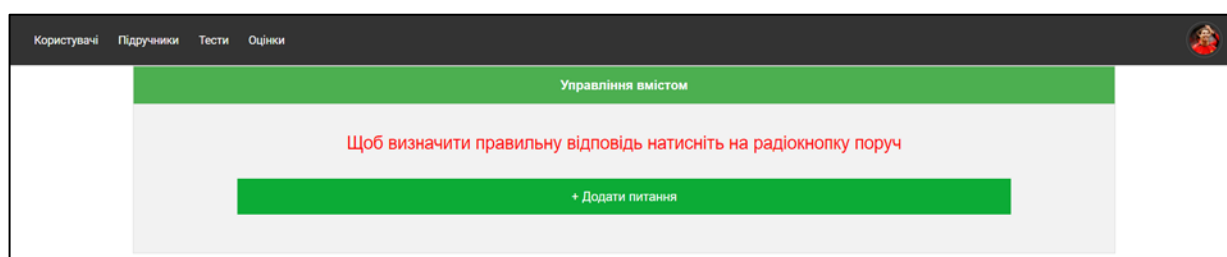


Рисунок 3.17 – Сторінка додавання вмісту теста

Питання тесту додаються натиском на відповідну кнопку, після чого з'являється функціонал для додавання і видалення відповідей даного питання (рисунок 3.18).

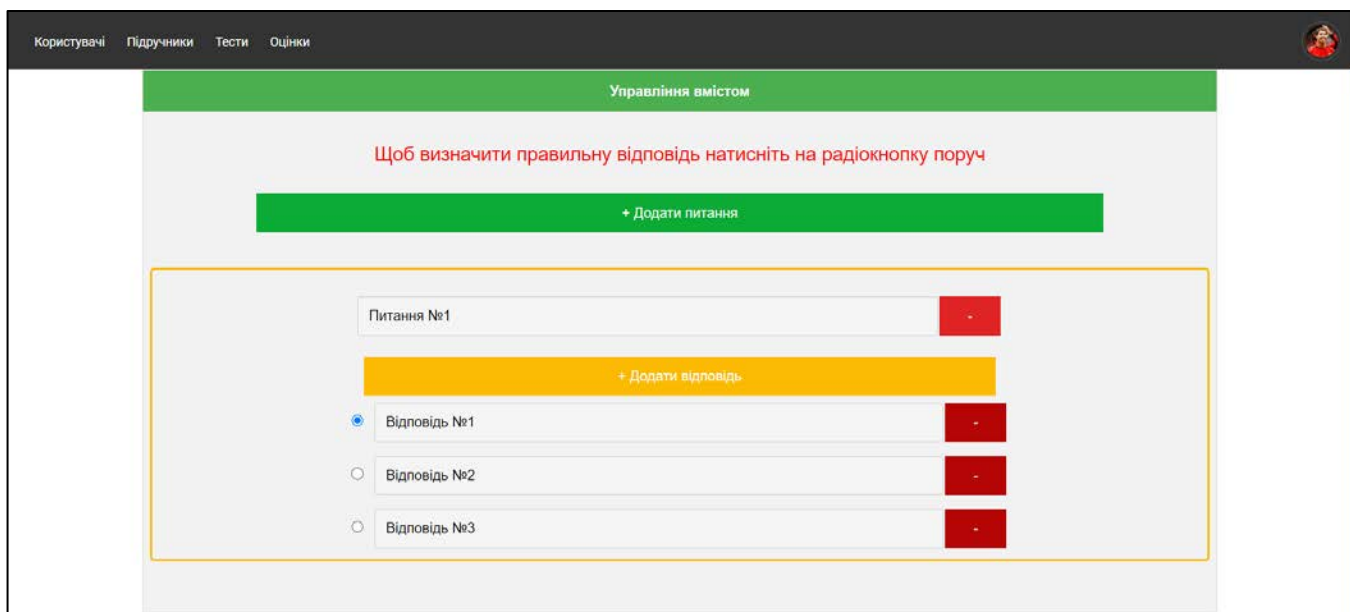


Рисунок 3.18 – Функціонал додавання й видалення питань та їх відповідей

У адміністратора також присутня можливість редагування оцінок учнів за пройдені тести на сторінці «Оцінки» (рисунок 3.19).

