

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВОЛИНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ЛЕСІ УКРАЇНКИ**

Кафедра загальної математики та методики навчання інформатики

На правах рукопису

ГАЛУШКА ЮРІЙ АНАТОЛІЙОВИЧ

**ПРОЄКТУВАННЯ ТА РОЗРОБКА ОСВІТНЬОЇ ВЕБПЛАТФОРМИ ДЛЯ
УПРАВЛІННЯ НАВЧАЛЬНИМ КОНТЕНТОМ ТА КОМУНІКАЦІЇ**

Спеціальність: 014 Середня освіта (Інформатика)
Освітньо-професійна програма «Середня освіта. Інформатика»
Робота на здобуття освітнього ступеня «Магістр»

Науковий керівник:
РОЙКО ЛАРИСА ЛЕОНІДІВНА,
кандидат педагогічних наук, доцент
кафедри загальної математики та
методики навчання інформатики

РЕКОМЕНДОВАНО ДО ЗАХИСТУ

Протокол № _____
засідання кафедри загальної математики
та методи навчання інформатики
від _____ 2025 року
Завідувач кафедри:

(підпис)

ПІБ

ЛУЦЬК – 2025

Анотація

Галушка Ю. А. Проектування та розробка вебплатформи для управління навчальним контентом та комунікації – Рукопис

Кваліфікаційна робота за спеціальністю 014 Середня освіта (Інформатика).
– Волинський національний університет імені Лесі Українки. Луцьк, 2025.

У кваліфікаційній роботі проаналізовано проблему цифровізації освітнього процесу та створення сучасних вебплатформ для підтримки навчального процесу в умовах дистанційного та змішаного навчання. На основі аналізу визначено основні вимоги до розробки власної навчальної платформи, включаючи функціональні, нефункціональні, педагогічні та технічні аспекти.

Розроблено освітню вебплатформу SmartClass для управління навчальним контентом та комунікацією між учасниками освітнього процесу. Платформа надає можливості керування навчальними класами, розміщення та використання навчальних матеріалів, виконання завдань, перевірки результатів навчання, а також забезпечує ефективну комунікацію між учнями та вчителями.

Особливостями програмної реалізації SmartClass є поєднання серверного підходу Blazor, чистої архітектури, асинхронної обробки запитів, гнучкого доступу до даних через Entity Framework Core та узагальненого репозитарію зі специфікаціями, використання компонентного інтерфейсу користувача та реалізація обов'язкових чатів у реальному часі у класі. Всі ці рішення забезпечують високу продуктивність, безпеку, зручність використання та створюють надійну основу для подальшого удосконалення платформи в умовах реального освітнього процесу.

Ключові слова: SmartClass, вебплатформа, освітня платформа, цифрові технології, навчальні ресурси, дистанційне навчання, змішане навчання, комунікація.

Abstract

Halushka Yu. A. Design and development of a web platform for educational content management and communication – Manuscript

Master's thesis in the specialty 014 Secondary education (Informatics). – Lesya Ukrainka Volyn National University. Lutsk, 2025.

The qualification work analyzed the challenges of digitalizing the educational process and the development of modern web platforms to support education in distance and blended learning environments. Based on the analysis, the main requirements for the development of one's own educational platform were determined, including functional, non-functional, pedagogical and technical aspects.

The SmartClass educational web platform has been developed to manage educational content and communication between participants in the educational process. The platform provides the ability to manage classrooms, post and use educational materials, organize tasks, check learning results, and also ensures effective communication between students and teachers.

The features of the SmartClass software implementation are a combination of the Blazor server-side approach, clean architecture, asynchronous query processing, flexible data access via Entity Framework Core and a generalized repository with specifications, the use of a component user interface, and the implementation of mandatory real-time chats in the classroom. All these solutions provide high performance, security, ease of use, and create a reliable basis for further improvement of the platform in the conditions of the real educational process.

Keywords: SmartClass, web platform, educational platform, digital technologies, educational resources, distance learning, blended learning, communication.

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ	9
1.1. Системи управління навчанням: сучасний стан, переваги та особливості .	9
1.2. Основні технологічні тренди в освітніх платформах	13
1.2.1. Класифікація освітніх вебплатформ	15
1.2.2. Класифікація систем управління навчанням за архітектурою розміщення.....	19
1.2.3. Управління навчальним контентом, підходи та технології.....	22
1.2.4. Комунікація на освітніх вебплатформах	25
1.3. Огляд та порівняльний аналіз сучасних освітніх платформ	31
1.4. Обґрунтування необхідності розробки власної вебплатформи	37
РОЗДІЛ 2. РОЗРОБКА ТА РЕАЛІЗАЦІЯ ОСВІТНЬОЇ ВЕБПЛАТФОРМИ ДЛЯ УПРАВЛІННЯ НАВЧАЛЬНИМ КОНТЕНТОМ ТА КОМУНІКАЦІЇ	39
2.1. Постановка задачі та формування вимог до вебплатформи SmartClass.....	39
2.2. Вибір моделі та методології розробки програмного забезпечення	41
2.3. Загальний опис проєкту та архітектури вебплатформи	43
2.4. Обґрунтування вибору інструментальних засобів розробки.....	46
2.4.1. Вибір Visual Studio для розробки веборієнтованої платформи.....	46
2.4.2. Застосування мови програмування C# та серверного фреймворку Blazor Server для розробки програмного засобу.....	49
2.4.3. Застосування SQL Server для зберігання та обробки даних.....	51
2.5. Особливості програмної реалізації.....	53
2.6. Організація тестування та налагодження програмного засобу	59
2.7. Рекомендації по використанню та впровадженню програмного засобу	60
ВИСНОВКИ.....	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	67
ДОДАТКИ.....	72

ВСТУП

Актуальність теми. На сьогодні навчальні заклади функціонують в умовах глибокої цифрової трансформації суспільства та освіти. Запровадження дистанційного та змішаного навчання, інтеграція цифрових компетентностей у навчальний процес та потреба в оперативній комунікації між усіма учасниками освітнього простору є визначальними факторами ефективності роботи закладу освіти. У цьому контексті проектування та розробка вебплатформи, яка інтегрує управління навчальним контентом і інструменти комунікації є актуальним завданням. Сучасне програмне забезпечення дозволяє створювати гнучкі, масштабовані та персоналізовані навчальні середовища, які відповідають потребам як учнів так і педагогів. З'являється все більше платформ, що надають такі інструменти. Цей процес не зупиняється, оскільки еволюція у технологіях і системі освіти спонукає вдосконалювати наявний функціонал та додавати нові унікальні можливості для задоволення потреб сучасних здобувачів освіти. Однак існує низка обмежень, пов'язаних зі складністю використання існуючих платформ, а також індивідуальними потребами навчальних закладів.

Актуальність даного дослідження зумовлена швидким розвитком технологій дистанційного та змішаного навчання, а також зростаючим попитом щодо вимог до гнучкості та адаптивності освітніх платформ. Традиційні системи управління навчанням (LMS) наприклад як: Moodle, Blackboard, Google Classroom і так далі, мають деякі обмеження, такі як складність налаштування, обмежена інтеграція з сучасними технологіями аналізу освітніх даних, неповна персоналізація процесу навчання та відсутність гнучких механізмів зв'язку. Сучасні ж освітні системи вимагають нових способів управління та спільного використання ресурсів. Це вимагає введення інноваційних підходів до проектування освітніх платформ, які відповідають новим вимогам і потребам користувачів. Таким чином, розробка освітньої вебплатформи, що буде поєднувати зручний інтерфейс, ефективне управління навчальними матеріалами та можливості для інтерактивної комунікації, є дуже актуальним завданням, що

сприятиме підвищенню якості освіти.

Актуальність дослідження полягає у створенні безпечної та цілісної платформи, яка забезпечуватиме управління навчальними матеріалами, домашніми завданнями, оцінюванням та комунікацією між учасниками освітнього процесу. Надаватиме інструменти для взаємодії у реальному часі, персоналізований доступ до контенту, а також можливість до подальшого розвитку та інтеграції сучасних сервісів. Такий комплекс функцій дозволить підвищити організованість, прозорість і ефективність навчання.

Мета дослідження полягає у проектуванні та розробці освітньої вебплатформи для управління навчальним контентом та комунікацією між учасниками освітнього процесу в умовах змішаного чи дистанційного навчання.

Відповідно до теми та мети визначено **завдання дослідження**:

1. Проаналізувати існуючі освітні системи, визначити їх переваги та недоліки.
2. Дослідити сучасні технологічні тренди, що впливають на розвиток нового функціоналу у вебплатформах а саме:
 - хмарні технології та сучасні архітектурні патерни (багатошарова /чиста архітектура);
 - безпека та захист даних, інтеграція протоколів авторизації (JWT, OAuth 2.0);
 - інтеграція та API-взаємодія: підтримка REST API, можливість підключення сторонніх сервісів.
3. Розробити концепцію адаптивного освітнього застосунку, що враховує вимоги змішаної і дистанційної форм навчання.
4. Обрати інструментальні засоби, що будуть використовуватися для розробки платформи та обґрунтувати вибір.
5. Створити модель архітектури вебзастосунку, розробити дизайн та функціонал для управління контентом і комунікацією.
6. Додати інструменти для аналітики навчального процесу та персоналізованого навчання.

7. Протестувати функціональність вебплатформи.

8. Оцінити ефективність розробленого рішення та сформулювати рекомендації щодо його впровадження.

Об'єкт дослідження – комп'ютерні системи навчального призначення.

Предмет дослідження – технології проектування, засоби створення та методичні особливості змістового наповнення вебплатформи для управління навчальним контентом і підтримки комунікації між учасниками освітнього процесу.

Наукова новизна: створений застосунок поєднує можливості управління навчальним контентом та комунікацію між користувачами. Вебплатформа SmartClass відрізняється зручністю використання, можливістю персоналізувати кабінет та включає сучасні технології для підвищення ефективності освітнього процесу. Удосконалено механізми інтерактивної комунікації між учнями та педагогами, що сприяє більш ефективному обміну знаннями та інформацією.

Практичне значення результатів дослідження: розроблено вебплатформу SmartClass, яка може бути запроваджена у закладах загальної середньої освіти для покращення якості організації дистанційного та змішаного навчання. Платформа надає практичні можливості керування навчальними класами, розміщення та використання навчальних матеріалів, організації завдань, перевірки результатів навчання, а також забезпечує ефективну комунікацію між учнями та вчителями.

Методологічну основу дослідження склали: теоретичні методи (аналіз наукової, педагогічної та психологічної літератури); емпіричні методи (спостереження, опитування, анкетування); методи оцінки ефективності; статистичні методи обробки отриманих даних.

Апробація результатів дослідження відбулася на базі Старопочаївської загальноосвітньої школи I-III ступенів Почаївської міської ради Тернопільської області під час педагогічної та переддипломної практик.

Результати дослідження були представлені на Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених «Сучасна освіта

і наука Волині» (тези); XIV Міжнародній науково-практичній конференції «Математика. Інформаційні технології. Освіта» (тези); мультидисциплінарній науково-практичній конференції «Сучасні тенденції соціально-гуманітарного розвитку суспільства» (тези).

1. Галушка Ю. А. Використання сучасних онлайн-платформ в освітньому середовищі. *Сучасна освіта і наука Волині* : матеріали Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених (м. Луцьк, 22 травн. 2025р.). Луцьк: ФОП Мажула Ю. М., 2025. С.268-269 [10]

2. Галушка Ю. А., Ройко Л. Л. Освітня платформа як інструмент ефективної взаємодії учасників навчального процесу. *Математика. Інформаційні технології. Освіта* : збірник тез доп. XIV міжнар. наук.-практ. конф. (м. Луцьк, 13-15 червн. 2025 р.). Луцьк, 2025. С. 180-181 [11]

3. Галушка Ю. А., Ройко Л. Л. Роль сучасних освітніх платформ для колективної взаємодії учасників навчального процесу. *Соціально-гуманітарний вісник* : зб. наук. праць. Харків : СГНТМ «Новий курс». 2025. Вип.61. С.25-26 [12]

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Системи управління навчанням: сучасний стан, переваги та особливості

Цифровізація освіти значною мірою переформатувала підходи до роботи з навчальним контентом. Відбувається перехід від класичного підручника як основного джерела знань до створення цифрового навчального середовища, що інтегрує різноманітні ресурси та забезпечує особистісний підхід до здобувача освіти.

Сучасна шкільна освіта потребує не просто електронного архіву файлів, а зручної, безпечної та інтуїтивної платформи, яка стане основним інструментом організації навчального процесу. Така система має полегшувати працю вчителя з підбору та структурування матеріалів, забезпечувати персоналізацію навчання для учня та давати батькам зручний інструмент для моніторингу навчального процесу їхнього школяра. Реалізація цих вимог сприятиме створенню сучасного, ефективного та безпечного цифрового освітнього середовища у закладі загальної середньої освіти.

Освітні вебсайти (системи управління навчанням, LMS; навчальні платформи; мобільні додатки) стали настільки важливим інструментом, що важко уявити будь-який навчальний предмет без нього. Вони об'єднують інструменти для організації навчального процесу, управління навчальним контентом, моніторингу, оцінки та інформування всіх учасників навчального процесу – учнів, вчителів, батьків, адміністрації.

Під навчальними середовищами розуміється інтегроване цифрове середовище, яке функціонує на основі Інтернету речей (IoT) та сприяє навчанню через дистанційні або хмарні системи. Основні функції таких платформ включають:

- розміщення та організація освітнього контенту (текстів, відео, звітів,

проектів);

- комунікацію в режимі реального часу або асинхронно;
- автоматизацію оцінювання;
- аналітику результатів навчання;
- адміністрування доступу до матеріалів та контроль прогресу.

Вебплатформи – це інструменти для реалізації принципів відкритості, мобільності, гнучкості та самостійного навчання. У науково-педагогічній літературі їх також часто розглядають як частину цифрового навчального середовища (ЦНС) або електронного навчання.

На сьогодні в освітній сфері України та світу активно використовуються наступні вебплатформи [5]:

- Moodle – безкоштовна LMS з широкими можливостями налаштування, поширена у вищих навчальних закладах;
- Google Classroom – хмарна платформа для шкіл, яка зручно інтегрується з іншими сервісами Google;
- Microsoft Teams for Education – потужна платформа для синхронного та асинхронного навчання через відео;
- Edmodo, Canvas, Blackboard – комерційні системи з розширеним функціоналом;
- нові українські розробки: «Одна школа», «Моя школа», «Урок» тощо.

Кожна платформа має свої переваги та недоліки. Google Classroom, наприклад, пропонує простоту використання, але має обмежені можливості аналітики. Moodle потужний, але складний у налаштуванні. Це створює потребу в інноваційних рішеннях, що поєднують гнучкість, інтуїтивність, інтерактивність та адаптивність.

Огляд науково-методичної літератури [20, 31, 39] дав можливість виявити ключові тренди та вимоги у розвитку освітніх вебплатформ:

1. Інтеграція штучного інтелекту (ШІ), який використовується для адаптивного навчання, автоматизованого відстеження завдань, прогнозування успіху, персоналізованих рекомендацій. Такі платформи, як Squirrel AI,

Education, вже демонструють успішне використання ІІІ в освіті.

2. Аналіз освітніх даних (Educational Analysis). Вебплатформи все більше зосереджуються на підходах, керованих даними: зборі та аналізі даних про поведінку, прогрес та залученість учнів. Це дозволяє вчителям приймати обґрунтовані навчальні рішення.

3. Багато сучасних платформ мають мобільні додатки або індивідуальні розробки, які дозволяють учням навчатися на будь-якому пристрої.

4. Розробка концепції мікросервісів забезпечує високу розширюваність за стосунку та дозволяє легко додавати додаткові компоненти - чат-боти, платіжні системи, відеохостинг тощо.

5. Сучасні платформи повинні підтримувати не лише навчання, а й спільну діяльність, комунікацію та обмін ідеями, що створює цифрову спільноту.

6. Що стосується обробки персональних даних, платформи повинні відповідати міжнародним стандартам захисту даних (GDPR, ISO 27001).

7. Використання імерсивної, віртуальної та доповненої реальності стає все більш популярним. Це дозволяє учням залишатися зосередженими та навчатися ефективніше.

8. Платформи, що працюють у різних країнах, повинні враховувати культурні та лінгвістичні фактори, адаптуючи контент до місцевих освітніх програм.

Впровадження систем управління навчанням стало одним із ключових факторів трансформації сучасної освіти. Такі системи все частіше функціонують не лише як додатковий технічний інструмент, а й як інтегроване освітнє середовище, що визначає нові моделі навчання, комунікації, оцінювання та управління контентом. Переваги LMS очевидні на багатьох рівнях серед яких можна виділити [30]:

1. Структурованість та централізація освітнього процесу: усі навчальні матеріали, завдання, інструкції, повідомлення та результати зібрані в одній системі, що дозволяє створити прозоре середовище для управління навчанням, де вчитель може ефективно планувати навчальний процес, а учень – легко

орієнтуватися у змісті уроку, відстежувати свій прогрес та отримувати зворотний зв'язок.

2. Доступність освіти: навчання можливе незалежно від географічного розташування, фізичних здібностей чи соціального статусу учня; підтримуючи асинхронну взаємодію, користувачі можуть навчатися у власному темпі, у зручний для них час, з повторенням складних матеріалів.

3. Автоматизація рутинних процесів, включаючи перевірку тестів, виставлення оцінок, збір статистики щодо активності учнів, що зменшує навантаження на вчителя та звільняє час для творчої педагогічної роботи; окрім того, системи управління навчанням зазвичай мають вбудовані аналітичні модулі, які дозволяють відстежувати відвідуваність, успішність, проблемні етапи уроку, прогнозувати результати та коригувати їх.

4. Гнучкість та масштабованість, що дозволяє адаптувати систему до різних форм навчання та специфіки навчального закладу чи безпосередньо заняття, а деякі з них дозволяють інтеграцію сторонніх сервісів (наприклад, відеоконференції, сервіси тестування, віртуальні лабораторії).

Окрім технічних та педагогічних переваг, варто відзначити мотиваційний та соціальний потенціал LMS. Сучасні системи передбачають наявність форумів, чатів, стрічок новин, значків, нагадувань – елементів, що підтримують інтерес, взаємодію та відчуття спільноти. Завдяки цьому платформа стає не лише місцем для отримання знань, а й соціальним простором, де формуються стабільні освітні зв'язки та культура взаємодії.

Системи управління навчанням також сприяють збереженню та повторному використанню контенту, що підвищує ефективність освітнього процесу в довгостроковій перспективі. Один і той самий урок можна використовувати кілька разів, оновлювати, адаптувати для нових груп, що зменшує витрати часу та ресурсів на підготовку матеріалу.

Системи управління навчанням відіграють особливу роль у кризових ситуаціях в освіті – під час пандемій, воєнних дій, надзвичайних ситуацій. У таких умовах системи управління навчанням стають ключовим інструментом

для забезпечення безперервності освіти, організації освітнього процесу без фізичної присутності учасників.

Використання систем управління навчанням забезпечує якісно новий рівень організації освітнього процесу, який відповідає викликам сучасного інформаційного суспільства. Їх впровадження дозволяє одночасно підвищити ефективність навчання, зменшити адміністративне навантаження, забезпечити гнучкість, інклюзію та цифрову трансформацію освіти. Ці переваги формують основу для доцільності розробки власних освітніх платформ, які поєднують найкращі операційні рішення з інноваційними технологіями аналітики, персоналізації та комунікації [47].

1.2. Основні технологічні тренди в освітніх платформах

У процесі цифровізації освіти розвивається роль веборієнтованої освіти, яка все більше стає багатофункціональною платформою для організації, управління, підтримки та вдосконалення освіти. Водночас самі ці платформи змінюються через глобальне ІТ-середовище. Розглянемо і проаналізуємо роль хмарних технологій, мобільного навчання та штучного інтелекту в освіті (AI in Education).

Хмарні технології (cloud computing) – це спосіб зберігання, обробки та доступу до даних через Інтернет без прив'язки до будь-якого фізичного пристрою [26].

У системі освіти хмара відкриває унікальні можливості для простого, ефективного та економічно вигідного управління освітніми ресурсами. Сучасні вебпортали, такі як Google, Educational Workspace, Moodle Cloud, Augmentation, Cloud Cloud, – це хмарні сервіси, які не потребують розміщення на одному пристрої. Це особливо важливо для закладів освіти, де ІТ-ресурси можуть бути обмежені. Переваги хмарних рішень для освітніх платформ:

- цілодобовий доступ з будь-якого пристрою, що робить навчання мобільним;

- легка масштабованість, яка дозволяє сайту зростати пропорційно кількості користувачів;
- спільна робота над документами, проєктами, презентаціями у режимі реального часу;
- безпека та автоматизоване управління даними;
- зниження витрат на ІТ-інфраструктуру та технічну підтримку.

Хмарні технології особливо зручні в умовах змішаного та дистанційного навчання, де потрібна постійна синхронізація ресурсів, швидке оновлення даних та доступ до освітніх ресурсів незалежно від того, де знаходяться учні та педагоги.

Мобільне навчання – це формат навчання, який дозволяє учням навчатися «на ходу», адаптуючи цифрові навчальні ресурси до смартфонів і планшетів у комфортному для них темпі. Цей формат відкриває нові можливості для персоналізованого навчання, особливо в країнах і регіонах, де бракує освітньої інфраструктури. Мобільне навчання доповнює традиційну освіту, підвищуючи її інтерактивність, доступність та зручність.

Функціональність платформ мобільного навчання включає:

- мобільні додатки освітніх систем (наприклад, Moodle Mobile, Google Classroom App, Kahoot App);
- Push-сповіщення, що заохочують регулярну активність учнів;
- інтерактивний контент: відео, аудіо, анімація, елементи доповненої реальності;
- інтеграцію з календарями, розкладами, системами нагадувань;
- підтримку офлайн-доступу до контенту.

Платформи, які не адаптовані до мобільного навчання, втрачають значну частину своєї аудиторії, особливо серед молодших користувачів. У майбутньому мобільність стане ключовою рисою системи освіти.

Новітнім технологічним трендом став штучний інтелект (ШІ), який поступово змінює не лише технології, а й методи навчання. У системі освіти ШІ використовується для аналізу поведінки учнів, прогнозування результатів,

оптимізації змісту курсів, автоматизації оцінювання тощо. ІІІ на платформах не просто допомагає – він «вчить, як навчати».

Основні сфери застосування ІІІ в освітніх умовах:

1. Індивідуалізоване навчання, наприклад системи Smart Sparrow або DreamBox, використовують персоналізовані стратегії навчання на основі прогресу учня.

2. Стратегічні рекомендації, коли платформи пропонують персоналізовані ресурси, вікторини або теми для оглядів на основі минулого досвіду учня.

3. Аналітика навчання – алгоритми ІІІ визначають закономірності успіху, інтерпретують рівні залученості та прогнозують проблеми розвитку.

4. Автоматизоване оцінювання – вікторини, різного роду завдання, а у деяких випадках навіть звіти можуть оцінюватися автоматично (за допомогою NLP – обробки природної мови).

5. Віртуальні наставники та інтерактивні інструменти, такі як штучний інтелект «павук», навчання Карнегі або інструменти для спільної роботи – надають пояснення, пропозиції та зворотний зв'язок у режимі реального часу.

Переваги ІІІ очевидні – налаштування, ефективність, економія часу вчителя, точність оцінювання. Однак, водночас виникають етичні та технічні виклики: захист персональних даних, прозорість системи, довіра до машинної діагностики.

У підсумку можна зазначити, що величезне навчальне середовище та технологічні можливості відображають не лише зовнішній вигляд платформи, але й гнучкість її методів навчання.

1.2.1. Класифікація освітніх вебплатформ

У процесі інформатизації освіти з'явилося багато цифрових рішень для підтримки, інтеграції та спрощення освітнього процесу. Ці рішення складають цілу категорію інструментів, які часто називають освітніми онлайн-платформами. Різноманітні види діяльності, сфери застосування, методи та

технологічні архітектури визначають вимоги до розробки таких систем. Їх класифікація дозволяє краще зрозуміти призначення конкретної системи, її функціональні можливості та потенціал для вирішення педагогічних завдань.

Найпоширенішими критеріями класифікації освітніх онлайн-платформ є:

- тип освітнього середовища (формальне, неформальне, корпоративне тощо);
- форма організації освіти (дистанційна, змішана, мікронавчання);
- функціональне призначення (управління освітнім процесом, проведення курсів, підтримка самостійного навчання);
- цільова аудиторія (школярі, студенти, дорослі, співробітники компаній);
- технічне керівництво (вебдодатки, мобільні додатки, настільні додатки тощо);

Виходячи з цих підходів, ми розглянемо основні типи освітніх онлайн-платформ.

Системи управління навчанням (Learning Management Systems – LMS) є найпоширенішими освітніми платформами. Вони призначені для організації, спрямування та моніторингу навчального процесу у формальному середовищі. Такі системи використовуються як у навчальних закладах (загальноосвітніх та вищих), так і в приватних освітніх центрах.

Основні функції LMS:

- створення та розповсюдження освітнього контенту;
- організація курсів, тем, модулів;
- тестування знань (тести, вікторини, автоматизовані оцінювання);
- зворотній зв'язок, форуми, чати;
- моніторинг прогресу та аналіз ефективності.

Наприклад:

1. Moodle – найпопулярніша відкрита система управління навчанням.
2. Google Classroom – хмарне рішення для закладів освіти.
3. Canvas, Blackboard – популярні комерційні системи управління.
4. Open edX, Chamilo, Schoology – використовуються у вищій та

післядипломній освіті.

Системи управління навчанням мають досить широкий функціонал та гнучку систему налаштування, але можуть вимагати значних зусиль з управління, налаштування та технічної підтримки.

Окрім LMS систем популярністю користуються так звані MOOC платформи, класифікацію яких подано на рисунку 1.1. Головна мета MOOC – надати вільний, безкоштовний або доступний за ціною доступ до якісної освіти через Інтернет.



Рисунок 1.1. Класифікація MOOC платформ

Найпопулярніші платформи MOOC: Coursera, edX, Udemy, Udacity, Prometheus.

Деяко схожими на попередні є корпоративні навчальні платформи (Enterprise LMS, LXP). Вони розроблені для внутрішнього навчання співробітників компанії. Часто виступають частиною стратегії управління та розвитку персоналу.

Ключові функції:

- навчання за професійними програмами;
- внутрішня система сертифікації;
- моніторинг ефективності навчання;

- інтеграція з HR-системами та KPI;
- траєкторії навчання для запуску (адаптації) нових співробітників.

У прикладі подібних платформ можна навести наступні: SAP Litmos, TalentLMS, Docebo, 360Learning, SuccessFactors або ж кастомні платформи на базі Moodle чи SharePoint.

Новим підтипом є Learning Experience Platforms (LXP) – системи, що базуються на штучному інтелекті та аналітиці, вони автоматично підбирають освітній контент на основі інтересів та потреб співробітників. Корпоративні платформи стають все більше схожими на ринок освіти – інтегруючи внутрішні та зовнішні ресурси, відео, MOOC, навчальні курси, семінари.

З розвитком освітніх технологій з'явилися різні типи платформ, які не пов'язані безпосередньо з формальною освітою, але мають активний вплив. Такі платформи мають переваги простоти, інтерактивності та адаптивності до мобільного навчання. Їх часто використовують як доповнення до основної освітньої платформи або як інструмент для домашнього навчання рисунок 1.2.



Рисунок 1.2. Найпопулярніші платформи неформальної освіти

Серед них можна виділити наступні: Skillshare, LinkedIn Learning, Khan Academy (навчальні матеріали представлені у вигляді коротких курсів, міні-уроків, інфографіки); Duolingo, Memrise (гейміфіковане вивчення мови); Quizlet, Kahoot!, Wordwall (експериментальна станція, дизайнерські ігри та класні заняття).

Класифікація освітніх вебплатформ дозволяє чітко визначити їх функціональність, сферу застосування та цільову аудиторію. LMS орієнтовані на організацію систематичного навчання у формальному середовищі. MOOC-платформи роблять знання масовими та відкритими. Корпоративні системи є ефективними інструментами для особистісного розвитку. EdTech-проекти створюють нові форми навчання – короткі, мобільні, ігрові, гнучкі [20].

У контексті нашого дослідження розглянута класифікація стане основою для вибору та обґрунтування архітектури власної вебплатформи, спрямованої на поєднання управління контентом, ефективної комунікації та аналітики.

1.2.2. Класифікація систем управління навчанням за архітектурою розміщення

Успішна робота систем управління навчанням залежить не лише від їхніх навчальних можливостей та інтерфейсу користувача, але й від архітектурних рішень, що керують їхньою інфраструктурою. Архітектура розміщення LMS визначає, як система зберігається, керується, масштабується, інтегрується з іншими інструментами, а також забезпечує безпеку та зручність використання.

Залежно від розташування програмного забезпечення, навчальних даних та інфраструктури, системи управління навчанням можна класифікувати на локальні, хмарні, гібридні та децентралізовані.

Локальні системи управління навчанням розгортаються на фізичних серверах, що належать або управляються навчальним закладом. Програмне забезпечення встановлюється безпосередньо на сервері навчального закладу. Особливості роботи таких систем:

- повний контроль над даними та інфраструктурою;
- потреба в ІТ-фахівці для підтримки;
- поточні витрати на оновлення апаратного та програмного забезпечення;
- обмежена мобільність та доступ ззовні.

Подібні системи мають високий рівень безпеки та конфіденційності даних та здатні гнучко адаптуватися до специфіки закладу. Проте існують і недоліки: високі витрати на розробку та обслуговування, складність у масштабуванні та низьку доступність у разі аварії або збою.

Хмарні системи управління навчанням працюють на сторонній інфраструктурі (наприклад, Amazon Web Services, Google Cloud, Microsoft Azure) та доступні через Інтернет. Вони не потребують локального розгортання – управління, обслуговування та оновлення здійснює постачальник.

Приклади: MoodleCloud, Google Classroom, Canvas Cloud, TalentLMS. :

Особливості роботи хмарних систем:

- швидкий запуск;
- доступ з будь-якого пристрою та місця розташування;
- автоматичне масштабування;
- регулярні оновлення та резервні копії.

Подібні системи не потребують великих витрат на обслуговування інфраструктури, прості у розгортанні та підтримці. Вони мають високу доступність та продуктивність і надають можливість гнучкого такрифікування (оплати за використання). Не дивлячись на свої переваги є також і недоліки:

- обмежений контроль даних;
- залежність від постачальника та наявності стабільного підключення до інтернету;
- можливі обмеження в налаштуванні.

Хмарні системи є найпоширенішими серед сучасних навчальних платформ, особливо в середовищах дистанційного та змішаного навчання.

Наступним етапом розвитку стали гібридні системи, що поєднують елементи локального і хмарного розміщення. Вони забезпечують розподіл даних

між локальними та хмарними ресурсами зберігаючи баланс між безпекою та мобільністю і здатні адаптуватися до особливостей організації. Гібридні моделі стають популярними серед великих компаній, які хочуть поєднати гнучкість хмари з безпекою локального сховища.

Загалом до переваг можна віднести:

- підвищену гнучкість;
- контроль над критичними даними при використанні хмарних сервісів для некритичних даних;
- підвищення продуктивності завдяки оптимізації навантаження.

Недоліки:

- адміністративна складність;
- необхідність налаштування інтеграції між відділами;
- додаткові витрати на підтримку обох середовищ.

Децентралізовані LMS – системи побудовані на блокчейн-технологіях, peer-to-peer мережах або інших моделях, що не передбачають централізованого управління. Приклади: ODEM (Open Education Market), Blockchain Education Network, BitDegree. Вони забезпечують зберігання даних не в одному центрі, а розподіляються їх між вузлами забезпечуючи надійність. Ідентифікація користувача, контент та результати навчання можуть бути записані у реєстрі блокчейну. Хоча децентралізовані системи лише розвиваються, вони мають потенціал трансформувати уявлення про навчання як про особистий ресурс, а не лише про робоче місце.:

Переваги:

- високий рівень безпеки;
- фактичне володіння даними для всіх користувачів;
- можливість отримання міжнародних сертифікатів та визнання компетенцій (наприклад, через підтвердження визнання NFT).

Недоліки:

- новизна і слабка підтримка в освіті;
- складність експлуатації;

- технічна та правова невизначеність.

Отже, архітектура LMS безпосередньо впливає на стабільність, безпеку, масштабованість та ефективність освітньої платформи. Вибір між локальною, хмарною, гібридною або децентралізованою має ґрунтуватися на ресурсах навчального закладу, навчанні персоналу, обсягах даних, потребах у мобільності та інформаційних потребах.

У контексті нашого дослідження розробка освітньої платформи буде зосереджена на хмарній архітектурі з масштабованістю та можливостями модульної розробки, що дозволить досягти збалансованої функціональності, простоти та ефективності управління контентом та комунікацією.

1.2.3. Управління навчальним контентом, підходи та технології

Основою будь-якої освітньої платформи є освітній контент, який відображає знання, вміння, навички та діяльність, які має сформувати здобувач освіти. Ефективність навчання у цифровому середовищі значною мірою залежить від того, наскільки добре цей контент організований, структурований, доступний та представлений у адаптованій формі. В умовах постійного збільшення освітньої інформації та диференціації освітніх запитів управління контентом стає окремою галуззю досліджень та практики [13].

Підготовлена навчальна програма повинна базуватися на педагогічних та технологічних принципах, забезпечуючи її ефективність, доступність, актуальність та інтерактивність. Основні принципи:

1. Досягає цілей освіти, програм та результатів навчання. Інтелектуальне навчання базується на принципах – теорія необхідна, практика має бути практичною.

2. Створити унікальний навчальний досвід (модулі, розділи, уроки) для створення можливостей для самостійного навчання, створити можливості для створення цілеспрямованих навчальних можливостей.

3. Сьогоднішні інтерактивні елементи – тести, симуляції,

міждисциплінарні підходи – покращують результати навчання.

4. Розробка різних методів навчання (візуальних, слухових, кінестетичних) з використанням аудіо, відео, звукових та тактильних інструкцій.

5. Нове використання та додаткова інформація для редагування. Кров, друк, чорнило, зображення (альтернативний текст), відео та інші сертифіковані методи.

6. Бази знань клієнтів повинні бути детальними, перевіреними, протестованими та постійно оновлюватися.

7. Користувачі можуть вибирати, читати та спрощувати свій контент.

Завдяки цифровим технологіям були розроблені уніфіковані формати та стандарти представлення освітнього контенту, що полегшує зберігання, передачу, адаптацію та повторення в локальному середовищі.

Основні формати:

1. SCORM (SCORM Harable Content Reference Model) — те, що в більшості випадків є повторюваною модульною освітньою річчю. Забезпечує узгодженість та розуміння активності користувачів.

2. xAPI (Experience API, Participate API) – їжа, яку ви пишете. Ви можете коментувати (переглядати відео, брати участь в опитуваннях, брати участь у форумах) незалежно від платформи.

3. HTML5 – це інтерактивний вбудований медіа-стандарт в Інтернеті.

4. LTI (Tembóra Learning Interoperability) – дозволяє зовнішнім авторам отримувати доступ (тести, навчання, модулі) до LMS.

Типові елементи цифрового контенту:

- обговорення, відеоуроки, письмові статті, інфографіка;
- тестові роботи, виконавчі вправи;
- проєкти, такі як інтерактивні симуляції;
- форуми, чати, вебінари.

Завдяки наявності єдиних форматів появилась можливість переносити контент з однієї платформи на іншу, переробляти його під потреби різних курсів,

зберігати у відкритих освітніх ресурсах.

В освітньому середовищі управління контентом відіграє важливу роль у забезпеченні доступності та створенні ефективної системи навчання, яка відповідає індивідуальним потребам кожного користувача. Використання різних методів категоризації дозволяє організувати контент організовано та зрозуміло, що полегшує пошук та доступ до матеріалу. Класифікація за темою, рівнем кваліфікації, типом контенту, складністю та стилем подання допомагає створити чітку та логічну систему, яка враховує різноманітність вимог та потреб користувачів. Можна виділити наступні методи категоризації:

- за предметом: інформатика, фізика, історія тощо.
- за рівнем освіти: початкова, початкова, середня, вища освіта;
- за типом контенту: текст, відео, симуляція, тест;
- за складністю: базова, середня, просунута;
- за стилем презентації: інтерактивна, статична, гейміфікована.

Для методів персоналізації важливими аспектами виділяють вимоги:

- система пропонує контент на основі попередніх результатів користувача;
- контент вибирається на основі тем, що цікавлять конкретного читача;
- автоматично пропонують контент на основі активності користувача (подібно до youtube або netflix).
- використання методів навчання: візуальних, слухових, текстових.

Як результат, інтеграція категоризації та персоналізації контенту в навчальне середовище не лише підвищує ефективність навчального процесу, але й створює комфортні умови для кожного користувача, реагуючи на його конкретні потреби. Такий підхід допомагає створити гнучку та адаптивну освітню систему, яка може змінюватися відповідно до потреб учнів, забезпечуючи більш персоналізований та ефективний процес навчання.

Використання штучного інтелекту (ШІ) принципово змінює парадигму управління освітнім контентом. Раніше це було ручним завданням для вчителів та адміністраторів, але зараз зростає частка автоматизованих рішень, заснованих

на алгоритмах аналізу даних. Приклади застосування ШІ в управлінні контентом:

1. Штучний інтелект визначає рівень освіти користувача та вибирає контент, який відповідає складності чи стилю. Це дозволяє створювати індивідуальні траєкторії навчання.

2. Використання моделей GPT (наприклад, ChatGPT) дозволяє автоматично генерувати текстові пояснення, приклади, тести, поради тощо.

3. Штучний інтелект допомагає виконувати інтелектуальний пошук контенту на основі зв'язків між контентом, ключовими словами, темами, концепціями.

4. Алгоритми можуть аналізувати, які матеріали переглядаються найчастіше, які з них призводять до високих результатів, а які ігноруються або викликають труднощі.

5. Вони використовуються для керівництва учнями, пояснення складних тем та надання порад щодо вибору теми.

6. Деякі платформи використовують штучний інтелект (ШІ) для відстеження застарілих матеріалів та їх оновлення (наприклад, автоматично перевіряючи актуальність джерел, фактів та посилань).

Використання ШІ може значно оптимізувати управління контентом, зменшити навантаження на викладачів, а також підвищити точність і адаптивність навчального процесу.

1.2.4. Комунікація на освітніх вебплатформах

Ефективна комунікація є ключовим елементом навчального процесу. У цифровому навчальному середовищі комунікація забезпечує взаємну підтримку між вчителем та учнями, сприяє обміну інформацією, створює навчальні спільноти, розвиває мотивацію.

За форматом комунікація на освітніх платформах поділяється на синхронну (відеоконференції, чат, віртуальні класи) та асинхронну (форуми,

коментарі, email, блоги).

Синхронна комунікація передбачає взаємодію між учасниками навчального процесу в режимі реального часу. Прикладами такої взаємодії є онлайн-заняття через Zoom або Google Meet, участь у вебінарах та спільна робота над документами через Google Docs з одночасним редагуванням та обговоренням. Головною перевагою синхронної комунікації є висока швидкість обміну інформацією, можливість негайного зворотного зв'язку та створення ефекту присутності [9].

Асинхронна комунікація відбувається із затримкою у часі. Це можуть бути повідомлення на форумах, обговорення в чатах із затримкою відповіді, коментарі до завдань, електронні листи або взаємодія через систему управління навчанням (LMS). Такий тип комунікації забезпечує гнучкість, оскільки учасники мають можливість обміркувати відповідь, вивчити матеріал.

Асинхронна взаємодія часто є більш ефективною для розвитку критичного мислення, рефлексії та самостійного аналізу. Водночас, це має недолік у вигляді зниження динаміки процесу комунікації, а також ризику втрати емоційного контакту.

У більшості випадків оптимальним є поєднанням обох видів комунікації з врахування індивідуальних потреб учнів, особливостей та цілей уроку. На сучасних освітніх платформах така комбінована модель реалізується за допомогою різноманітних інструментів та медіа.

За типом взаємодії комунікація на освітніх платформах ділиться:

- вчитель – учень;
- учень – учень;
- групова комунікація;
- масова комунікація (анонси, сповіщення).