Тест	Дата	Учень	Оцінка	Дії
Англійська мова. Загальний тест	6/4/21, 1:26 AM	Chris Toster	10	Видалити
ЗНО онлайн 2021 року з української мови – демонстраційний варіант	5/28/21, 4:42 PM	Chris Toster	12	Видалити
Англійська мова. Загальний тест	5/31/21, 4:54 PM	Anthony	3	Видалити
Англійська мова. Загальний тест	6/2/21, 12:51 AM	Alex	1	Видалити

Рисунок 3.19 – Сторінка редагування отриманих оцінок учнів

Рисунок 3.19 демонструє наявність доступу в адміністратора до дати проходження тесту користувачем, його результату та назви пройденого тесту. Це може бути корисним при редагуванні оцінки, коли учень з нею не погоджується через певний ряд причин.

### **3.5 Вимоги до розгортання інформаційної системи**

Для розгортання інформаційної системи, будь-який персональний комп'ютер повинен володіти певним рядом характеристик, а також містити необхідний набір встановлених додатків. Оскільки, інформаційна система містить клієнтську та серверну частини, то є необхідним визначення мінімальних або рекомендованих вимог для кожної з частин.

До рекомендованих апаратних вимог запуску сервера належить:

- ОС Windows 10;
- 4 гігабайти RAM;
- 10 гігабайт вільного простору на жорсткому диску.

Рекомендованими апаратними вимогами для запуску клієнтської частини є:

- ОС Windows 7 або пізніша версія, у випадку macOS 10.5 та новіша;
- 3 гігабайти RAM;
- 10 гігабайтів вільного простору на жорсткому диску.

У випадку програмних засобів, рекомендується мати встановленим:

- Node.js версії 14.15.3 та пізніше;
- Git починаючи з версії 2.29;
- MongoDB версії 4.4.2 та пізніше;
- Angular CLI 11.0.5 й вище.

## Висновки

Метою кваліфікаційної роботи бакалавра була розробка інформаційної системи для забезпечення дистанційної форми освіти учнів загальноосвітніх навчальних закладів. Під час виконання даної роботи, був проведений аналіз предметної області та існуючого програмного забезпечення для отримання знань побудови програмних продуктів, які можуть нести актуальність в області освіти. Також був проведений аналіз сучасних засобів створення програмних продуктів із зазначенням їх слабких та сильних сторін, що у підсумку привело до обрання найбільш оптимального варіанту.

Розроблена інформаційна система повинна бути наділена рядом функцій, що забезпечує реєстрацію та авторизацію її користувачам, також на адміністративній стороні повинна бути присутня можливість редагування вмісту бази даних. Крім цього система повинна надавати учням навчальний матеріал у вигляді електронних підручників з можливістю перевірки засвоєних знань шляхом проходження прикріплених тестів. Відповідно після процесу проходження тесту повинен відображатись результат у вигляді оцінки. Також в учня повинна бути присутня можливість швидкого пошуку як необхідного електронного підручника так і його конкретної частини.

В результаті виконаної роботи був побудований вебзастосунок, який виконує всі поставлені перед собою задачі. Отримана програма пройшла процес тестування задля визначення її слабких сторін та їх відповідного позбуття. У сукупності виконання ряду вище описаних дій вказує на набуття студентом нових навичок, а також використання раніше отриманих. Підсумовуючи можна сказати, що проведена робота переформувала студента у фахівця області інформаційних технологій.