В онлайн-середовищі навчання використовуються різні інструменти комунікації, кожен з яких виконує різні функції та має свої переваги та недоліки. Наприклад, форум – це віртуальна відкрита аудиторія або читальний зал, де можна обговорити матеріал занять, задати питання, посперечатися та допомогти

один одному. Форум сприяє розвитку академічної комунікації та створенню академічного репозиторію, оскільки всі повідомлення архівуються та доступні для подальшого використання.

Комунікація через чат є ефективним методом прямого спілкування, наближеного до синхронної взаємодії, хоча в академічних колах вона часто використовується асинхронно. Вона підходить для обміну інформацією, інтерпретації, технічних та організаційних висновків. Однак через швидкий темп та динамічний характер розмови ця розмова має менший освітній потенціал, ніж форум, хоча є важливою для підтримування розмови «у прямому ефірі».

Відеоконференції стали найпопулярнішою платформою під час пандемії, військового стану. Вони пропонують інтерактивні можливості навчання, семінари, консультування через відеочати, обмін відео, цифрові нотатки тощо. Такі сервіси, як Zoom, Microsoft Teams, BigBlueButton, стали стандартом у віртуальному навчальному середовищі. Однак, особисті зустрічі та відеодзвінки вимагають стабільного інтернет-з'єднання, швидкості та техніки. Крім того, дистанційні відеолекції можуть призвести до втоми, вигорання та зниження залученості школярів, особливо за відсутності інтерактивних функцій.

Тому вибір інструментів навчальної комунікації повинен враховувати мету, стиль комунікації, склад групи та її можливості.

Онлайн-спілкування суттєво відрізняється від традиційної взаємодії у фізичному класі, що створює як нові можливості, так і виклики для учасників освітнього процесу. Однією з основних проблем є відсутність невербальних сигналів – міміки, жестів, інтонації, що ускладнює сприйняття емоційного стану співрозмовника, знижує рівень емпатії та соціальної взаємодії. Це може призвести до непорозумінь, відчуття ізоляції або емоційної дистанції між учнями та вчителем.

Ще одним аспектом є відчуття анонімності або дистанції, що іноді знижує бар'єри для участі в комунікації – сором'язливі або тривожні школярі активніше беруть участь в онлайн-дискусіях. Водночас це також може призвести до формального ставлення, низької відповідальності за власні висловлювання та

втрата академічної культури. Формування довірливого та підтримуючого середовища відіграє ключову роль.

В онлайн-освіті важливо створити простір, де учасники відчують себе частиною навчальної спільноти. Це можливо завдяки регулярному зворотному зв'язку, визнанню зусиль, використанню гумору, емпатії та неформальному стилю спілкування, особливо на початковому етапі курсу.

Крім того, важливими є емоційна регуляція та самодисципліна. У дистанційному навчанні зовнішній контроль знижується, тому учням потрібно бути більш самостійними, вміти планувати час, керувати своїми емоціями. У цьому контексті регулярне спілкування, підтримка педагога та поради щодо організації навчання сприяють покращенню психологічного клімату.

Успішне впровадження онлайн-комунікації в освітню платформу вимагає уваги до психологічних аспектів – не лише технологічної інтеграції інструментів, але й формування довірливих стосунків, підтримки, мотивації та чутливості до потреб здобувачів освіти.

З розвитком Web 2.0 та Web 3.0 зростає вплив соціальних технологій в освіті. Сучасні освітні онлайн-платформи все частіше включають соціальні елементи, підтримують неформальне спілкування та створюють відчуття приналежності до навчальної спільноти.

До таких елементів належать системи рейтингів, значки, лайки, коментарі, особисті профілі користувачів з фотографіями, біографіями та досягненнями. Платформи також впроваджують стрічки новин, сповіщення про активність інших учасників, можливість створювати групи за інтересами, ділитися контентом та брати участь у спільних проектах. Такі функції запозичені із соціальних мереж, але адаптовані до освітніх потреб.

Використання соціальних елементів створює умови для активного конструктивного навчання, що стимулює співпрацю, взаємну підтримку та розвиток комунікативних навичок, критичного мислення та цифрової грамотності. Крім того, соціальна активність на платформі може виступати мотивуючим фактором, сприяючи появі «лідерів думок» серед педагогів та

учнів.

Однак надмірна гейміфікація або зосередженість на винагородах може знизити внутрішню мотивацію та перетворити навчання на формальне змагання. Тому важливо підтримувати баланс між навчанням та розвагами.

Соціальні функції, інтегровані в освітню платформу, не лише розширюють комунікаційний потенціал, а й сприяють перетворенню платформи з інформаційного ресурсу на реальне освітнє середовище, де формуються не лише знання, а й цінності, ставлення та культура освітнього діалогу.

Для проєктування комунікаційного функціоналу освітньої вебплатформи необхідно ґрунтуватися на усталених теоретичних моделях, які описують механізми та типи взаємодії між учасниками.

Аналіз науково-педагогічних джерел [8, 16, 21] дозволяє виділити наступні ключові моделі соціальної взаємодії в онлайн-середовищі:

1. Модель взаємодії за типами комунікації (за часовим фактором):

Синхронна взаємодія (в реальному часі): модель імітує очне спілкування і забезпечує швидкий обмін думками, зменшуючи комунікаційні бар'єри. Прикладами на платформі можуть бути: відеоконференції, групові чати, віртуальні класні кімнати. Перевагами є: миттєвий зворотний зв'язок, відчуття спільності, можливість проведення мозкових штурмів. Недоліки: необхідність синхронізації за часом для всіх учасників, можливі технічні складнощі.

Асинхронна взаємодія (із затримкою в часі): модель є основою гнучкого навчання, дозволяючи кожному учаснику працювати у власному темпі. Прикладами на платформі можуть бути: форуми, електронна пошта, коментарі до завдань, дискусійні дошки, блоги. Перевагами є: гнучкість часу, можливість глибшого осмислення та формулювання думок, доступність для учасників з різних часових поясів. Недоліки: можлива затримка у відповідях, ризик зниження мотивації через відсутність миттєвої реакції.

2. Модель взаємодії за напрямками комунікації: модель класифікує потоки інформації між учасниками освітнього процесу:

- *Учень – Контент.* Базова взаємодія, де студент безпосередньо працює з

навчальними матеріалами. Завданням платформи зробити цю взаємодію максимально інтерактивною та ефективною (наприклад, через інтерактивні тести, симуляції).

- *Учень – Вчитель*. Ключова взаємодія для надання зворотного зв'язку, інструктажу та оцінювання. Платформа має забезпечувати зручні та різнобічні канали для такого зв'язку (особисті повідомлення, коментарі до робіт, канали для консультацій).

- *Учень – Учень*. Ця модель є основою соціального конструктивізму, де знання створюються шляхом спільного обговорення та вирішення проблем. Розвиток цієї взаємодії є ключем до створення справжньої навчальної спільноти (learning community).

Форми реалізації: групові проєкти, взаємооцінювання, дискусійні форуми, загальні вікі-сторінки.

- *Вчитель – Вчитель*. Взаємодія між педагогами для спільного планування уроків, обміну досвідом та контентом.

3. Модель «Спільнота практиків» описує групу осіб, об'єднаних спільною діяльністю (навчальним процесом) і які взаємодіють постійно. Її структура включає три ключові елементи:

- *Спільна домовленість* (учасники постійно взаємодіють через механізми платформи).

- *Розділене розуміння цілей навчання* (наприклад, успішне завершення курсу).

- *Спільний репертуар* (сукупність спільних ресурсів, мовленнєвих зворотів, інструментів, які створюються у процесі спільної діяльності – наприклад, глосарій уроків, архів успішних робіт).

4. Модель СОМ для аналізу якості онлайн-дискусій за трьома рівнями:

- *Соціальний рівень*: встановлення особистих зв'язків, підтримка.

- *Когнітивний рівень*: обмін інформацією, її аналіз, синтез, оцінка.

- *Метакогнітивний рівень*: рефлексія щодо власного процесу навчання, усвідомлення власних стратегій мислення.

Для ефективного проєктування освітньої вебплатформи необхідно інтегрувати інструменти, що підтримують різноманітні моделі взаємодії. Система має забезпечувати:

1. *Баланс між синхронними та асинхронними комунікаціями* для забезпечення як гнучкості, так і миттєвості зв'язку.

2. *Підтримку всіх напрямків комунікації («учень-вчитель», «учень-учень» тощо)* через спеціалізовані інструменти (форуми, чати, групові робочі простори).

3. *Створення умов для формування «Спільноти практиків»* шляхом заохочення спільної роботи над контентом, ведення групових блогів та спільних проєктів.

4. *Спроектований інтерфейс, який стимулює якісну комунікацію* на соціальному та когнітивному рівнях, наприклад, через шаблони для ведення дискусій, інструменти для рефлексії.

Таким чином, врахування цих моделей дозволить створити не просто набір інструментів для спілкування, а цілісне комунікаційне середовище, спрямоване на підвищення ефективності навчального процесу та якості освіти в цілому.

1.3. Огляд та порівняльний аналіз сучасних освітніх платформ

Сучасне освітнє середовище характеризується широким спектром цифрових платформ, що реалізують різні методи організації освітнього процесу. За своєю функціональністю, масштабом, спрямованістю та технічною архітектурою освітні платформи можуть використовуватися як у формальній та і неформальній освіті [36].

У контексті дослідження проведемо огляд та порівняльний аналіз навчальних платформ, що дозволить обґрунтувати вибір і розробку власного програмного рішення. Розглянемо та порівняємо найвідоміші освітні платформи: Moodle, Google Classroom, Canvas, Coursera та Prometheus.

Moodle (Modular Dynamic Object-Oriented Learning Environment) –

платформа, яка має модульну архітектуру, високий рівень налаштування та підтримує стандарти SCORM та xAPI. Moodle переважно використовується у вищих навчальних закладах завдяки своїй гнучкості, системі ролей користувачів, розширеним інструментам для організації курсів, форумів, тестів та аналітики. Однак основним недоліком є складність налаштування та необхідність технічної підтримки.

Moodle – це приклад локальної або гібридної системи, яку можна використовувати, як на власних серверах, так і у хмарі (MoodleCloud). Приклад інтерфейсу можна побачити на рисунку 1.3.

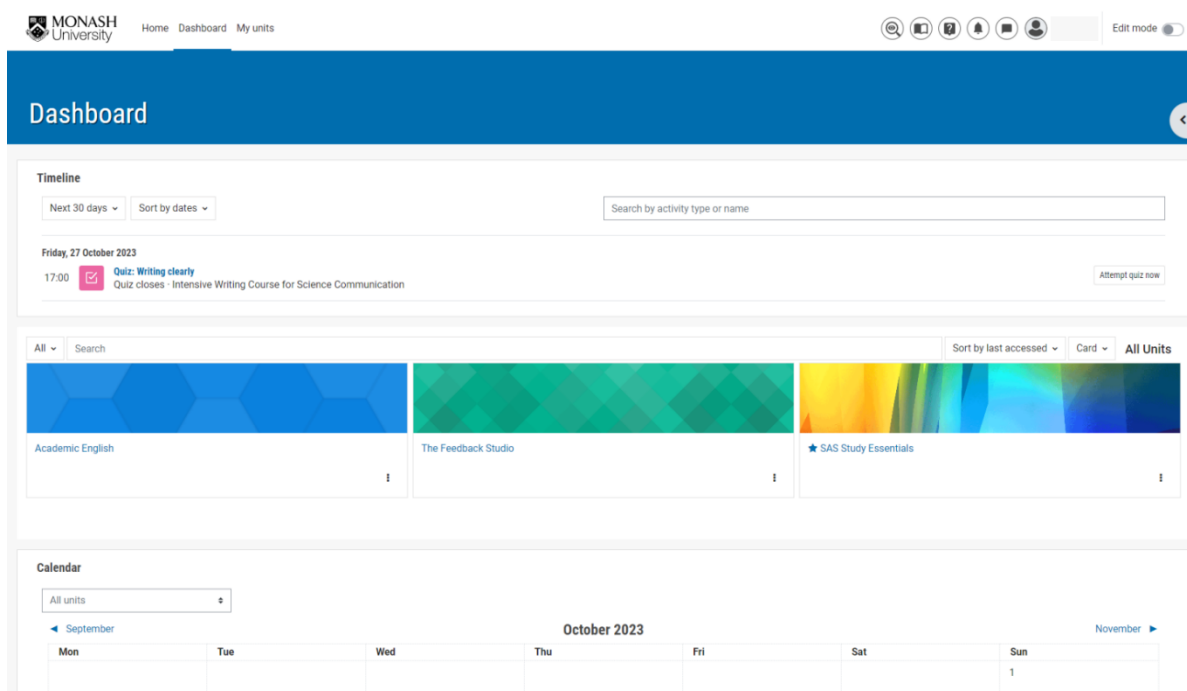


Рисунок 1.3. Приклад вигляду інтерфейсу освітньої платформи Moodle

Google Classroom – це хмарна система управління навчанням (LMS), що входить до складу Google Workspace for Education, яка у першу чергу орієнтована на заклади загальної середньої освіти. Вона легко встановлюється, легко масштабується та інтегрується з іншими сервісами Google (Docs, Drive, Meet, Calendar). Приклад інтерфейсу подано на рисунку 1.4.

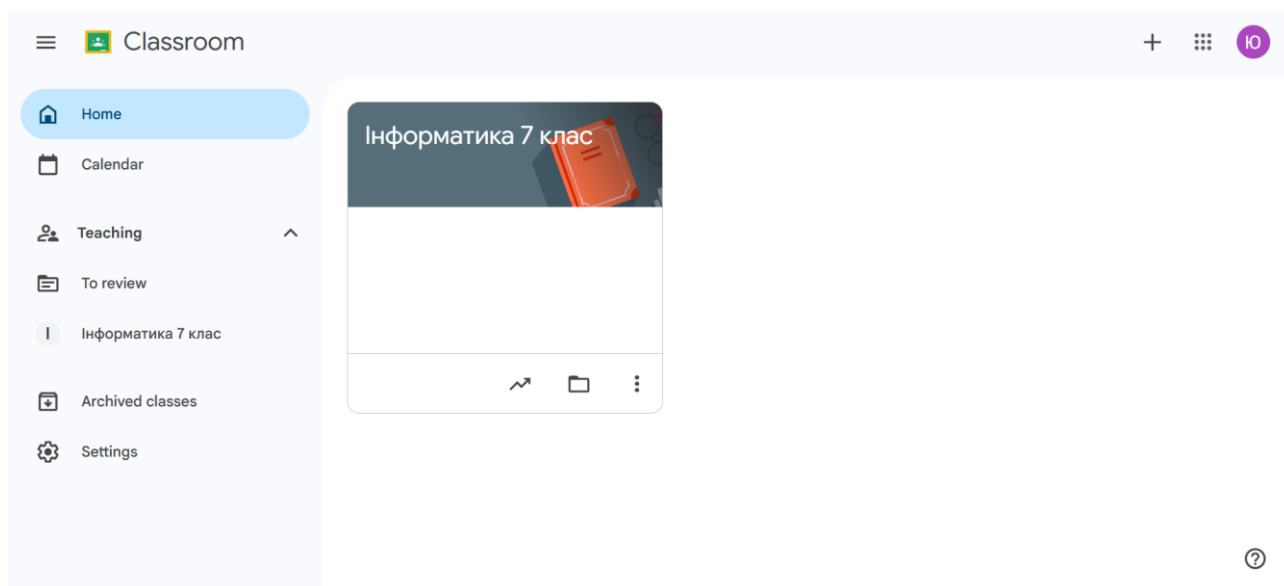


Рисунок 1.4. Приклад інтерфейсу Google Classroom

Переваги Google Classroom:

- безкоштовна та швидка установка;
- інтуїтивно зрозумілий та зручний інтерфейс;
- проста інтеграція з іншими продуктами Google;
- підтримка мобільного доступу;
- автоматичне створення папок на Google диску для зберігання завдань;
- просте керування групами студентів та курсами.

Недоліки Google Classroom:

- відсутність складної логіки уроків та тестів;
- обмежений аналіз навчального процесу;
- відсутність підтримки SCORM та інших міжнародних стандартів;
- залежність від екосистеми Google.

Canvas LMS – одна з найсучасніших комерційних систем управління навчанням, що широко використовується у Європі. Вона побудована на хмарній архітектурі, має інтуїтивно зрозумілий інтерфейс, підтримує інтеграцію з різними інструментами через відкритий API.

Педагоги мають можливість кастомізувати навігацію по курсу, схеми кольорів та розташування елементів, забезпечуючи тим самим оптимізований та

зручний користувацький досвід для всіх учасників навчального процесу. Приклад інтерфейсу подано на рисунку 1.5.

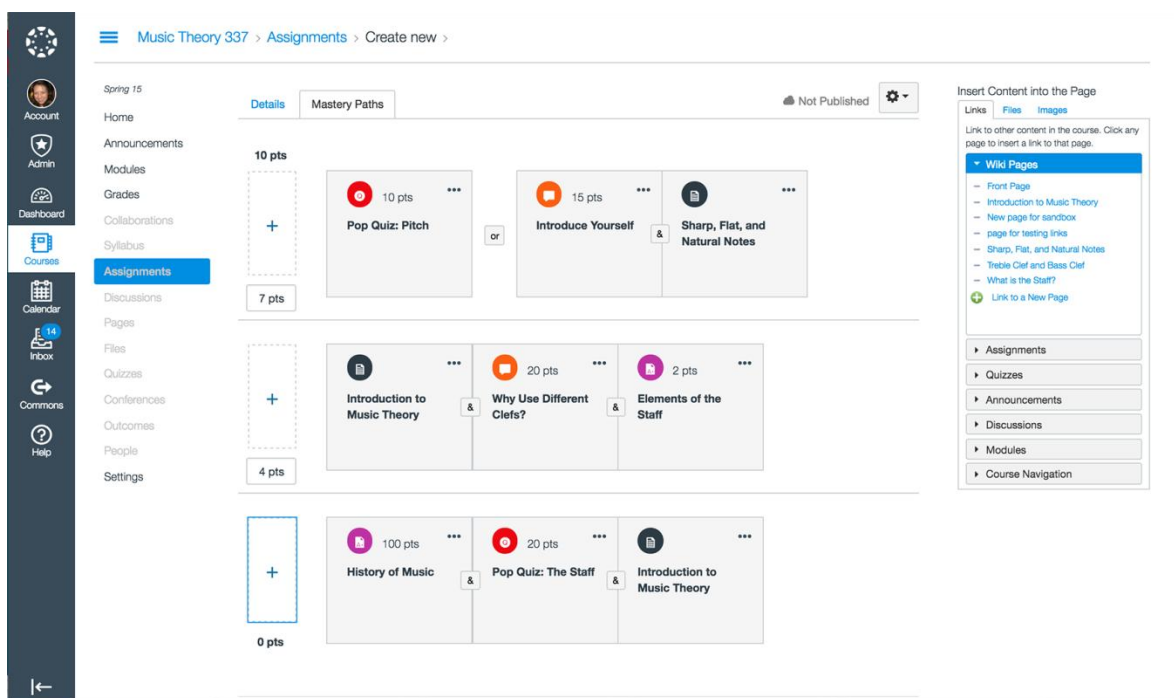


Рисунок 1.5. Приклад інтерфейсу системи управління навчанням Canvas

Переваги:

- сучасний, простий інтерфейс з підтримкою перетягування;
- потужна аналітика, включаючи Learning Analytics;
- підтримка інтеграцій (SCORM, xAPI, LTI, API);
- мобільні додатки для студентів та викладачів;
- проста у використанні система створення курсів та зворотного зв'язку;
- розширені інструменти для оцінювання, календарі, нагадування.

Недоліки:

- комерційна модель – повне використання платне;
- деякі функції недоступні в безкоштовній версії;
- високі вимоги до підключення до інтернету та сучасних пристроїв.

Coursera має найсучаснішу програму MOOC (масові відкриті онлайн-курси) для тестування. Вона зосереджена переважно на самостійному навчанні, підвищенні кваліфікації та професійної перепідготовки для дорослих. На

рисунку 1.6. можна побачити приклад інтерфейсу системи.

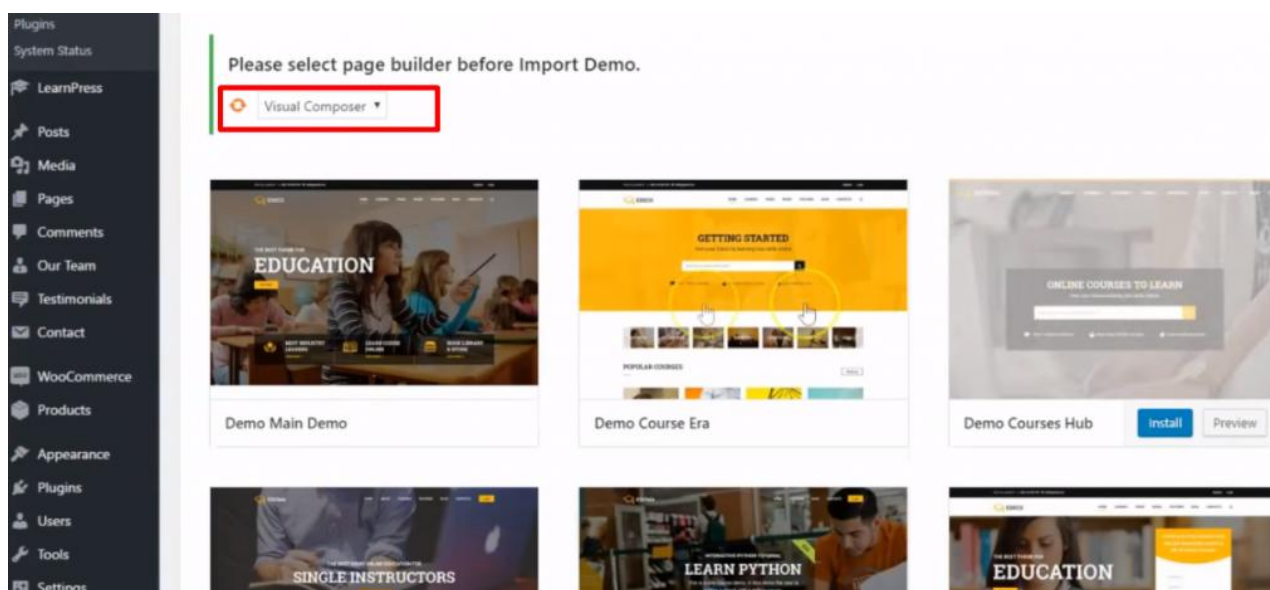


Рисунок 1.6. Вигляд інтерфейсу системи управління навчанням Coursera

Переваги Coursera:

- курси від провідних університетів;
- крім того, ви можете переглянути унікальний та аудіо текст;
- варіанти сертифікації;
- адаптивне навчання, субтитри, переклади;
- включає тести, вправи, дискусії, автоматичне оцінювання.

Недоліки:

- платний доступ до більшості повних курсів;
- відсутність можливості самостійної публікації навчальних матеріалів;
- орієнтована на самонавчання, а не повноцінну LMS для груп.

Prometheus – це українська безкоштовна платформа MOOC, яка є прикладом локально адаптованого освітнього ресурсу. Вона була створена для забезпечення доступу до високоякісної онлайн-освіти для українців з урахуванням місцевих особливостей, мови та соціального контексту. На рисунку 1.7 показано інтерфейс сторінки курсів на сайті.

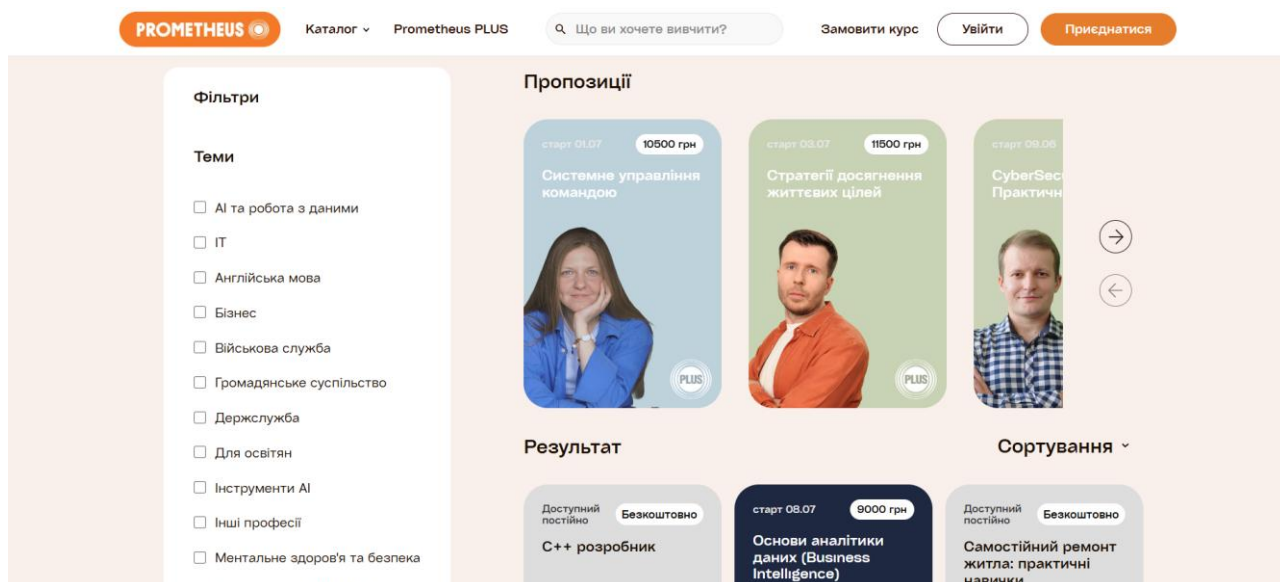


Рисунок 1.7. Інтерфейс сторінки з курсами на сайті Prometheus

Переваги Prometheus:

- безкоштовні курси від українських університетів та партнерів;
- контент українською мовою;
- зручний інтерфейс, підтримка мобільної версії;
- сертифікація після завершення курсу;
- теми, актуальні для українських реалій: громадянська освіта, підприємництво, зовнішнє оцінювання.

Недоліки Prometheus:

- обмежений набір курсів порівняно з міжнародними MOOC;
- менш потужні системи аналізу та оцінювання;
- більше орієнтована на індивідуальне навчання, ніж на повноцінну LMS.

Аналіз порівняльних характеристик дозволяє зробити наступні висновки. Кожна платформа має свої особливості, наприклад Moodle та Canvas підходить для складних і масштабних освітніх систем. Google Classroom – для полегшення навчання в школі. Coursera та Prometheus – для самостійного навчання та професійного розвитку. По-друге, основні відмінності між платформами полягають у рівні налаштування, стандартній підтримці, аналітичних функціях та моделі розробки. По-третє, жодна з платформ не є універсальною, вибір має

ґрунтуватися на педагогічній меті, технічних можливостях закладу та потребах аудиторії. Детальніше порівняння систем можна переглянути у таблиці 1.1.

Таблиця 1.1.

Порівняльна таблиця за основними критеріями

Платформа	Тип	Підтримка SCORM/ xAPI	Аналітика	Комуні-кація	Мобільність	Вартість
Moodle	LMS	+	+	+	+	Безкоштовна
Classroom	LMS (хмара)	-	-	+	+	Безкоштовна
Canvas	LMS	+	+	+	+	Платна
Edmodo	LMS/social	-	-	+	+	Безкоштовна
Coursera	MOOC	-	Часткова	Часткова	+	Умовно платна
Prometheus	MOOC	-	-	Часткова	+	Безкоштовна

Таблиця створена за джерелом [35]

Порівняльний аналіз демонструє, що жодна система не є універсальною. Кожна платформа має свої сильні сторони, що визначаються сферою застосування, цільовою аудиторією, технічною реалізацією та дидактичними цілями. Платформи LMS, такі як Moodle та Canvas, забезпечують глибоку інтеграцію навчального процесу, тоді як платформи MOOC, такі як Coursera або Prometheus, підкреслюють масовість та відкритість. Водночас Edmodo та Classroom зосереджуються на простоті, доступності та швидкій комунікації.

1.4. Обґрунтування необхідності розробки власної вебплатформи

Розвиток цифрової освіти у світі та в Україні призвів до стрімкого зростання інтересу до використання онлайн-платформ як ключового інструменту організації освітнього процесу. Аналіз сучасного ринку освітніх систем демонструє наявність великої кількості різних LMS, MOOC та соціально-освітніх середовищ. Однак, незважаючи на різноманітність функціональних

можливостей, більшість існуючих платформ мають суттєві обмеження, що перешкоджають ефективному впровадженню інноваційних підходів до навчання та не повністю відповідають вимогам конкретного освітнього середовища чи цільової аудиторії.

Одним з головних недоліків поширених платформ є їхня жорсткість та обмежена кастомізація, особливо у випадку безкоштовних або умовно-безкоштовних систем. Такі платформи, як Google Classroom або Edmodo, хоча й зручні у використанні, не дозволяють повністю змінити архітектуру курсу, налаштувати логіку оцінювання, впровадити аналітичні інструменти чи персоналізовані моделі навчання. Canvas та Moodle мають значні можливості кастомізації, але Moodle вимагає складного управління, а Canvas вимагає значних фінансових витрат.

Також існує потреба в локалізації контенту, зокрема, для адаптації мовного середовища та змісту курсів до українських стандартів, вимог та культурного контексту. Деякі платформи не підтримують українську мову або працюють з нею обмежено, що створює труднощі у роботі школярів та педагогів, а також знижує мотивацію до навчання. Створення власної онлайн-платформи дозволяє проектувати інтерфейс, навігацію, сповіщення та звіти з урахуванням освітньої реальності.

Варто відзначити відсутність платформи, яка б поєднувала всі необхідні функції в одній системі, а саме: управління освітнім контентом, гнучку систему комунікації, аналіз прогресу, персоналізацію, адаптивні механізми, соціальні елементи, а також зручність використання з мобільних пристроїв. Існуючі рішення спеціалізуються лише на окремих функціях – наприклад, Prometheus сильний у моделі MOOC, Google Classroom у комунікації, Moodle у середовищах курсів. Однак бракує комплексної універсальної системи, орієнтованої саме на потребу закладів загальної середньої освіти, яка б поєднувала всі ці функції.

РОЗДІЛ 2

РОЗРОБКА ТА РЕАЛІЗАЦІЯ ОСВІТНЬОЇ ВЕБПЛАТФОРМИ ДЛЯ УПРАВЛІННЯ НАВЧАЛЬНИМ КОНТЕНТОМ ТА КОМУНІКАЦІЇ

2.1. Постановка задачі та формування вимог до вебплатформи SmartClass

Основною метою розробки вебплатформи SmartClass є створення єдиного цифрового освітнього середовища, яке дозволить управляти навчальним контентом та надаватиме можливість ефективної комунікації між учнями і вчителями. Актуальність завдання зумовлена зростаючою потребою навчальних закладів у зручних, доступних та функціональних вебінструментах для підтримки дистанційного чи змішаного навчання. Платформа повинна забезпечувати безпечне місце для завантаження, розповсюдження та використання навчальних матеріалів, а також підтримувати взаємодію у межах навчальних класів.

Реалізувати завдання було вирішено за допомогою мови програмування C# та написаного для неї фреймворку Blazor. Після детального аналізу та огляду різних подібних вебзастосунків, були визначені найважливіші вимоги до проєкту:

- система має забезпечувати автентифікацію та авторизацію користувачів (аспект, який необхідний для забезпечення безпеки та конфіденційності інформації та полегшення взаємодії);
- підтримувати розділення ролей користувачів (вчитель, учень), що забезпечить розподіл можливостей учасників на платформі;
- забезпечувати можливість створювати та управляти навчальними контентом;
- підтримувати приєднання учнів до занять за допомогою унікального коду;
- надавати можливість роботи з навчальними матеріалами;

- забезпечувати систему сповіщення через оголошень та інформаційні повідомлення;
- підтримувати можливість комунікації у режимі реального часу за допомогою класних чатів;
- надавати доступ до особистого кабінету користувача.

На етапі формування функціональних вимог до платформи було визначено, що система повинна забезпечувати повний цикл взаємодії користувачів у рамках навчального процесу. Вчитель повинен мати можливість створювати навчальні класи, запрошувати на них учнів, розміщувати навчальні матеріали, публікувати оголошення, використовувати засоби комунікації та контролювати активність учасників. Учень у свою чергу повинен мати доступ до освітнього контенту, мати можливість переглядати оголошення, спілкуватися з педагогом та іншими учасниками заняття, а також керувати власним профілем.

У процесі формулювання було зосереджено увагу на нефункціональних характеристиках системи. Наприклад, вебплатформа повинна мати простий візуальний стиль та інтуїтивно зрозумілий інтерфейс, адаптивний дизайн для роботи на різних пристроях та бути стабільною в умовах одночасного доступу багатьох користувачів. Також до важливих нефункціональних вимог відноситься безпека персональних даних, захист від несанкціонованого доступу, стабільність серверної частини та можливість подальшого розширення функціоналу з додаванням нових інструментів.

Отже, завданням розробки вебплатформи SmartClass є створення доступного, функціонального та безпечного цифрового освітнього середовища, яке буде відповідати сучасним вимогам закладів середньої освіти, сприятиме підвищенню ефективності освітнього процесу та розвитку цифрових компетенцій. Сформовані вимоги стали основою для подальшого проєктування архітектури системи, вибору технологій та впровадження програмного продукту.

2.2. Вибір моделі та методології розробки програмного забезпечення

Сучасний процес розробки програмного забезпечення передбачає використання ефективної моделі життєвого циклу та відповідної методології, що забезпечать структурованість, надійність, масштабованість та можливість подальшого розширення і розвитку програмного продукту. Враховуючи ці вимоги, під час створення освітньої вебплатформи SmartClass було здійснено обґрунтований вибір моделі життєвого циклу програмного продукту та методологічний підхід до організації процесу розробки і архітектурних принципів системи.

Після детального вивчення для реалізації проєкту було вирішено обрати ітераційну модель розробки програмного забезпечення, яка передбачає поетапне створення функціональних компонентів системи з їх подальшим тестуванням, аналізом результатів та доопрацюванням. Такий підхід дозволяє отримувати робочі версії продукту на ранніх етапах роботи та поступово їх удосконалювати.

Використання ітераційної моделі доцільне для подібних вебплатформ, оскільки вимоги до функціоналу будуть змінюватися у процесі розробки, з'являтимуться нові потреби у користувачів, а також коригуються сценарії використання системи. Ітераційний підхід дозволяє швидко вносити зміни до програмного продукту без необхідності повного перегляду всієї логіки системи. Кожна наступна ітерація базується на результатах попередньої, що забезпечує поступове покращення якості. Цикл складається з власних окремих етапів або ж стадій, які по чергово виконуються у всіх повтореннях (рисунок 2.1).

Паралельно з ітераційною моделлю застосовувались принципи гнучкої методології розробки програмного забезпечення (Agile). Основна ідея такого підходу полягає в орієнтації на потреби користувача, швидкому реагуванні на зміну вимог, тісній взаємодії між розробником і кінцевим користувачем, а також створенні працездатного програмного продукту на кожному етапі роботи.



Рисунок 2.1. Принцип роботи ітераційної моделі розробки

Гнучка методологія передбачає поділ процесу розробки на короткі спринти, у рамках яких реалізується конкретний набір функціональних можливостей. Після завершення кожного з таких циклів отримані результати аналізуються, перевіряється їх функціональність та проводиться необхідне удосконалення. Даний підхід дозволив поступово впровадити основні функції платформи SmartClass, протестувати їх на практиці та покращити оптимізацію.

Використання методології Agile розробки особливо ефективно для освітніх систем, оскільки у процесі створення таких платформ часто змінюються функціональні вимоги, а через швидкий розвиток у цій сфері та швидкої появи різноманітних інструментів виникає необхідність розширення та їх додавання до системи. І саме гнучкий підхід дозволяє підтримати високу гнучкість та забезпечує швидкі зміни без втрати стабільності програмного продукту.

Особливу увагу було приділено організації процесу тестування у рамках кожного з етапів розробки. Після впровадження чергового модуля перевірялася його працездатність, відповідність вимогам та коректність взаємодії з іншими частинами системи. Це дозволило своєчасно виявляти помилки, усунути логічні неточності та забезпечити стабільність роботи програмного продукту.

Застосована модель та методологія розробки також сприяли ефективній організації роботи над проєктом. Процес створення програмного продукту був

чітко структурований, а кожен етап мав визначені цілі та результати. Це дозволило забезпечити системний підхід до розробки, уникнути хаотичних змін та зберегти послідовність у реалізації функціональності платформи.

Загалом, вибір ітераційної моделі життєвого циклу та використання гнучкої методології розробки, забезпечили ефективну організацію процесу створення вебплатформи SmartClass, гнучкість у внесенні змін та можливість його подальшого розвитку відповідно до потреб освітнього середовища.

2.3. Загальний опис проєкту та архітектури вебплатформи

Освітня вебплатформа SmartClass була реалізована як серверний вебзастосунок на базі технології Blazor Server, що забезпечує роботу інтерфейсу користувача безпосередньо на сервері з передачею стану в режимі реального часу до браузера за допомогою SignalR. Такий підхід дозволяє зменшити навантаження на клієнтську сторону, забезпечити централізовану обробку логіки і підвищити рівень безпеки даних.

Платформа SmartClass була розроблена відповідно до принципів Clean Architecture (чистої архітектури), що забезпечує чіткий розподіл відповідальності між рівнями системи, гнучку підтримку та можливість подальшого масштабування.

Уся система побудована як багаторівнева клієнт-серверна архітектура, в якій кожен модуль виконує певну функцію. Основним принципом цього типу архітектури є правило, що зовнішні ресурси залежать від внутрішніх, а не навпаки (рисунок 2.2).

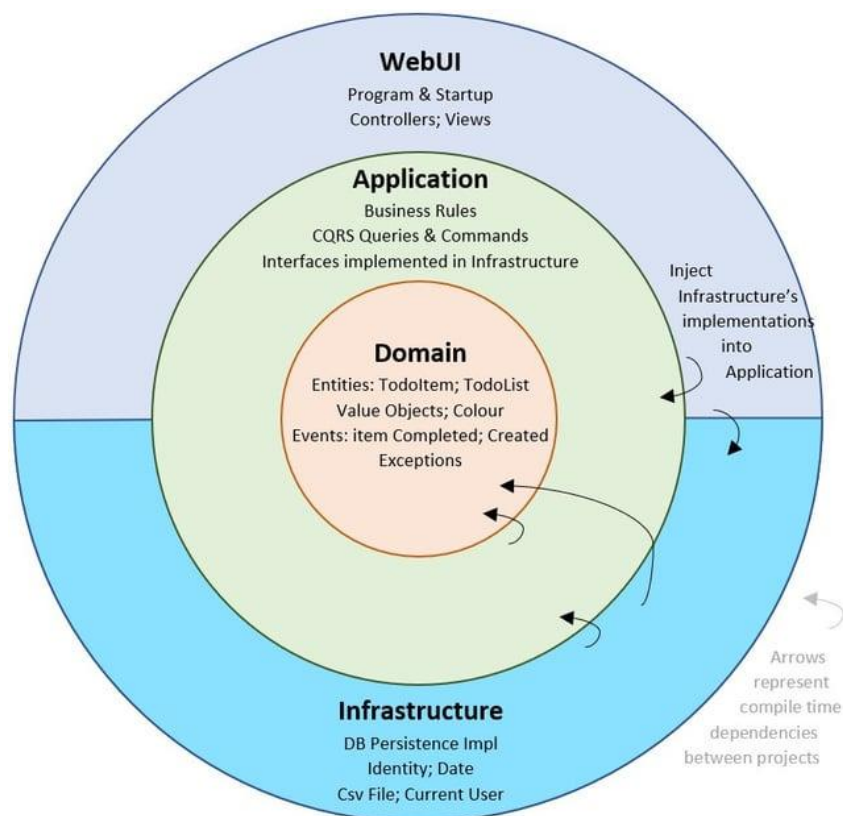


Рисунок 2.2 Схема використання Clean Architecture

Приклад реальної загальної структури проєкту можна подивитися на рисунку 2.3.