## Перелік посилань

1. Роль інформаційних технологій у сучасному світі. URL: [http://dSPACE.mnau.edu.ua/jspui/bitstream/123456789/7350/1/zbirnyk\\_tez\\_15-17\\_05\\_19\\_46.pdf](http://dSPACE.mnau.edu.ua/jspui/bitstream/123456789/7350/1/zbirnyk_tez_15-17_05_19_46.pdf)
2. Освіта – Вікіпедія. URL: <https://cutt.ly/MnvoSPG>
3. Загальна освіта – Вікіпедія. URL: <https://cutt.ly/ZnvoGND>
4. Загальна середня освіта. URL: <http://education-ua.org/zagalna-serednya-osvita.php>
5. Загальна середня освіта в Україні – Вікіпедія. URL: <https://cutt.ly/hnvoKpD>
6. Дистанційна школа Оптіма – Перша онлайн школа в Україні. URL: <https://optima.school/>
7. НВК №2. URL: <https://nvk2.khnu.km.ua/page.aspx>
8. Топ фреймворки для веброзробки в 2018 році – JetRuby. URL: <https://jetruby.com/ru/blog/top-freimworki-2018/>
9. React або Angular або Vue.js – що обрати? / Хабр. URL: <https://habr.com/ru/post/476312/>
10. Що краще React чи Angular? URL: <https://artjoker.ua/ru/blog/chto-luchshe-react-ili-angular/>
11. Недоліки Ruby on Rails? – Хабр Q&A. URL: <https://qna.habr.com/q/197211>
12. Обираємо Yii2 або laravel / Хабр. URL: <https://habr.com/ru/post/353434/>
13. PHP фреймворк Laravel. Огляд, плюси, мінуси, відгуки. URL: <https://unetway.com/blog/laravel-framework-review>
14. ASP.NET – Вікіпедія. URL: <https://uk.wikipedia.org/wiki/ASP.NET>
15. The significant weaknesses of ASP.Net – Hire dot net developers. URL: <https://sites.google.com/site/hiredotnetdevelopers/the-significant-weaknesses-of-asp-net>
16. PHP vs. Node.JS – BairesDev. URL: <https://www.bairesdev.com/blog/php-vs-node-js/>
17. Advantages and Disadvantages of PHP – GeeksForGeeks. URL: <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-php/>

18. Node.js – Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Node.js>
19. Advantages & Disadvantages of Node.js : Why to use Node.js? URL: <https://www.simform.com/nodejs-advantages-disadvantages/>
20. Advantages and Disadvantages of JavaScript. URL: <https://www.tutorialspoint.com/advantages-and-disadvantages-of-javascript>
21. Why Use MongoDB & When to Use It? | MongoDB. URL: <https://www.mongodb.com/why-use-mongodb>
22. All about MongoDB NoSQL Database: Advantages and Disadvantages. URL: [https://acodez.in/mongodb-nosql-database/#Example\\_of\\_MongoDB\\_Operations](https://acodez.in/mongodb-nosql-database/#Example_of_MongoDB_Operations)
23. PostgreSQL – Вікіпедія. URL: <https://cutt.ly/3nvo91F>
24. What is PostgreSQL? – Amazon Web Services. URL: <https://aws.amazon.com/rds/postgresql/what-is-postgresql/>
25. What is PostgreSQL? | Features | Advantages and Disadvantages. URL: <https://www.educba.com/what-is-postgresql/>
26. Sublime Text – Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Sublime\\_Text](https://uk.wikipedia.org/wiki/Sublime_Text)
27. Visual Studio Code – Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://uk.wikipedia.org/wiki/Visual_Studio_Code)