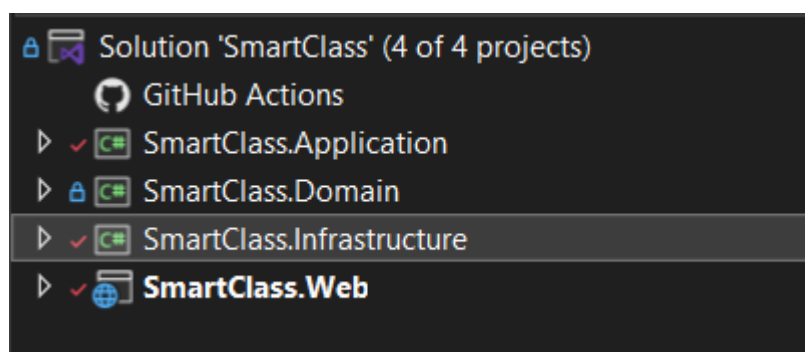


Рисунок 2.3. Вигляд структури проєкту SmartClass

1. Проєкт **SmartClass.Domain** містить бізнес-модель системи. Він описує основні сутності предметної області, такі як користувач, навчальний клас, учасник заняття, навчальні матеріали, повідомлення та інші об'єкти. Цей рівень

не має залежностей від жодних зовнішніх бібліотек чи фреймворків і лише визначає логіку предметної області.

2. Проєкт **SmartClass.Application** відповідає за реалізацію варіантів використання системи, визначає контракти взаємодії між шарами, містить об'єкти DTO (об'єкти передачі даних), інтерфейси сервісів, валідатори та логіку обробки команд і запитів. Саме на цьому рівні реалізується програмна логіка, яка координує роботу сутностей домену без прив'язки до конкретної реалізації інфраструктури.

3. Проєкт **SmartClass.Infrastructure** забезпечує технічну реалізацію доступу до даних та сервісів. Тут робота з базою даних здійснюється через Entity Framework Core, реалізована система автентифікації та авторизації користувачів, налаштовується робота з файлами, JWT-токенами та інші зовнішні інтеграції. Цей рівень залежить від програми, але не навпаки, що відповідає принципам чистої архітектури.

4. Проєкт **SmartClass.Web** – це клієнтський рівень вебплатформи, який реалізує інтерфейс користувача на основі Blazor Server, мінімальний API для взаємодії з бекенд сервісами, систему обміну повідомленнями в реальному часі через SignalR та конфігурацію контейнера для впровадження залежностей. Цей рівень відповідає за взаємодію користувача з застосунком.

Загальний принцип роботи полягає в тому, що всі дії користувача (реєстрація, вхід, створення класів, завантаження матеріалів, надсилання повідомлень) обробляються на сервері, а результати одразу передаються клієнту у вигляді оновлень інтерфейсу в режимі реального часу. Це забезпечує високу швидкість відгуку системи, синхронізацію станів та зручність для користувача. Архітектурне рішення SmartClass дозволяє ізолювати бізнес-логіку від реалізації технічних деталей, що значно покращує тестування, обслуговування та подальшу розробку. У майбутньому до платформи можна підключати додаткові функціональні модулі без необхідності повного оновлення всього рішення.

Головною перевагою проєкту є можливість подальшого розширення платформи з можливістю додавання електронних журналів, обліку успішності,

відеоконференцій, аналітики навчання та інтеграції із зовнішніми освітніми сервісами. Таким чином, SmartClass розглядається не лише як окремий програмний продукт, але й як основа для побудови комплексного цифрового освітнього середовища.

2.4. Обґрунтування вибору інструментальних засобів розробки

Розробка сучасного програмного продукту вимагає використання надійних, перевірених технологічно просунутих та зручних інструментів, що забезпечать високу якість застосунку, безпеку даних, масштабованість та простоту обслуговування. Під час розробки вебплатформи SmartClass було обрано набір сучасних інструментів, які повністю відповідають вище перерахованим технічним вимогам.

Основні інструменти, що використовувалися у процесі створення SmartClass, включають:

- інтегроване середовище розробки Visual Studio;
- мова програмування C#;
- серверний фреймворк Blazor Server;
- система керування базами даних Microsoft SQL Server;
- інструмент документації API та тестування Swagger;
- бібліотека компонентів інтерфейсу користувача Radzen.

Використання саме цього стеку технологій дозволило забезпечити повну реалізацію всіх основних функціональностей платформи SmartClass та створити стабільний, гнучкий та розширюваний програмний продукт.

2.4.1. Вибір Visual Studio для розробки веборієнтованої платформи

Інтегроване середовище розробки (IDE) є одним із ключових інструментів у створенні сучасних програмних продуктів, оскільки воно забезпечує зручність у написанні коду, налагодженні, тестуванні та підтримці програмного

забезпечення. Містить різноманітні додаткові функції, що спрощують розробку і написання коду. Visual Studio було обрано основним середовищем для розробки освітньої вебплатформи SmartClass, це є одне із найпотужніших та найпоширеніших середовищ для створення додатків на платформі .NET.

Вибір цієї IDE перш за все зумовлений її тісною інтеграцією з мовою програмування C# та .NET платформою. Це середовище забезпечує повну підтримку серверних вебдодатків, зокрема додатків на базі Blazor Server, що стала базовою технологією, яка використовувалася для реалізації продукту. Visual Studio дозволяє розробнику працювати з єдиним середовищем для клієнтської та серверної частин додатка, що значно підвищує ефективність розробки.

Однією з найважливіших переваг є наявність інтелектуальної системи автодоповнення коду (IntelliSense) та III помічника, що значно пришвидшує процес програмування, зменшує кількість синтаксичних помилок та дозволяє швидше орієнтуватися у великих програмних проєктах і виправляти помилки.

Visual Studio надає потужні інструменти налагодження (debugging), які дозволяють покроково проходити програмний код, переглядаючи значення змінних у режимі реального часу, аналізувати стек викликів та швидко виявляти логічні помилки. Завдяки цьому вдалося ефективно усувати помилки в логіці сервера, перевіряти правильність бізнес-операцій та взаємодію між компонентами системи SmartClass. Приклад інтерфейсу середовища розробки Visual Studio під час розробки вебплатформи (рисунок 2.4).

Середовище містить і розширені інструменти для роботи з багатопроєктними рішеннями, що є необхідною умовою для реалізації принципів чистої архітектури. У рамках одного рішення SmartClass.sln було організовано кілька взаємопов'язаних проєктів (Domain, Application, Infrastructure, Web), між якими забезпечувалася зручна навігація та прозора система залежностей і контроль за коректністю зв'язків.

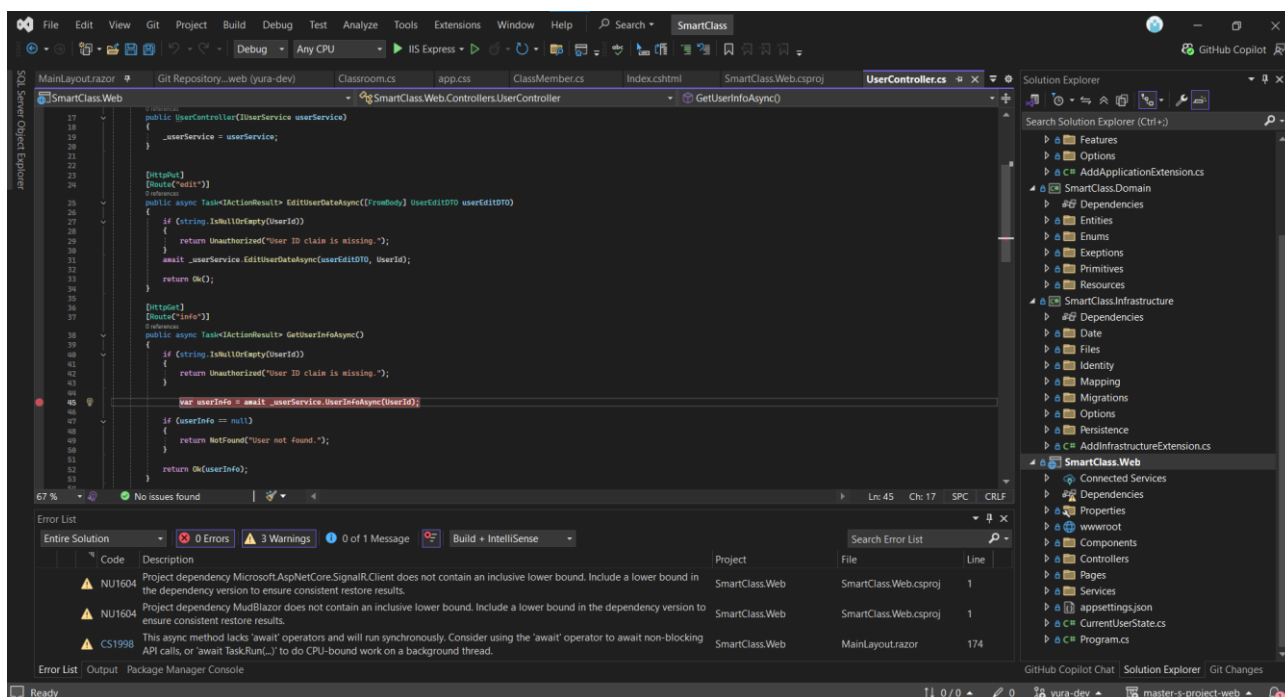


Рисунок 2.4. Інтерфейс середовища розробки Visual Studio під час створення вебплатформи SmartClass

Важливою перевагою Visual Studio є підтримка систем контролю версій, зокрема Git. Це дозволяє вести історію змін проекту, працювати з різними гілками розробки, повертатися до попередніх версій коду та безпечно експериментувати з новими функціями платформи без будь-якого ризику втратити стабільну версію програмного продукту.

Окрему роль у процесі розробки відіграють вбудовані інструменти управління залежностями, які дозволяють підключати сторонні бібліотеки через менеджер пакетів. Це дозволило швидко підключати та оновлювати більшість компонентів для роботи з базою даних, автентифікацією, дизайном та іншими сервісними модулями платформи. Варто відзначити зручні засоби запуску та тестування вебпрограми через середовище розробки. Це дозволяє перевірити роботу нових функцій у браузері, протестувати робочий інтерфейс, перевірити коректність виконання реєстрації користувачів, створення класів, завантаження матеріалів та роботи чатів.

Отже, вибір Visual Studio, як основного середовища розробки для вебплатформи SmartClass є повністю виправданим, оскільки воно забезпечує

повний цикл створення програмного продукту – від написання коду та налагодження до тестування, контролю версій та підготовки до розгортання. Використання цього середовища дозволило покращити якість реалізації програмного забезпечення, скоротити час розробки та забезпечити стабільність і зручність підтримки освітньої вебплатформи.

2.4.2. Застосування мови програмування C# та серверного фреймворку Blazor Server для розробки програмного засобу

Вибір мови програмування для реалізації вебплатформ є одним з ключових етапів розробки будь-якого програмного забезпечення, оскільки саме це буде визначати майбутню продуктивність, безпеку, масштабованість та простоту підтримки готового продукту. Для створення освітньої вебплатформи SmartClass базовою мовою програмування була обрана C#, а для реалізації вебінтерфейсу – серверний фреймворк Blazor Server.

C# – це сучасна об'єктно-орієнтована мова, яка широко використовується для розробки вебдодатків, серверних сервісів, настільних додатків та корпоративних інформаційних систем. Вона характеризується високою стабільністю, суворою типізацією, підтримкою асинхронного програмування та широкими можливостями реалізації складної бізнес-логіки. Це робить C# розумним вибором для створення масштабованих та надійних вебплатформ.

У рамках проєкту SmartClass C# використовується для реалізації всієї серверної логіки а саме для обробки запитів користувачів, автентифікації та авторизації, управління класами, взаємодії з базою даних, реалізації бізнес-правил а також для управління потоками даних між рівнями системи. Це дозволяє створити єдину логічну модель платформи без розділення логіки між різними мовами програмування.

Великою перевагою використання C# є його тісна інтеграція з платформою .NET, яка надає розробнику потужну стандартну бібліотеку, інструменти для роботи з мережею, базою даних, криптографією, файлами та потоками. Це

дозволяє реалізувати всі необхідні програмні механізми для коректної роботи без використання сторонніх або нестабільних рішень.

Для побудови інтерфейсу користувача був використаний серверний фреймворк Blazor Server, який дозволяє створювати інтерактивні вебінтерфейси з усією логікою, що виконується на стороні сервера. Однією з ключових переваг є можливість створення повноцінного інтерфейсу без використання JavaScript як основної мови програмування. Усі компоненти інтерфейсу, події, форми та логіка обробки реалізовані на C#, що покращує структуру проекту, зменшує кількість використовуваних технологій та забезпечує цілісність програмного коду. У такій моделі браузера користувачі виконують лише функцію відображення інтерфейсу та передачі подій на сервер, тоді як уся бізнес-логіка, обробка даних та стани управління забезпечуються на сервері.

Завдяки постійному з'єднанню між клієнтом і сервером, зміни в інтерфейсі користувача відбуваються миттєво без перезавантаження сторінки. Це особливо важливо для таких функцій, як чат, оновлення списків учасників, поява нових оголошень та взаємодія з навчальним контентом. Ще однією з важливих переваг використання Blazor Server у складі SmartClass є підвищений рівень безпеки. Оскільки бізнес-логіка не переноситься на сторону клієнта, а залишається на сервері, знижується ризик компрометації коду та несанкціонованого доступу до внутрішніх механізмів системи. Це критично важливо усіх платформ, які працюють з персональними даними користувачів.

Використання C# у поєднанні з серверним фреймворком Blazor Server дозволяє реалізувати єдину програмну платформу для всієї системи, в якій як серверна логіка так і інтерфейс користувача побудовані на одній мові програмування та єдиній технологічній платформі .NET. Це значно спростило процес розробки, налагодження та подальшої підтримки програмного продукту. Таким чином, вибір мови C# та серверного фреймворку Blazor Server для створення освітньої вебплатформи SmartClass є повністю виправданим як з технічної та функціональної так і з безпекової точок зору. Обрані технології

дозволили створити стабільний, безпечний та зручний у використанні програмний інструмент.

2.4.3. Застосування SQL Server для зберігання та обробки даних

Одним із ключових компонентів будь-якої сучасної інформаційної системи є база даних, що слугує центральним сховищем усієї інформації, необхідної для функціонування системи. Зокрема, вона зберігає облікові записи користувачів, їхні ролі та права доступу, інформацію про створені заняття, зв'язки учасниками, навчальні матеріали, повідомлення в чаті, оповіщення, а також службову інформацію, що забезпечує стабільну роботу платформи. З урахуванням необхідних умов було вирішено використовувати реляційну систему керування базами даних SQL Server. Однією з ключових переваг цієї системи є розширені механізми безпеки, що забезпечують захист персональних даних, розмежування прав доступу та контроль дій користувачів. Це особливо важливо для платформ, що працюють з інформацією про користувачів. Надійні засоби автентифікації, шифрування та адміністрування доступу дозволяють мінімізувати ризики витоку даних та несанкціонованого втручання в систему. Крім того, SQL Server забезпечує високу продуктивність роботи із запитамі, підтримує складні операції обробки даних та гарантує цілісність інформації. Механізми резервного копіювання та відновлення дозволяють підтримувати стабільність системи навіть у разі технічних збоїв. Масштабованість СУБД дозволяє використовувати її як для невеликих навчальних груп, так і для великих навчальних закладів з великою кількістю користувачів.

Великою перевагою використання SQL Server є підтримка реляційної моделі даних, яка дозволяє чітко структурувати інформацію у вигляді таблиць та встановлювати зв'язки між ними, забезпечувати цілісність даних та уникати дублювання інформації. Завдяки цьому платформа має логічно впорядковану структуру зберігання даних, що спрощує їх обробку та аналіз. Для доступу до бази даних у рамках проекту використовується підхід ORM (об'єктно-реляційне

відображення), який дозволяє взаємодіяти з таблицями бази даних за допомогою об'єктів мови програмування C#. Такий підхід значно зменшує обсяг ручного SQL-коду, знижує ймовірність помилок під час роботи з даними та підвищує безпеку обробки інформації. Розробник працює з даними у вигляді об'єктів, що є зручнішим та зрозумілішим у рамках великого програмного проекту. Зручні інструменти управління бази даних дозволяють контролювати її стан, оптимізувати запити, а також створювати резервні копії та відновлювати дані за потреби. Це важливо для забезпечення безперервної роботи платформи в умовах навчального процесу.

Під час розробки вебплатформи SmartClass також використовувалося локальне сховище баз даних, яке розгорнуто безпосередньо в середовищі розробки Visual Studio. Це дозволяє повноцінно протестувати всі функції обробки даних на етапі проектування без необхідності розгортання віддаленого сервера. Локальна база даних використовується для налагодження запитів, перевірки коректності збереження та отримання інформації, а також для тестування логіки взаємодії між програмними модулями.

Використання локального середовища сховища значно спрощує процес розробки, дозволяє швидко вносити зміни до структури бази даних, перевіряти працездатність нових функцій та зменшує споживання ресурсів на початкових етапах створення програмного продукту. Пізніше, після завершення етапу розробки та тестування, базу даних можна легко перенести на сервер для повноцінного впровадження в реальних умовах експлуатації.

Таким чином, використання SQL Server у поєднанні з локальним сховищем у середовищі Visual Studio забезпечує надійну, безпечну та зручну основу для зберігання та обробки даних для вебплатформи SmartClass, а також створює необхідні умови для її стабільної роботи та подальшого розвитку.

2.5. Особливості програмної реалізації

Розробка подібних до SmartClass проєктів вимагає ретельного планування структури коду, уважного ставлення до поділу відповідальностей між компонентами та використання сучасних підходів до організації програмної логіки. Основною метою було створити не просто робочий вебзастосунок, а платформу, яку буде можливо надалі розвивати, масштабувати та доповнювати новими модулями (електронним журналом, аналітикою, відеоконференціями тощо) без критичного втручання в уже реалізовані частини. Саме тому була обрана клієнт-серверна архітектура на базі Blazor Server з чітким поділом на кілька окремих шарів відповідно до принципів чистої архітектури.

Рішення організовано у вигляді єдиного рішення SmartClass.sln, яке включає чотири основні проєкти в каталозі src: SmartClass.Domain, SmartClass.Application, SmartClass.Infrastructure та SmartClass.Web приклад структури на рисунку 2.3. Подібний підхід чітко відображає логічний поділ між бізнес-моделлю, логікою програми, деталями інфраструктури та вебінтерфейсом. Архітектурна схема (рисунок 2.2) показує, як ці проєкти пов'язані один з одним: рівень домену ні від чого не залежить, рівень програми залежить лише від домену, інфраструктура реалізує контракти логіки програми, а вебрівень у свою чергу, використовує сервіси з нижчих рівнів через механізм впровадження залежностей.

Інтерфейс у SmartClass реалізовано за допомогою вищезгаданого Blazor Server, що дозволяє використовувати мову C# як для серверної логіки, так і для клієнтських компонентів. Це означає, що вся логіка обробки подій, взаємодії з даними, перевірки прав доступу та вмісту сторінок виконується на сервері, а вже сформований стан компонентів передається до браузера користувача. Серверна частина базується на ASP.NET Core, що забезпечує надійну основу для маршрутизації, автентифікації, роботи з базами даних та інтеграції з необхідними сторонніми бібліотеками.

Для зберігання та обробки даних у SmartClass використовується об'єктно-реляційний підхід, заснований на Entity Framework Core. Центром роботи з базою даних є клас `AppDbContext`, який успадковується від `IdentityDbContext` та об'єднує всі основні таблиці платформи: класи, учасники, навчальні матеріали, файли, завдання, оцінки, повідомлення, канали, зустрічі, учасники зустрічей, сповіщення та токени оновлення (рисунк 2.5). Метод `OnModelCreating` виконує детальне налаштування первинних ключів, унікальних індексів, зовнішніх ключів, каскадних правил видалення та обмежень довжини полів. Такий підхід забезпечує цілісність даних, усуває дублювання інформації та підвищує продуктивність запитів шляхом індексації полів, що найчастіше використовуються.

```
namespace SmartClass.Infrastructure.Persistence
{
    25 references
    public class AppDbContext
        : IdentityDbContext<ApplicationUser, ApplicationRole, Guid>
    {
        0 references
        public AppDbContext(DbContextOptions<AppDbContext> options) : base(options) { }

        // Core
        8 references
        public DbSet<Classroom> Classrooms => Set<Classroom>();
        0 references
        public DbSet<ClassMember> ClassMembers => Set<ClassMember>();
        0 references
        public DbSet<Material> Materials => Set<Material>();
        0 references
        public DbSet<FileResource> Files => Set<FileResource>();
        0 references
        public DbSet<Assignment> Assignments => Set<Assignment>();
        0 references
        public DbSet<Submission> Submissions => Set<Submission>();
        0 references
        public DbSet<Grade> Grades => Set<Grade>();
        1 reference
        public DbSet<Notification> Notifications => Set<Notification>();

        // Communications
        0 references
        public DbSet<Channel> Channels => Set<Channel>();
        0 references
        public DbSet<ChannelMember> ChannelMembers => Set<ChannelMember>();
        0 references
        public DbSet<Message> Messages => Set<Message>();
        0 references
        public DbSet<Meeting> Meetings => Set<Meeting>();
        0 references
        public DbSet<MeetingParticipant> MeetingParticipants => Set<MeetingParticipant>();

        // Auth
        4 references
        public DbSet<RefreshToken> RefreshTokens => Set<RefreshToken>();
    }
}
```

Рисунок 2.5. Оголошення класу `AppDbContext` з `DbSet` основних сутностей

Для організації доступу до даних у проєкті використовується узагальнений шаблон репозиторію, який реалізований у класі `BaseRepository<TEntity>` (рисунк 2.6). Він працює з будь-якими типами сутностей, використовуючи загальний `AppDbContext`, та забезпечує виконання базових CRUD-операцій уніфікованим способом. Особливістю цієї реалізації є використання специфікаційного підходу на основі бібліотеки `Ardalis.Specification`. Це дозволяє описувати складні запити до бази даних у вигляді окремих об'єктів специфікацій, що містять умови фільтрації, правила сортування та список властивостей навігації для завантаження. Завдяки цьому бізнес-логіка сервісів залишається чистою та не перевантаженою складними LINQ-запитами, а доступ до даних стає більш гнучким та масштабованим.

```
using Ardalis.Specification;
using Ardalis.Specification.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore;
using SmartClass.Application.Abstractions;
using SmartClass.Infrastructure.Persistence;
using System.Linq.Expressions;

namespace SmartClass.Infrastructure.Data.Repositories
{
    2 references
    public class BaseRepository<TEntity> : IRepository<TEntity> where TEntity : class
    {
        protected readonly AppDbContext _dbContext;
        protected readonly DbSet<TEntity> _dbSet;
        0 references
        public BaseRepository(AppDbContext dbContext) ...
        3 references
        public async Task<TEntity> GetFirstBySpecAsync(ISpecification<TEntity> specification) ...
        1 reference
        private IQueryable<TEntity> ApplySpecification(ISpecification<TEntity> specification) ...
        1 reference
        public async Task<IEnumerable<TEntity>> GetAllAsync() ...
        11 references
        public async Task<TEntity> GetByKeyAsync(TKey key) ...
        1 reference
        public async Task<TEntity> GetByPairOfKeysAsync(TFirstKey, TSecondKey
            (TFirstKey firstKey, TSecondKey secondKey) ...
        8 references
        public async Task<TEntity> AddAsync(TEntity entity) ...
        4 references
        public async Task UpdateAsync(TEntity entity) ...
        2 references
        public async Task DeleteAsync(TEntity entity) ...
        /* public async Task DeleteRangeAsync(IEnumerable<TEntity> entities) ...
        11 references
        public async Task<int> SaveChangesAsync() ...
        1 reference
        public async Task AddRangeAsync(List<TEntity> entities) ...
        18 references
        public async Task<IEnumerable<TEntity>> GetListAsync(
            Expression<Func<TEntity,
            bool>> filter = null,
            Func<IQueryable<TEntity>,
            IOOrderedQueryable<TEntity>> orderBy = null,
            string includeProperties = null) ...
        1 reference
        public async Task<TEntity> GetEntityAsync(
            Expression<Func<TEntity, bool>> filter = null,
            string includeProperties = null) ...
    }
}
```

Рисунок 2.6. Узагальнений репозиторій для роботи з базою даних

У BaseRepository всі базові операції з базою даних реалізовані в асинхронному режимі. Методи отримання списків (GetAllAsync, GetListAsync), пошуку за ключем (GetByKeyAsync, GetByPairOfKeysAsync), додавання (AddAsync, AddRangeAsync), оновлення (UpdateAsync), видалення (DeleteAsync) та збереження змін (SaveChangesAsync) використовують механізм async/await. Це дозволяє виключити блокування потоків виконання під час запитів до бази даних та забезпечує кращу продуктивність системи, коли велика кількість користувачів працює одночасно, що типово для освітнього середовища.

Інтерфейс користувача SmartClass побудовано на компонентах Blazor з використанням бібліотеки компонентів Radzen. Кожна сторінка реалізована як окремий компонент Razor з власною маршрутизацією, логікою та пов'язаними залежностями. Наприклад, сторінка логізації користувача містить поля для введення персональних даних, компонент пароля, компонент для емайлу, кнопку надсилення форми та посилання навігації на сторінку реєстрації (рисунок 2.7).

```
@page "/login"
@using Radzen
@using SmartClass.Application.Contracts.Auth
@using SmartClass.Web.Services
@inject AuthApiClient AuthApi
@inject CurrentUserState CurrentUser
@inject NavigationManager Navigation
@inject Radzen.NotificationService Toast
@inject IJSRuntime JS

<EditForm Model="@UserLogin" OnValidSubmit="@HandleLogin">
  <div class="row">
    <div class="col-md-6 offset-md-3 col-lg-4 offset-lg-4">
      <section>
        <h2 class="text-center text-login">Use a local account to log in.</h2>
        <hr />
        <div class="form-group m-2">
          <label class="text-login">Email</label>
          <RadzenTextBox @bind-Value="userLogin.Email" Placeholder="name@example.com" class="form-control" Name="Email" />
        </div>
        <div class="form-group m-2">
          <label class="text-login">Password</label>
          <RadzenPassword @bind-Value="userLogin.Password" Placeholder="Password" class="form-control" Name="Password" />
        </div>
        <div class="form-check mb-3">
          <RadzenCheckBox @bind-Value="rememberMe" class="form-check-input" />
          <label class="form-check-label text-login">Remember me</label>
        </div>
        <div class="text-center">
          <RadzenButton Text="Log in" ButtonType="ButtonType.Submit" Icon="login" class="btn bg-secondary btn-block" />
        </div>
        <div class="text-center mt-3">
          <RadzenLink Style="color: white" Text="Forgot your password?" Path="/forgot-password" />
        </div>
        <div class="text-center">
          <RadzenLink Style="color: white" Text="Register as a new user" Path="/register" />
        </div>
        <div class="text-center">
          <RadzenLink Style="color: white" Text="Resend email confirmation" Path="/resend-email-confirmation" />
        </div>
      </section>
    </div>
  </div>
</EditForm>
```

Рисунок 2.7. Фрагмент компонента Razor на сторінці логіну

Логіка обробки форми реалізована в коді компонента та включає генерацію об'єкта DTO, надсилання HTTP-запиту до API та обробку відповіді сервера з відображенням повідомлень про успіх або помилки (рисунки 2.8).

```
@code {
    private LoginDto userLogin = new LoginDto();
    private bool rememberMe;

    private async Task HandleLogin()
    {
        try
        {
            var tokens = await AuthApi.LoginAsync(userLogin);
            if (tokens == null)
                throw new Exception("Login failed: empty token response");

            // 1. Зберегти токени в localStorage
            await JS.InvokeVoidAsync("localStorage.setItem", "accessToken", tokens.AccessToken);
            await JS.InvokeVoidAsync("localStorage.setItem", "refreshToken", tokens.RefreshToken.Token);

            // 2. Поставити Bearer у HttpClient, який використовує AuthApiClient
            AuthApi.SetBearer(tokens.AccessToken);

            // 3. Отримати поточного користувача з api/User/info
            var me = await AuthApi.GetCurrentUserAsync();
            if (me != null)
            {
                CurrentUser.SetAuth(tokens, me);
            }

            // 4. Повідомлення про успішний вхід
            Toast.Notify(new NotificationMessage
            {
                Severity = NotificationSeverity.Success,
                Summary = "Login successful",
                Detail = "Welcome back!",
                Duration = 3000
            });

            // 5. Перейти на головну
            Navigation.NavigateTo("/");
        }
        catch (Exception ex)
        {
            Toast.Notify(new NotificationMessage
            {
                Severity = NotificationSeverity.Error,
                Summary = "Error",
                Detail = ex.Message,
                Duration = 5000
            });
        }
    }
}
```

Рисунки 2.8. Опис логіки обробки форми сторінки логіну

Однією з ключових особливостей програмної реалізації SmartClass є обов'язкова наявність окремого чату для кожного класу. Для цього розроблено механізм обміну даними у реальному часі на основі SignalR (рисунки 2.9). Для кожного з класів автоматично створюється окремий канал зв'язку, завдяки чому доступ до нього можуть отримати лише учасники. Повідомлення, надіслані в чат, миттєво транслюються всім підключеним користувачам і одночасно зберігаються у базі даних. Це забезпечує як ефект живого спілкування, так і можливість відновлення історії повідомлень після повторного входу в систему.

```

0 references
public async Task SendMessage(Guid channelId, string text)
{
    if (string.IsNullOrEmpty(text))
        return;

    var userId = GetUserId();

    var member = await channelMembersRepo.GetEntityAsync(
        m => m.ChannelId == channelId && m.UserId == userId);

    if (member == null)
        throw new HubException("You are not a member of this channel.");

    text = text.Trim();

    var msg = new Message
    {
        ChannelId = channelId,
        AuthorId = userId,
        Text = text,
        CreatedAt = DateTime.UtcNow
    };

    await messagesRepo.AddAsync(msg);
    await messagesRepo.SaveChangesAsync();

    var dto = new ChatMessageDto(
        msg.Id,
        msg.ChannelId,
        msg.AuthorId,
        GetUserName(),
        msg.Text,
        msg.CreatedAt
    );

    await Clients.Group(ChannelGroup(channelId))
        .MessageReceived(dto);
}

```

Рисунок 2.9. Реалізація методу надсилання повідомлень у SignalR-хабі

Впровадження чатів є основою для подальшого розвитку платформи, оскільки відеоконференції, голосові кімнати, автоматичні сповіщення та інтеграція із зовнішніми сервісами онлайн-навчання можуть бути побудовані на одній інфраструктурі. Таким чином, обраний підхід закладає технологічну основу для подальшого розширення функціональності SmartClass.

Система активно використовує механізм впровадження залежностей для реєстрації сервісів бізнес-логіки, репозиторіїв, контексту бази даних, сервісів автентифікації, сигнальних хабів, сервісів сповіщень та допоміжних бібліотек. Це зменшує заплутаність у зв'язку між компонентами, спрощує тестування та робить структуру більш надійною.

Отже, особливостями програмної реалізації SmartClass є поєднання серверного підходу Blazor, чистої архітектури, асинхронної обробки запитів, гнучкого доступу до даних через Entity Framework Core та узагальненого репозиторію зі специфікаціями, використання компонентного інтерфейсу користувача та реалізація обов'язкових чатів у реальному часі у класі. Всі ці

рішення забезпечують високу продуктивність, безпеку, зручність використання та створюють надійну основу для подальшого розвитку платформи в умовах реального освітнього процесу.

2.6. Організація тестування та налагодження програмного засобу

Перш за все, тестування проводилося на рівні сервера шляхом перевірки роботи методів API через середовище Swagger UI. Після створення кожного нового контролера та відповідних методів перевірялася коректність обробки HTTP-запитів типу GET, POST, PUT та DELETE. Це дозволило перевірити коректність реєстрації користувачів, авторизації, створення навчальних класів, приєднання учнів за кодом доступу, розміщення навчальних матеріалів, створення завдань, обробки заявок, виставлення оцінок та відправлення повідомлень у чаті. Особлива увага приділялася перевірці обробки помилкових запитів, перевірці доступу відповідно до ролей користувачів та обмеженню несанкціонованих дій.

Основним каналом взаємодії між клієнтською та серверною частинами платформи є HTTP-запити до API сервера. Всі передані та отримані дані обробляються у форматі JSON, що забезпечує універсальність обміну інформацією та можливість подальшої інтеграції з іншими системами. Після успішного тестування логіки API було перевірено коректність обробки даних у інтерфейсі користувача.

Фронтенд платформи SmartClass побудовано з використанням бібліотеки Radzen для Blazor, яка надає набір готових компонентів інтерфейсу, таких як поля введення, списки, діалоги, кнопки, сповіщення, таблиці та модальні вікна. Це дозволило нам значно скоротити час розробки та зосередитися на логіці взаємодії користувача з системою.

Для коректного відображення сторінок, адаптації до різних розмірів екрана та забезпечення зручної навігації було використано власну таблицю стилів CSS та базову розмітку HTML.

У процесі тестування перевірялася коректність реалізації основних функціональних можливостей платформи SmartClass, а саме: робота реєстрації та авторизації користувачів, створення навчальних класів та приєднання до них за допомогою унікального коду доступу, робота з навчальними матеріалами, створення та управління завданнями, подання учнівських робіт, виставлення оцінок, а також спілкування між учасниками через чати класів у режимі реального часу. Окремо перевірялася коректність роботи особистих облікових записів користувачів, зберігання їх даних.

Тестування також включало перевірку стабільності системи під час одночасного доступу кількох користувачів, що особливо важливо в умовах реального навчального процесу, коли до одного класу може бути підключена значна кількість учнів.

Під час тестування було налагоджено обробку помилок, оптимізовано обробку запитів та покращено продуктивність окремих компонентів.

Результати роботи основних модулів платформи SmartClass можна побачити на скріншотах інтерфейсу, наведених у Додатку Б. У ньому подано: сторінку реєстрації аканту користувача, сторінку входу у особистий кабінет, головну сторінку сайту, сторінку зі списком навчальних класів користувача, вигляд вікна з чатом для учасників класу, приклад сторінки панелі оцінювання для вчителя, приклад персональної сторінки з оцінками учня, сторінку з матеріалом та сторінку із завданнями.

2.7. Рекомендації по використанню та впровадженню програмного засобу

Освітня вебплатформа SmartClass розроблена на базі технологій Blazor Server та ASP.NET Core і призначена для використання в закладах середньої освіти для підтримки очного, дистанційного та змішаного навчання. Наразі вона реалізована як локальний програмний продукт і не розміщена на публічному вебхостингу. Платформа запускається та використовується на персональному комп'ютері або сервері навчального закладу за умови наявності відповідного

програмного забезпечення. Такий підхід дозволяє забезпечити контроль даних, підвищити безпеку та адаптувати систему до умов конкретного освітнього середовища.

Платформа SmartClass надає можливість керування навчальними класами, розміщення та використання навчальних матеріалів, організації завдань, перевірки результатів навчання, а також забезпечує ефективну комунікацію між учнями та вчителями.

Для коректної роботи програмного забезпечення користувач повинен мати стабільне підключення до Інтернету та сучасний веббраузер. Перед початком роботи він повинен створити власний обліковий запис у системі та увійти в систему. Після авторизації вчитель має можливість створювати навчальні класи, створювати навчальний контент, додавати завдання, перевіряти подані роботи та проводити оцінювання. Після входу в систему учень може приєднатися до відповідного класу, використовуючи унікальний код доступу, переглядати матеріали, виконувати завдання, спілкуватися з вчителем та іншими учасниками класу через чат, а також відстежувати власні результати навчання.

Для впровадження платформи SmartClass на новому пристрої або сервері необхідно встановити середовище розробки Microsoft Visual Studio з підтримкою ASP.NET та веброботки, платформу .NET та систему керування базами даних Microsoft SQL Server. Програму можна запускати як на операційних системах Windows, так і на macOS за умови відповідного налаштування середовища виконання.

Рекомендації по використанню вебплатформа SmartClass (інструкція користувача)

1. Загальна інформація про програмне забезпечення. Освітня вебплатформа SmartClass призначена для організації освітнього процесу в умовах онлайн та офлайн навчання. Вона забезпечує управління класом, розміщення навчальних матеріалів, створення та виконання завдань, оцінювання результатів навчання та проведення комунікативних тестів між учнями та вчителями у режимі реального часу.

На поточному етапі SmartClass функціонує як локальний вебзастосунок, який може працювати на персональному комп'ютері або сервері навчального закладу та не потребує публічного доступу до Інтернету.

2. *Вимоги користувача та технічні умови.* Для запуску та належної роботи платформи SmartClass користувачу знадобиться:

- операційна система Windows або macOS;
- середовище розробки Microsoft Visual Studio;
- встановлений модуль ASP.NET та веб-розробки;
- платформа .NET;
- система керування базами даних Microsoft SQL Server;
- сучасний веббраузер (Google Chrome, Microsoft Edge тощо);
- стабільне локальне або мережеве з'єднання.

3. *Запуск програмного засобу.* Для запуску платформи необхідно:

- відкрити рішення SmartClass.sln у Microsoft Visual Studio;
- перевірити підключення бази даних до файлу конфігурації;
- запустити проект SmartClass.Web;
- після запуску програма доступна у веббраузері за адресою локального сервера;
- після запуску користувач може продовжити роботу на платформі через вебінтерфейс.

4. *Реєстрація і вхід до системи.* Щоб розпочати, потрібно створити обліковий запис. Користувач переходить на сторінку реєстрації, заповнює особисті дані (ім'я, прізвище, електронну пошту, пароль, дату народження, роль) та підтверджує реєстрацію.

Після успішної реєстрації користувач входить у систему, використовуючи ім'я користувача та пароль. Далі після авторизації отримує доступ до функціоналу відповідно до своєї ролі (вчитель чи учень).

5. *Функціональні можливості вчителя.* Вчитель має можливість:

- створювати класи;
- додавати навчальні матеріали;

- створювати завдання;
- перевіряти роботу учнів;
- виставляти оцінки;
- спілкуватися через класний чат.

6. Функціональні можливості учня. Учень може:

- приєднатися до заняття за допомогою коду доступу;
- переглядати навчальні матеріали;
- виконувати завдання;
- надсилати роботу на перевірку;
- переглядати оцінки;
- спілкуватися в чаті.

7. Робота з чатами. Кожен клас має власний чат для обміну повідомленнями у режимі реального часу. Використовується для консультацій, уточнення завдань та організації навчальної взаємодії.

8. Особистий кабінет користувача. У своєму особистому кабінеті користувач може:

- переглядати персональні дані;
- змінювати інформацію облікового запису;
- переглядати активні заняття;
- відстежувати свою навчальну діяльність.

9. Правила безпечної роботи. Користувачам рекомендується:

- не передавати свої паролі третім особам;
- використовувати складні паролі;
- виходити з системи після використання;
- використовувати платформу лише з перевірених пристроїв.

Платформу SmartClass рекомендується використовувати, як допоміжний інструмент для організації навчального процесу під час занять, виконання домашніх завдань, індивідуальної та групової роботи, а також для підтримки дистанційного навчання. Доцільно впроваджувати платформу як елемент єдиного інформаційно-освітнього середовища навчального закладу з

поступовим розширенням функціональності відповідно до потреб педагогів та учнів. Практичне використання SmartClass в освітньому процесі сприяє покращенню організації освітньої діяльності, зручності доступу до матеріалів, покращенню комунікації між учасниками освітнього процесу та розвитку цифрових компетенцій учнів.

Апробація результатів дослідження відбулася на базі Старопочаївської загальноосвітньої школи I-III ступенів Почаївської міської ради Тернопільської області під час переддипломної практики. З учнями було проведено анкетування щодо оцінки роботи вебзастосунку SmartClass за основними функціональними та інтерфейсними характеристиками. В опитуванні взяли участь 18 респондентів, які дали відповідь на 13 запитань за чотирирівневою шкалою: «дуже добре», «добре», «середнє», «погано». Зразок анкети подано у додатку В.

Узагальнені результати тестування (Додаток Г) підтверджують дуже високий рівень зручності, стабільності та функціональної придатності розробленого вебзастосунку SmartClass. Переважна більшість респондентів оцінили всі ключові функції на рівні «дуже добре» та «добре», що свідчить про якісну реалізацію інтерфейсу, логіку роботи системи та її адаптивність.

ВИСНОВКИ

У кваліфікаційній роботі проаналізовано сучасний стан та особливості систем управління навчанням, здійснено огляд та порівняльний аналіз освітніх вебплатформ. На основі аналізу визначено основні вимоги до проєктування та розробки власної навчальної платформи, включаючи функціональні, нефункціональні, педагогічні та технічні аспекти.