# ДОДАТКИ

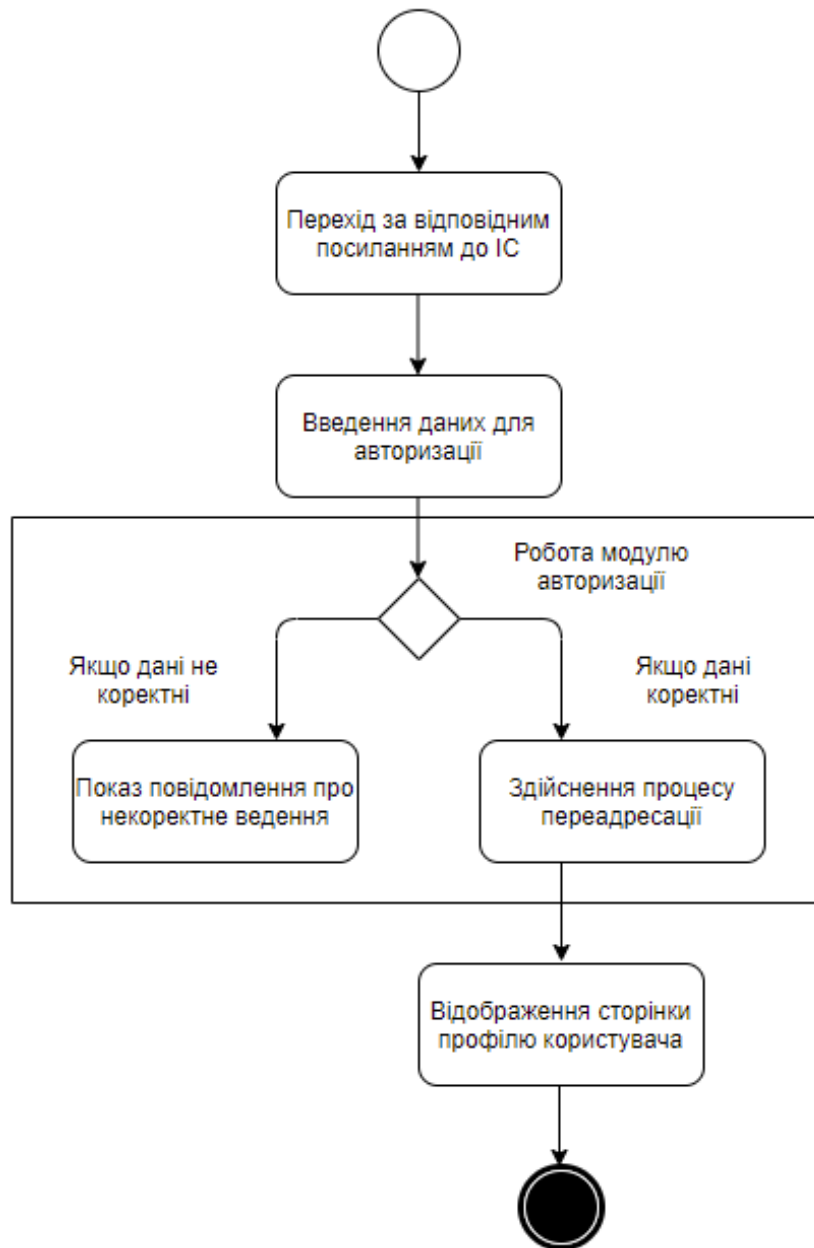
## Додаток А

### Діаграма варіантів використання інформаційної системи



## Додаток Б

### Діаграма діяльності модуля авторизації



## Додаток В

### Програмні коди

#### Лістинг login.component.html:

```
<div class="auth-wrapper">
  <div class="auth">
    <div class="auth-title">Вхід</div>
    <form class="auth-form"
      (ngSubmit)="login()">
      <div class="waw-input mb15">
        <span>Email:</span>
        <input
          (keydown.enter)="login()"
          placeholder="Email"
          [(ngModel)]="user.email">
      </div>
      <div class="waw-input mb15">
        <span>Пароль:</span>
        <input
          (keydown.enter)="login()"
          type="password"
          placeholder="Пароль"
          [(ngModel)]="user.password">
      </div>
      <div class="auth-form_btn"><button class="waw-btn _primary"
        type="submit">Увійти</button></div>
    </form>
    <div class="auth-link">
      <a routerLink="/reset">Забули
        пароль?</a>
      <a routerLink="/sign">Немає
        акаунту?</a>
    </div>
  </div>
</div>
```

#### Лістинг login.component.html:

```
import { Component } from '@angular/core';
import { Router } from '@angular/router';
import { HashService, HttpService, AlertService }
from 'wacom';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.scss']
})
export class LoginComponent {
  constructor(private router: Router,
    private hash: HashService,
    private http: HttpService,
    private alert: AlertService) {
    this.user.email =
      this.hash.get('email')||'ceo@webart.work';
    this.user.password =
      this.hash.get('password')||'asdasdasdasd';
    if(this.hash.get('login'))
      this.login();
  }
  public user:any = {};
  login() {
    console.log(this.alert.show);
```

```
placeholder="Пароль"
name="password"
[(ngModel)]="user.password">
</div>
<div class="auth-form_btn"><button class="waw-btn _primary"
  type="submit">Увійти</button></div>
</form>
<div class="auth-link">
  <a routerLink="/reset">Забули
    пароль?</a>
  <a routerLink="/sign">Немає
    акаунту?</a>
</div>
</div>
</div>
if(!this.user.email) {
  return this.alert.show({
    type: "error",
    text: 'Enter your
    email',
    position: "center"
  });
}
this.hash.set('email',
this.user.email);
if(!this.user.password) {
  return this.alert.show({
    type: "error",
    text: 'Enter your
    password'
  });
}
this.http.post('/api/user/status',
this.user, (resp:any) => {
  if(resp.email && resp.pass) {
    this.http.post('/api/user/login', this.user,
    (user:any) => {
      if(!user){
        return this.alert.show({
          type: "error",
```



```

        constructor(public us: UserService, public
ui: UiService) {}
        public new_user:any={};
        delete(){
            for(let i = this.us.users.length-1;
i>=0; i--){

                if(this.us.users[i].generated){

                    this.us.delete(this.us.users[i]);
                        }
                    }
                }
            }
        }
    }
}