Основним результатом дослідження є проєктування та розробка програмного забезпечення SmartClass – освітньої вебплатформи, призначеної для управління заняттями, навчальними матеріалами, завданнями з можливістю комунікації між учасниками освітнього процесу. Під час розробки реалізовано ключові програмні модулі, зокрема, систему реєстрації та авторизації користувачів, обмеження доступу на основі ролей, створення та управління класами, приєднання учнів за кодом доступу, роботу з навчальними матеріалами, завданнями та оцінюванням, систему обміну повідомленнями, а також комунікацію в класі через чат.

У процесі розробки програмного продукту використано сучасні методи проєктування, які використовують принципи чистої архітектури, асинхронної обробки запитів, моделей взаємодії клієнт-сервер та компонентного підходу до проєктування інтерфейсу. Реалізовано безпечне зберігання даних, зберігання інформації в базах даних та механізми захисту персональної інформації користувачів.

Особливу увагу приділено педагогічним аспектам використання платформи. Впроваджена система дозволяє проводити навчальну діяльність у зручному цифровому середовищі, допомагає підвищити мотивацію учнів до навчання, формувати їхні цифрові навички, розвивати самостійність та відповідальність за результати навчання. Для вчителя платформа створює можливості оперативного контролю знань, зворотного зв'язку з учнями та зручного управління навчальним контентом.

Тестування платформи в умовах реального освітнього процесу підтвердили її практичну доцільність, операційну стабільність та відповідність вимогам. Використання SmartClass допомагає покращити організацію навчальної діяльності, комунікацію між учасниками навчального процесу та ефективність роботи вчителя за допомогою навчальних інструментів.

Розроблена освітня вебплатформа SmartClass має великий потенціал для подальшого розвитку. Перспективними напрямками для вдосконалення можуть бути: впровадження електронного журналу, розширених модулів аналізу успішності, системи відеоконференцій, інтеграції з іншими освітніми сервісами та використання систем рекомендацій.

Таким чином, мета кваліфікаційної роботи досягнута, поставлені завдання виконані, а отримані результати мають теоретичне та практичне значення для використання у закладах загальної середньої освіти. Розроблену платформу SmartClass можна використати як інструмент для підтримки дистанційного та змішаного навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ASP.NET core blazor. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/aspnet/core/blazor>
2. Codinblue. Perform CRUD operations using ASP.NET - razor pages - entity framework and SQL server database, 2023. YouTube. URL: <https://www.youtube.com/watch?v=nlR1MxHl1O0>.
3. SQL server 2022 | microsoft. Your request has been blocked. This could be due to several reasons. URL: <https://www.microsoft.com/uk-ua/sql-server/sql-server-2022>.
4. Work with components | radzen blazor studio docs. Radzen.com. URL: <https://www.radzen.com/blazor-studio/documentation/design-time/components>.
5. Бардадим О. В. Класифікація освітніх вебресурсів. *Наукові записки. Серія: Педагогічні науки*. 2022. Вип. 207. С. 89-99.
6. Биков В. Ю., Буров О. Ю. Цифрове навчальне середовище: нові технології та вимоги до здобувачів знань. *Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців, методологія, теорія, досвід, проблеми* : Збірник наукових праць. Вінниця : ТОВ «Друк плюс», 2020. Вип. 55. С.11-22.
7. Биков В. Ю., Шишкіна М. П. Теорія і практика розвитку цифрових компетентностей майбутніх учителів інформатики : монографія. Київ : Інститут інформаційних технологій і засобів навчання НАПН України, 2021. 287 с.
8. Васильченко Л., Шацька Н. Дистанційне (онлайн) навчання у закладах зсо: проблеми, виклики, рішення. *Education. Innovation. Practice*. 2022. Т. 10, № 7. С. 19-24.
9. Гайтан О. М. Порівняльний аналіз можливостей використання інструментарію вебінарорієнтованих платформ Zoom, Google Meet та Microsoft Teams в онлайн-навчання. *Інформаційні технології і засоби навчання*. 2022. Т. 87. №1. С. 33-67.

10. Галушка Ю. А. Використання сучасних онлайн-платформ в освітньому середовищі. *Сучасна освіта і наука Волині* : матеріали Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених (м. Луцьк, 22 травн. 2025р.). Луцьк: ФОП Мажула Ю. М., 2025. С.268-269

11. Галушка Ю. А., Ройко Л. Л. Освітня платформа як інструмент ефективної взаємодії учасників навчального процесу. *Математика. Інформаційні технології. Освіта* : збірник тез доп. XIV міжнар. наук.-практ. конф. (м. Луцьк, 13-15 червн. 2025 р.). Луцьк, 2025. С. 180-181

12. Галушка Ю. А., Ройко Л. Л. Роль сучасних освітніх платформ для колективної взаємодії учасників навчального процесу. *Соціально-гуманітарний вісник* : зб. наук. праць. Харків : СГНТМ «Новий курс». 2025. Вип.61. С.25-26

13. Гончарук Л. В. Інформаційні системи та технології в освіті. Київ : Професійний розвиток. 2023. 160 с.

14. Грабовський П. О. Сучасні підходи до розробки інтерфейсів користувача. *Наукові праці*. Київ : КНУ. 2022. № 2. С. 25-30.

15. Дзень В. Є., Кунинець М. С. Розробка та дослідження інноваційної технології мобільного навчання для підготовки розробників програмного забезпечення. *Вісник ЛДУБЖД*, 2020. С. 32

16. Дмитрієва Н. Б., Крикляс В. Г., Крикляс К. В., Чернякова Ж. Ю., Пенев В. В. Методи активного навчання: особливості використання в дистанційній освіті. *Вісник науки та освіти*. 2024. № 6(24). С. 596-610.

17. Друшляк М. Г., Семенов О. М., Грона Н. В., Пономаренко Н. П., Семеніхіна О. В. Типологія інтернет-ресурсів для розвитку інфомедійної грамотності молоді. *Інформаційні технології і засоби навчання*. 2022. Т. 88. №2. С. 1-22.

18. Іванова С. М., Вакалюк Т. А., Кільченко А. В., Новицька Т. Л. Технологія застосування сервісу Google analytics як інструменту моніторингу та підвищення ефективності використання освітніх вебресурсів. *Інформаційно-комунікаційні технології в освіті*. 2022. Вип. 49. Т. 1. С. 1-6.

19. Карабін О. Й., Поморський Д. В. Формування основ веб орієнтованих технологій на уроках інформатики в учнів старших класів закладів загальної середньої освіти. *Інноваційна педагогіка*. Одеса : Гельветика, 2020. Вип. 30. Т. 1. С. 53-57.
20. Коваль О., Умрик М. Практичний досвід впровадження авторської онлайн платформи в освітній процес закладу загальної середньої освіти. *Фізико-математична освіта*. 2025. Том 40, №1. С.34-41
21. Корчова Г. Л. Дистанційне та змішане навчання: теоретичний аспект. Педагогіка формування творчої особистості у вищій і загальноосвітній школах. Запоріжжя : КПУ, 2021. Вип. 76, Т. 2. С. 77-80.
22. Кривонос І. О. Особливості використання інформаційних технологій в освітній діяльності здобувачів освіти. *Українські студії в європейському контексті* : зб. наук. праць. 2022. № 5. С. 183-189.
23. Кухаренко В. М., Олійник Н. О. Технології дистанційного навчання : навч. посіб. Харків : В-во «НТМТ», 2022. 302 с.
24. Мар'єнко М. В., Шишкіна М. П., Коновал О. А. Методологічні засади формування хмаро орієнтованих систем відкритої науки у закладах вищої педагогічної освіти. *Інформаційні технології і засоби навчання*. 2022. Т. 89 № 3. С. 209-232.
25. Мартін Р. Ч. Чистий код. Створення і рефакторинг за допомогою Agile / пер. з англ. І.Бондар-Терещенко. Харків : Вид-во «Ранок»: Фабула, 2024. 448 с.
26. Меншиков В. В. Особливості впровадження хмарних сервісів та платформ у закладах загальної середньої освіти в умовах пандемії та воєнного стану. *Інформаційні технології в освіті*. 2023. № 3 (57). С. 57-66.
27. Морзе Н., Нанаєва Т., Пасічник О. Стан та перспективи навчання інформатики в закладах загальної середньої освіти України. *Інформаційні технології і засоби навчання*. 2022. Том 92. №6. С. 1-20
28. Мося І. А. Розвиток візуальної компетентності студентів-педагогів в умовах цифрового освітнього середовища. *Педагогіка і психологія*. 2024. № 1 (82). С. 121-129.

29. Мякшин А. С. Роль інформаційно-комунікаційних технологій у сучасній освіті. Матеріали VIII міжнародної науково-практичної інтернет-конференції (м. Київ, 26 січн. 2022 р.). Київ, 2022. С. 67-69.

30. Орлова А. А. Цифрові освітні технології: переваги, недоліки, перспективи впровадження. *Інновації та інтеграція цифрових трендів освітянського простору в економіку знань*. 2023. С.160-164

31. Платонова О. Використання інтернет додатків та платформ для організації освітнього процесу у закладах вищої освіти. *Вісник Національного університету «Чернігівський колегіум» імені Т. Г. Шевченка*. 2024. № 29-30 (185-186). С.125-130

32. Петренко Ю. А., Петренко В. С. Сучасні системи управління навчанням: огляд ринку. *Інформаційні технології в сучасному світі* : матеріали міжнародної науково-практичної здобувачів вищої освіти і молодих учених (м. Харків, 29 квітн. 2025 р.). Харків: ДБТУ, 2025. С. 427-430

33. Рибчук А. В., Журба І. Є., Процишин О. Р. Цифрова трансформація глобального освітнього середовища. *Вісник Хмельницького національного університету. Економічні науки*. 2022. № 1. С. 262-268.

34. Семеріков С. О., Поліщук В. В. Методика формування ІКТ-компетентностей майбутніх учителів засобами хмарних технологій : навч. посіб. Київ : Видавничий дім «Слово», 2020. 180 с.

35. Сірський Л. Вплив платформ для онлайн-курсів на навчання в школах. Освіта.UA. URL: <https://osvita.ua/news/89291/>.

36. Сисоєва С. О. Інноваційні технології навчання: теорія і практика. Київ : Освіта, 2019. 256 с.

37. Стечишин І. Системи управління навчанням: сучасні тенденції та перспективи розвитку. *Вісник науки та освіти*. 2025. № 12(30). С. 1203-1214.

38. Сучасні онлайн платформи та сервіси для організації дистанційного навчання. Освітній проект «На Урок» для вчителів. URL: <https://naurok.com.ua/suchasni-onlayn-platformi-ta-servisi-dlya-organizaci-distanciynogo-navchannya-382390.html>.

39. Сучасні цифрові технології та інноваційні методики навчання: досвід, тенденції, перспективи : матеріали XV Міжнар. наук.-практ. інтернет-конф. (м. Тернопіль, 10 квітня 2025 року). Тернопіль : ТНПУ імені Володимира Гнатюка, 2025. 328 с.

40. Твердохліб І. Дослідження факторів вибору мови програмування для вивчення в закладах загальної середньої освіти. *Проблеми сучасного підручника*. 2024. №32. С.284-298

41. Твердохліб І. А., Семко Л. П. Роль і місце задач прикладного спрямування в шкільному курсі інформатики. *Сучасні цифрові технології та інноваційні методики навчання: досвід, тенденції, перспективи* : матеріали IX Міжнародної науково-практичної інтернет-конференції (м. Тернопіль, 28 квітн. 2022). Тернопіль. 2022. С. 162-164.

42. Толочко С. В. Інноваційні технології навчання : навч. посіб. Київ : НАУ. 2021. 140 с.

43. Ушакова І. О. Підходи до створення інтелектуальних чат-ботів. *Системи обробки інформації*. 2019. Вип. 2 (157). С. 76-83.

44. Шмельова І. Інформаційна культура в умовах сучасного інформаційного простору. *Педагогіка і психологія*. 2020. № 5. С. 23-31.

45. Як розвивається ринок EdTech бізнесу в Україні. *Kyivstar Business Hub корпоративний блог для бізнесу*. URL: <https://hub.kyivstar.ua/articles/ed-tech-biznes-v-ukrayini-yak-rozvivayetsya-rinok-czifrovoyi-osviti>.

46. Яковлєва І. Використання освітніх платформ в освітньому середовищі. *Український Педагогічний журнал*. 2022. №3. С.137-148

47. Яремко С., Кузьміна О., Бевз С. Дослідження напрямків розвитку систем управління навчанням. *Економічні науки. Вісник Хмельницького національного університету*. 2022. №1. С. 252-255.

ДОДАТКИ

Додаток А

ТЕХНІЧНЕ ЗАВДАННЯ

1. Вступ

Технічне завдання описує практичну частину кваліфікаційної роботи – освітню вебплатформу SmartClass, призначену для управління навчальним контентом та комунікацією між учасниками освітнього процесу.

Створений вебзастосунок орієнтований на підтримку дистанційного, змішаного та очного навчання, забезпечення інтерактивної взаємодії між учнями та вчителями, систематизацію навчальних матеріалів, роботу з класами й ресурсами.

2. Підстави для розробки

Розробка була виконана відповідно до індивідуального завдання кваліфікаційної роботи та є логічним продовженням досліджень у сфері створення веборієнтованих освітніх середовищ. Необхідність такої розробки обумовлена потребою навчальних закладів у зручних, інтерактивних, адаптивних та доступних цифрових ресурсах, які здатні забезпечити ефективне управління навчальним процесом, комунікацію та контроль.

У межах педагогічної та переддипломної практик проводилася апробація системи у реальному освітньому середовищі – у Старопочаївській загальноосвітній школі I-III ступенів, що підтвердило актуальність платформи.

3. Призначення розробки

Функціональне призначення SmartClass полягає у створенні багатофункціонального вебзастосунку, який об'єднує інструменти для організації роботи класів, управління навчальними матеріалами, взаємодії між учасниками освітнього процесу та підтримки навчального контенту.

Платформа розроблена для використання вчителями, учнями та адміністрацією навчального закладу. Вона забезпечує:

- створення навчальних класів і керування ними;

- розміщення та структурування навчального контенту;
- обмін інформацією між учасниками;
- комунікацію у режимі реального часу;
- підтримку освітнього процесу у дистанційному й змішаному форматах.

Експлуатаційне призначення системи – створення єдиного цифрового навчального середовища, яке спрощує організацію занять, підвищує рівень взаємодії та сприяє ефективному засвоєнню навчального матеріалу.

4. Вимоги до програмного продукту

4.1. Вимоги до функціональних характеристик

Для SmartClass визначено наступний функціональний набір:

1. Реєстрація та автентифікація користувачів. Система повинна забезпечувати створення облікових записів із валідацією даних, вхід через логін і пароль, підтримку ролей (учитель, учень).
2. Створення та управління навчальними класами. Вчитель має можливість створювати клас, зазначати його назву, секцію та опис. Система повинна дозволяти редагування даних класу та його архівування.
3. Запрошення та приєднання учнів. Для кожного класу генерується унікальний код-запрошення або посилання. Учні мають можливість приєднатися, використовуючи цей код.
4. Управління членами класу. Учитель повинен мати право переглядати список учнів, надавати чи обмежувати доступ.
5. Розміщення навчальних матеріалів (Documents / Files). Система має дозволяти завантаження файлів, їх перегляд, редагування опису та видалення.
6. Інформаційні пости та оголошення. Реалізується можливість публікації вчителем або учасниками інформаційних повідомлень у межах класу.
7. Чат у реальному часі (SignalR). Учні та вчитель можуть обмінюватися повідомленнями в межах класу або окремих тем.
8. Персональний кабінет користувача. Профіль містить основні дані користувача (ім'я, email, роль), з можливістю їх редагування.
9. Сторінки інтерфейсу та навігація. Система повинна мати інтуїтивну

структуру сторінок, адаптивний дизайн та зручні переходи між модулями.

10. Модерація та контроль. Вчитель має можливість видаляти небажаний контент та керувати активністю у класі.

4.2. Вимоги до надійності

1. Система повинна коректно обробляти одночасні запити від багатьох користувачів.

2. Робота сервісу має бути стабільною, без збоїв при активній взаємодії.

3. Повинно бути проведене тестування функціональності, навантаження та безпеки.

4. Має здійснюватися логування критичних подій та помилок.

5. Авторизація та автентифікація забезпечуються токенами (JWT).

6. Забезпечена цілісність та захист персональних даних.

4.3. Умови експлуатації

Платформа повинна працювати у режимі 24/7, бути доступною з будь-якого пристрою, що підтримує сучасні вебстандарти. Дозволяється короткочасна зупинка роботи під час оновлень.

4.4. Вимоги до технічних засобів

Для клієнтської частини достатньо будь-якого пристрою з браузером Chrome, Firefox, Edge або Safari та стабільним інтернетом. Серверна частина розгортається на середовищі з підтримкою ASP.NET Core та SQL Server.

4.5. Вимоги до інформаційної сумісності

1. Підтримка роботи у всіх основних браузерах.

2. Адаптація під мобільні пристрої.

3. Повна робота в онлайн-режимі.

4.6. Вимоги до збереження даних

Всі дані сайту повинні зберігатися у надійному сервері з відповідними стандартами безпеки.

5. Вимоги до програмної документації

Документація повинна містити:

- опис архітектури системи;
- опис усіх основних сторінок платформи;
- пояснення алгоритмів роботи сервісів;
- інструкцію користувача для обох ролей – учня та вчителя.

6. Техніко-економічні показники

Розробка SmartClass не потребує значних фінансових ресурсів, оскільки використовуються безкоштовні фреймворки та інструменти. Очікувана економічна ефективність полягає у підвищенні якості освітнього процесу, зменшенні навантаження на вчителя, оптимізації часу та централізації навчальних матеріалів.

7. Стадії та етапи розробки

1. Визначення всіх головних та аналіз можливих функціональних та нефункціональних вимог до системи.

2. Проектування та розробка архітектурної частини проекту та дизайну інтерфейсу для користувача.

3. Розробка та імплементація функціональних модулів:

- розробка і налаштування бази даних для зберігання і отримання всієї необхідної інформації про користувача, завдання, класи;
- інтеграція системи авторизації;
- створення відповідних сервісів для взаємодії з БД та користувачами
- розробка і опис необхідних сторінок та елементів на сайті;

4. Тестування, відлагодження та оптимізація.

5. Впровадження та налаштування системи.

6. Подальша підтримка для коректної експлуатації системи.

7. Проведення регулярних заходів з оновлення і розширення, щоб врахувати нові вимоги користувачів або вдосконалювати наявний функціонал.

8. Проводити моніторинг продуктивності системи, здійснювати процеси оптимізацію для забезпечення її найкращої роботи.

8. Необхідні сторінки та структурні елементи сайту SmartClass (у вигляді таблиці)

Назва сторінки модуля	Призначення та детальний опис
MainLayout	Базовий шаблон сайту, який використовується для всіх сторінок. Містить шапку з логотипом платформи SmartClass, навігаційне меню, інформацію про авторизованого користувача, кнопку виходу з акаунта та основну область для відображення контенту.
Home	Головна стартова сторінка платформи. Містить загальний опис можливостей SmartClass, інформацію про переваги системи, кнопки переходу до реєстрації та входу.
Register	Сторінка реєстрації нового користувача. Містить поля для введення імені, email, пароля та підтвердження пароля і вибору ролі на ресурсі. Усі поля мають відповідну валідацію даних.
Login	Сторінка входу в систему. Користувач виконує авторизацію шляхом введення email та пароля.
Profile / MyAccount	Особистий кабінет користувача. Містить інформацію про користувача, роль у системі, можливість редагування персональних даних та зміни пароля.
My Classrooms / Dashboard	Головна робоча сторінка після авторизації. Відображає список усіх класів, до яких має доступ користувач.
Create Classroom	Сторінка створення нового класу. Вчитель вводить назву класу, секцію та опис, після чого підтверджує створення.
Join Classroom	Сторінка приєднання учня до класу за унікальним кодом або запрошувальним посиланням.
Classroom	Головна сторінка конкретного класу. Містить навчальні

View	матеріали, оголошення, список учасників та чат.
Members	Сторінка зі списком усіх учасників класу. Вчитель має право керувати учнями.
Materials / Files	Сторінка керування навчальними матеріалами. Містить можливість завантаження, перегляду, редагування опису та видалення файлів.
Chat	Модуль обміну повідомленнями в режимі реального часу на базі SignalR.
Classroom Settings	Сторінка налаштувань класу. Вчитель може змінювати назву, опис, доступи, архівувати або видаляти клас.
Admin Panel	Адміністративна панель для системного керування користувачами, класами та статистикою.
Help / FAQ	Довідковий розділ системи. Містить інструкції для користувачів та відповіді на типові запитання.
Privacy Policy	Сторінка політики конфіденційності. Містить інформацію про обробку персональних даних.
NotFound / Error	Службова сторінка для відображення повідомлень про помилки та неіснуючі сторінки.