```

## Лістинг textbooks.component.html:

```

<div class="main">
    <button (click)="tbs.create()" class="btn-
create">Створити підручник</button>
    <table id="textbooks">
        <tr>
            <th>Назва</th>
            <th>Предмет</th>
            <th>Автор</th>
            <th>Клас</th>
            <th>Дії</th>
        </tr>
        <tr *ngFor="let textbook of
tbs.textbooks | search: search: 'name tb_author'">
            <td>
                <input type="text"
[(ngModel)]="textbook.name"
(ngModelChange)="tbs.update(textbook)" id="name">
            </td>
            <td>
                <select id="subject"
name="subject" [(ngModel)]="textbook.subject"
(ngModelChange)="tbs.update(textbook)">
                    <option
value="Алгебра">Алгебра</option>
                    <option
value="Англійська мова">Англійська мова</option>
                    <option
value="Геометрія">Геометрія</option>
                    <option
value="Інформатика">Інформатика</option>
                    <option
value="Математика">Математика</option>

```

```

                    <option
value="Музичне мистецтво">Музичне мистецтво</option>
                    <option
value="Образотворче мистецтво">Образотворче
мистецтво</option>
                    <option
value="Трудове навчання">Трудове навчання</option>
                    <option
value="Хімія">Хімія</option>
                    <option
value="Українська література">Українська
література</option>
                    <option
value="Українська мова">Українська мова</option>
                    <option
value="Фізика">Фізика</option>
                </select>
            </td>
            <td>
                <input type="text"
[(ngModel)]="textbook.tb_author"
(ngModelChange)="tbs.update(textbook)" id="author">
            </td>
            <td>
                <select id="class"
name="class" [(ngModel)]="textbook.class"
(ngModelChange)="tbs.update(textbook)">
                    <option
value="4">4 клас</option>
                    <option
value="5">5 клас</option>
                    <option
value="6">6 клас</option>
                    <option
value="7">7 клас</option>
                    <option
value="8">8 клас</option>
                    <option
value="9">9 клас</option>
                    <option
value="10">10 клас</option>
                    <option
value="11">11 клас</option>
                </select>
            </td>
            <td>
                <button class="btn-
image" (click)="tbs.textbook=textbook;
modal.show({component: 'image'})">Зображення</button>
                <button class="btn-
content" (click)="tbs.textbook=textbook;
modal.show({component: 'textbook'})">Вміст</button>

```

```

                <button class="btn-
delete"
(click)="tbs.delete(textbook)">Видалити</button>
                </td>
            </tr>
        </table>
    </div>

```

### Лістинг tests.component.html:

```

<div class="main">
    <button (click)="ts.create()" class="btn-
create">Створити тест</button>
    <table id="tests">
        <tr>
            <th>Тест</th>
            <th>Дії</th>
        </tr>
        <tr *ngFor="let test of ts.tests">
            <td>
                <input type="text"
                [(ngModel)]="test.name"
                (ngModelChange)="ts.update(test)" id="name">
            </td>
            <td>
                <button class="btn-
content"
(click)="goToTestContent(test._id)">Управляти
вмістом</button>
                <button class="btn-
delete" (click)="ts.delete(test)">Видалити</button>
            </td>
        </tr>
    </table>
</div>

```

### Лістинг tests.component.ts:

```

import { Component } from '@angular/core';
import { TestService } from '@services';
import { Router } from '@angular/router';
@Component({
    selector: 'tests',
    templateUrl: './tests.component.html',
    styleUrls: ['./tests.component.scss']
})
export class TestsComponent{
    constructor(public ts: TestService, public r:
Router) { }

```

```

        goToTestContent(id): void {
            this.r.navigate(['/admin/test/' +
id]);
        }
    }
}

```

### Лістинг test.component.html:

```

<div class="main">
    <table id="content">
        <tr>
            <th>Управління вмістом</th>
        </tr>
        <tr>
            <td>
                <div class="note">Щоб
визначити правильну відповідь натисніть на
радіокнопку поруч</div>
                <button
(click)="ts.createQuestion(ts._tests[id])"
class="btn-createQuestion">+ Додати питання</button>
                <div *ngFor="let
question of ts._tests[id].question; index as i"
class="questionsBlock">
                    <input
type="text" [(ngModel)]="question.name"
(ngModelChange)="ts.update(ts._tests[id])"
class="questionName">
                    <button
(click)="ts._tests[id].question.splice(i,1);
ts.update(ts._tests[id])" class="btn-
deleteQuestion">-</button>
                    <button
(click)="ts.createAnswer(question, ts._tests[id])"
class="btn-createAnswer">+ Додати відповідь</button>
                </div>
                <div
*ngFor="let answer of question.answer; index as j">
                    <input type="radio" [name]='correct'+i"
[value]="j" class="rightAnswer"
[(ngModel)]="answer.correct"
(ngModelChange)="ts.update(ts._tests[id])">
                    <input type="text" class="answerName"

```

```

[(ngModel)]="answer.name"
(ngModelChange)="ts.update(ts._tests[id])">
    <td>{{mark.test}}</td>
    <td>{{mark.date|date:
'short'}}</td>
    <td>{{user.name}}</td>
    <td>
        <input
type="text" [(ngModel)]="mark.value"
(ngModelChange)="us.save(user)" class="mark">
    </td>
    <td>
        <button
class="btn-delete" (click)="user.mark.splice(i,1);
us.save(user)">Видалити</button>
    </td>
    </tr>
</ng-container>
</table>
</div>

```

## Лістинг test.component.ts:

```

import { Component } from '@angular/core';
import { TestService } from '@services';
import { Router } from '@angular/router';
@Component({
  selector: 'test',
  templateUrl: './test.component.html',
  styleUrls: ['./test.component.scss']
})
export class TestComponent{
  constructor(public ts: TestService, public r:
Router) { }

  public id =
this.r.url.toLowerCase().replace('/admin/test/', '');
}

```

## Лістинг marks.component.html:

```

<div class="main">
  <input type="text" placeholder="Для пошуку
натисніть тут.." name="search" [(ngModel)]="search"
class="search">

  <table id="marks">
    <tr>
      <th>Тест</th>
      <th>Дата</th>
      <th>Учень</th>
      <th>Оцінка</th>
      <th>Дії</th>
    </tr>
    <ng-container *ngFor="let user of
us.users | search: search: 'name'">
      <tr *ngFor="let mark of
user.mark; index as i">

```

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

## ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ

Направляється студент Малайко А. С. на захист дипломного проекту (роботи)  
(прізвище, ініціали)

за спеціальністю 122 - Комп'ютерні науки

На тему: Інформаційна система для забезпечення дистанційної форми освіти учнів загальноосвітніх навчальних закладів

Дипломний проект (робота), рецензія і довідка про перевірку на плагіат додаються.

Декан факультету



САВЕНКО О. С.

(прізвище та ініціали)

### ДОВІДКА УСПІШНОСТІ

Малайко А. С. за період навчання на факультеті програмування та комп'ютерних і телекомунікаційних систем з 2017 по 2021 роки повністю виконав навчальний план спеціальності з такими розподілом оцінок за:

національною шкалою: відмінно 37,50 %, добре 56,25 %, задовільно 6,25 %.

шкалою ЄКТС: А 32,73 %, В 27,27 %, С 30,91 %, D 5,45 %, E 3,64 %.

Методист факультету

[Signature]  
(підпис)

(прізвище та ініціали)

### ВИСНОВОК КЕРІВНИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ) ТА ОБГРУНТУВАННЯ ОЦІНКИ

Студент Малайко А. С. виконав кваліфікаційну роботу «Інформаційна система для забезпечення дистанційної форми освіти учнів загальноосвітніх навчальних закладів» у відповідності до завдання в ньому обсязі

Оцінка дипломного проекту (роботи) відмінно

Керівник дипломного проекту (роботи)

[Signature]  
(підпис)

Петровський  
(прізвище та ініціали)

" 01 " 06 2021 р.

### ВИСНОВОК КАФЕДРИ ПРО ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Дипломний проект (роботу) розглянуто. Студент Карлечук Д. Т. допускається до захисту цього

Завідувач кафедри

КНІТ

(назва)

" 09 " 06 2021 р.

[Signature]  
(підпис, прізвище, ініціали)

## Anti-Plagiarism v-15.257

**Максимальное совпадение с одним документом 6.0%**

Словари проверки: en\_US, ru\_RU, ua\_UA. **Ошибок в документах: 8%**

ID: 92666 Название: Інформаційна система для забезпечення дистанційної форми освіти учнів загальноосвітніх навчальних закладів Добавлено в БД: 2021-06-08 Авторы: А.С. Малайко Руководители: С.С. Петровський Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	71035	643	5269 (7%)	60 (9%)

### Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

Ім'я користувача:  
Кафедра КН

ID перевірки:  
1008226832

Дата перевірки:  
08.06.2021 12:17:28 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
08.06.2021 12:18:47 EEST

ID користувача:  
100005671

Назва документа: КРБ\_файл для плагіату Lite

Кількість сторінок: 65 Кількість слів: 10623 Кількість символів: 79434 Розмір файлу: 8.92 MB ID файлу: 1008300611

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

**1.89%**  
**Схожість**

Найбільша схожість: 1.2% з джерелом з Бібліотеки (ID файлу: 1008298363)

0.83% Джерела з Інтернету

53

Сторінка 67

1.47% Джерела з Бібліотеки

39

Сторінка 67

**0% Цитат**

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

**0%**  
**Вилучень**

Немає вилучених джерел

**Модифікації**

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування

13  
сторінок

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ**  
**КАФЕДРИ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**  
**ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Інформаційна система для забезпечення дистанційної форми освіти учнів загальноосвітніх навчальних закладів

Автор: студент групи КН-17-1 Малайко Андрій Сергійович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: доцент кафедри КНІТ Петровський С.С.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<i>відповідає</i>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;

2) усі запозичення фрагментарні;

3) до запозичень входять фрагменти програмного коду, що на мають авторства і містять поширені конструкції;

4) серед запозичень знаходяться загальновідомі терміни, скорочення та визначення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 1,9 % і адресується до першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

  
\_\_\_\_\_

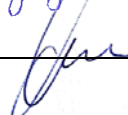
С.С. Петровський

Гарант ОП

  
\_\_\_\_\_

О. В. Мазурець

Завідувач кафедри КНІТ

  
\_\_\_\_\_

О. В. Бармак

## РЕЦЕНЗІЯ

### на кваліфікаційну роботу бакалавра

Студента: групи КН-17-1 Малайка Андрія Сергійовича

За темою: Інформаційна система для забезпечення дистанційної форми освіти учнів загальноосвітніх навчальних закладів.

1. Актуальність і значення теми: розроблена система призначена для рішення проблем, що пов'язані з відсутністю очної форми отримання загальної середньої освіти дітьми шкільного віку. Створений вебзастосунок надає змогу учням, отримувати нові знання читаючи електронні підручники, а також перевіряти засвоєний матеріал шляхом проходження відповідних тестів з відображенням результатів.

2. Оцінка запропонованих моделей, підходів, алгоритмів, інформаційної складової та засобів розробки: використані підходи та ідеї реалізації модулів системи та інших її компонентів забезпечують виконання поставлених завдань.

3. Оцінка розробленої інформаційної системи, її практична цінність та економічна доцільність: розроблена інформаційна система є саме тим рішенням, яке може надати учням навчально-виховних комплексів можливість здобувати освіту дистанційно.

4. Загальний висновок: в результаті виконаної роботи був побудований вебзастосунок, який виконує всі поставлені перед собою задачі. Отримана програма пройшла процес тестування задля визначення її слабких сторін та їх відповідного позбуття.

Робота заслуговує на оцінку « Відмінно »

Рецензент к. ер.-м. н. доц. Яценюк Н. О. Друж 08.06.2021 р.