9. Порядок контролю і прийняття програмного засобу

Контроль якості та прийняття освітньої вебплатформи SmartClass здійснюється поетапно, відповідно до вимог, визначених у технічному завданні та індивідуальному завданні на магістерську роботу. Перевірка коректності роботи програмного продукту проводиться на кожному етапі розробки з метою своєчасного виявлення помилок та їх усунення.

Поточний контроль включає перевірку працездатності модулів реєстрації та автентифікації користувачів, створення та управління класами, приєднання учнів за кодом доступу, функціонування чату, завантаження і перегляду навчальних матеріалів, роботи профілю користувача та коректності навігації між

сторінками. Також перевіряється стабільність роботи системи при одночасному доступі декількох користувачів.

Фінальне прийняття програмного засобу здійснюється після проведення комплексного тестування, яке включає функціональне тестування, тестування безпеки, перевірку роботи з базою даних, а також перевірку відповідності інтерфейсу вимогам зручності та доступності.

Програмний продукт вважається прийнятим до експлуатації у разі:

- відповідності всіх реалізованих функцій технічному завданню;
- відсутності критичних помилок у роботі;
- стабільної роботи серверної та клієнтської частин;
- позитивних результатів апробації у навчальному закладі під час переддипломної практики.

10. Додатки до технічного завдання

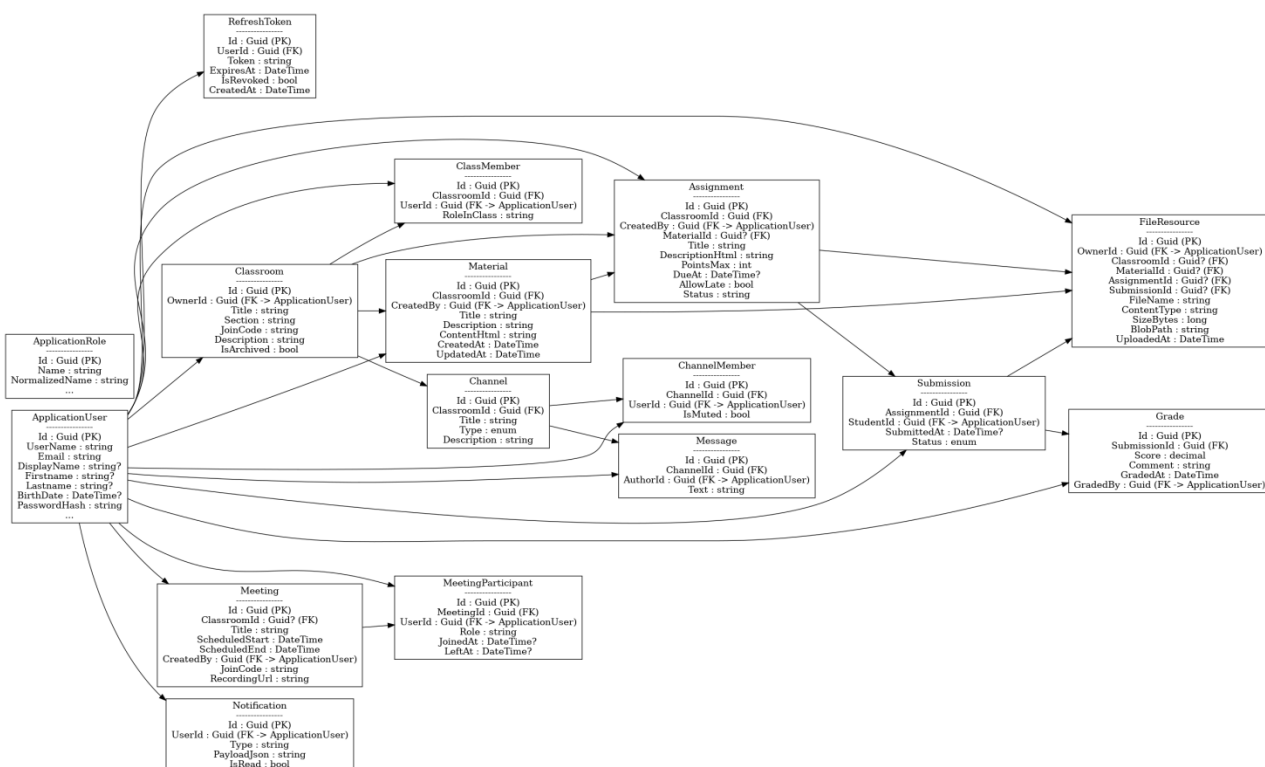


Рисунок А.1. Схема структури реляційної бази даних

СКРІНШОТИ ІНТЕРФЕЙСУ

SMART CLASS УВІЙТИ РЕЄСТРАЦІЯ

Створити акаунт

Заповніть форму для реєстрації

Ім'я: Іван Прізвище: Петренко

Електронна пошта: name@example.com

Пароль:

Дата народження: Оберіть дату Роль: Student

Зареєструватися

[Вже є акаунт?](#)

Рисунок Б.1. Сторінка реєстрації акаунту користувача

SMART CLASS УВІЙТИ РЕЄСТРАЦІЯ

Вхід

Увійдіть, щоб продовжити навчання

Електронна пошта: yura@gmail.com

Пароль: [Забули пароль?](#)

Увійти

[Ще не маєте акаунту?](#)

[Створити акаунт](#)

Рисунок Б.2. Сторінка входу у особистий кабінет

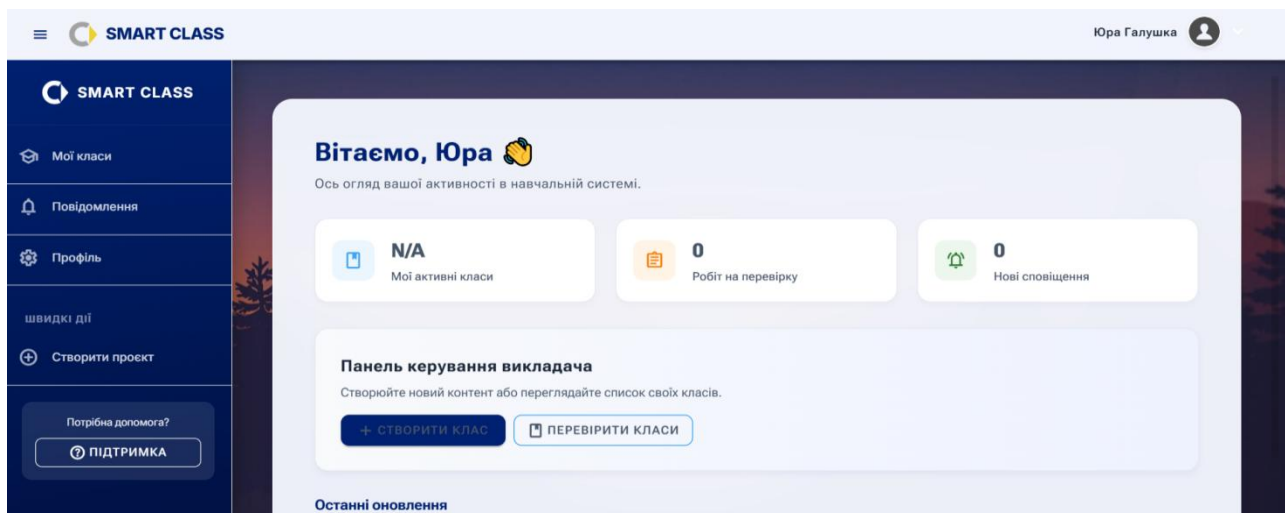


Рисунок Б.3. Головна сторінка сайту

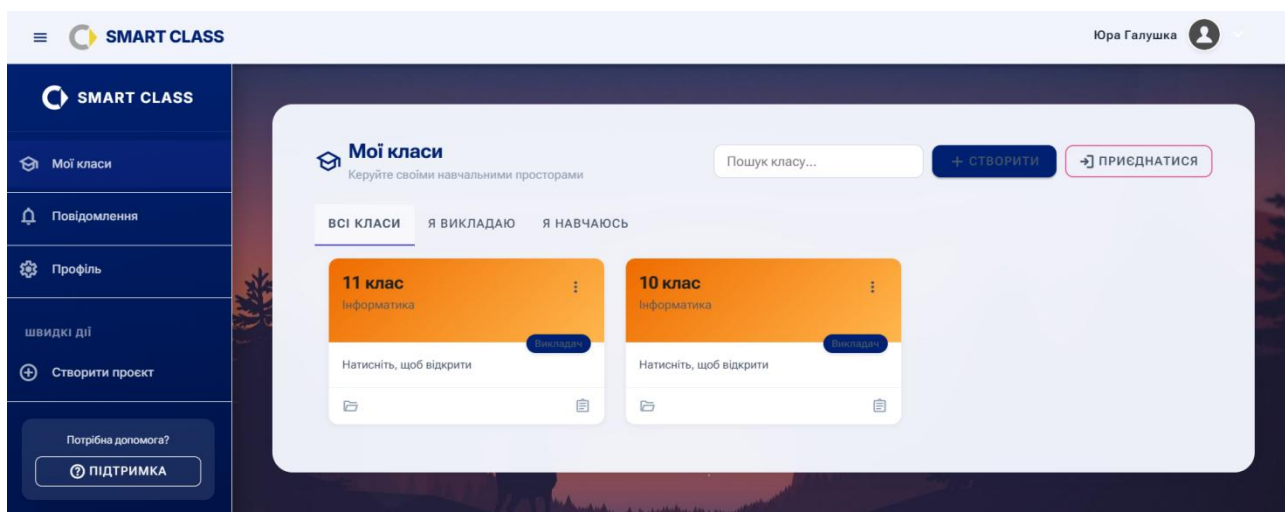


Рисунок Б.4. Сторінка зі списком навчальних класів користувача

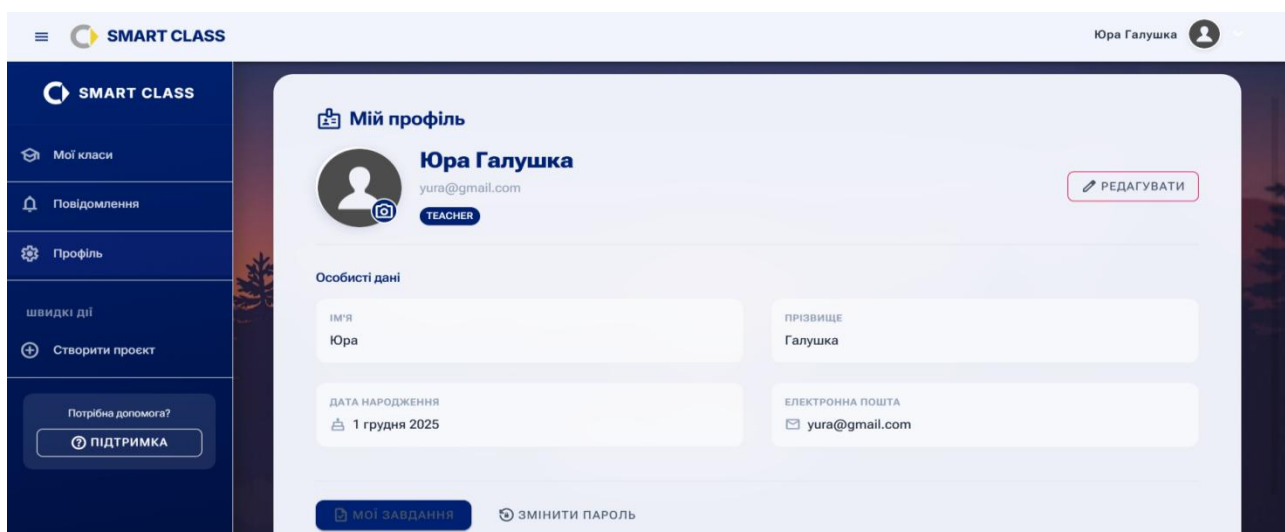


Рисунок Б.5. Особистий кабінет користувача

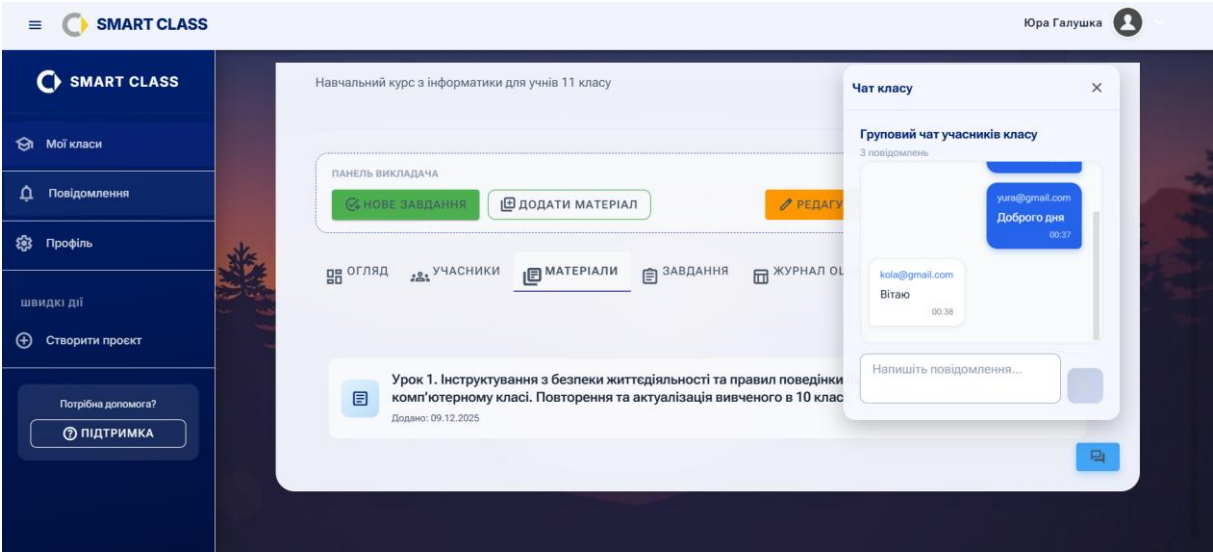


Рисунок Б.6. Вигляд вікна з чатом для учасників класу

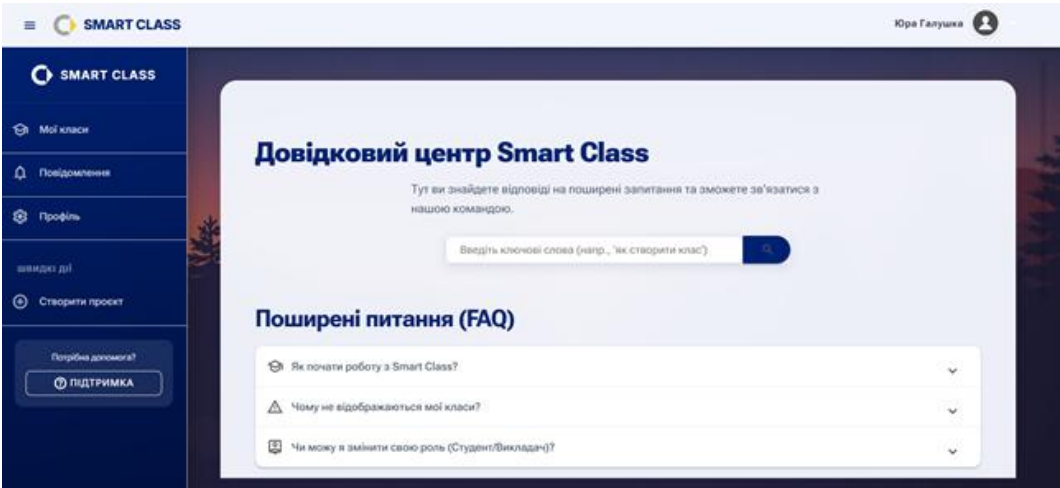


Рисунок Б.7. Сторінка підтримки з відповідями на питання

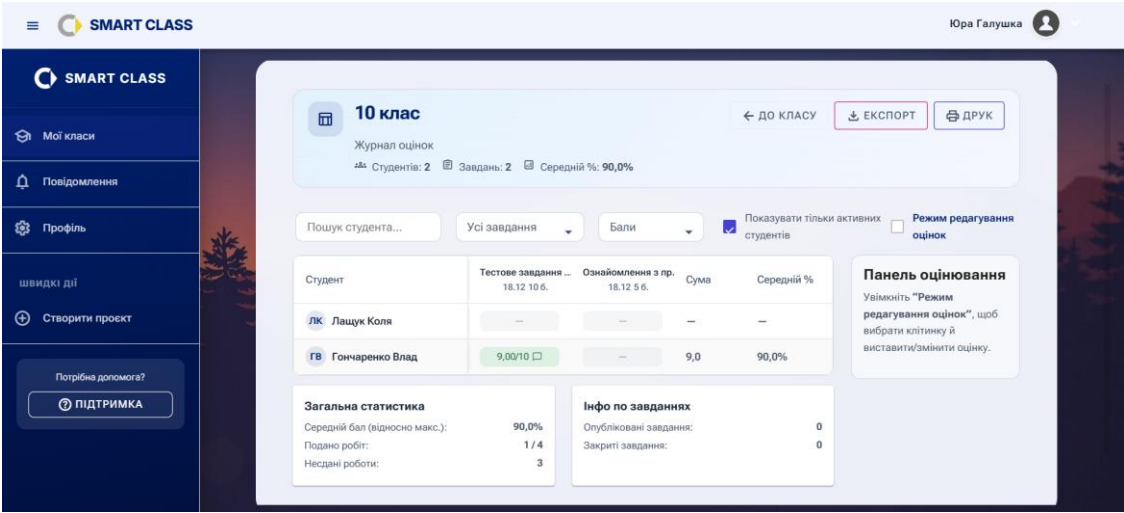


Рисунок Б.8. Приклад сторінки панелі оцінювання, для вчителя

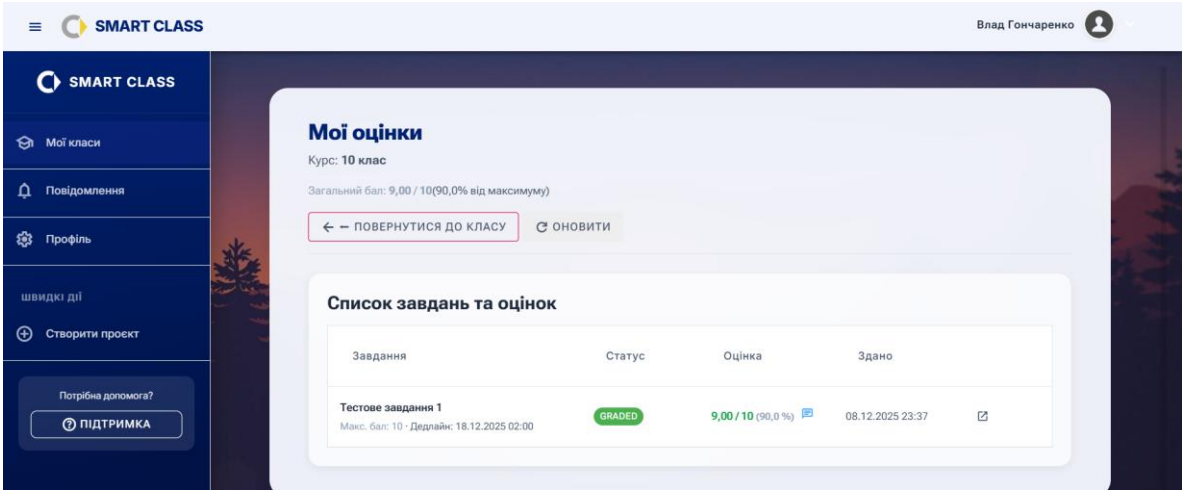


Рисунок Б.9. Приклад персональної сторінки з оцінками учня

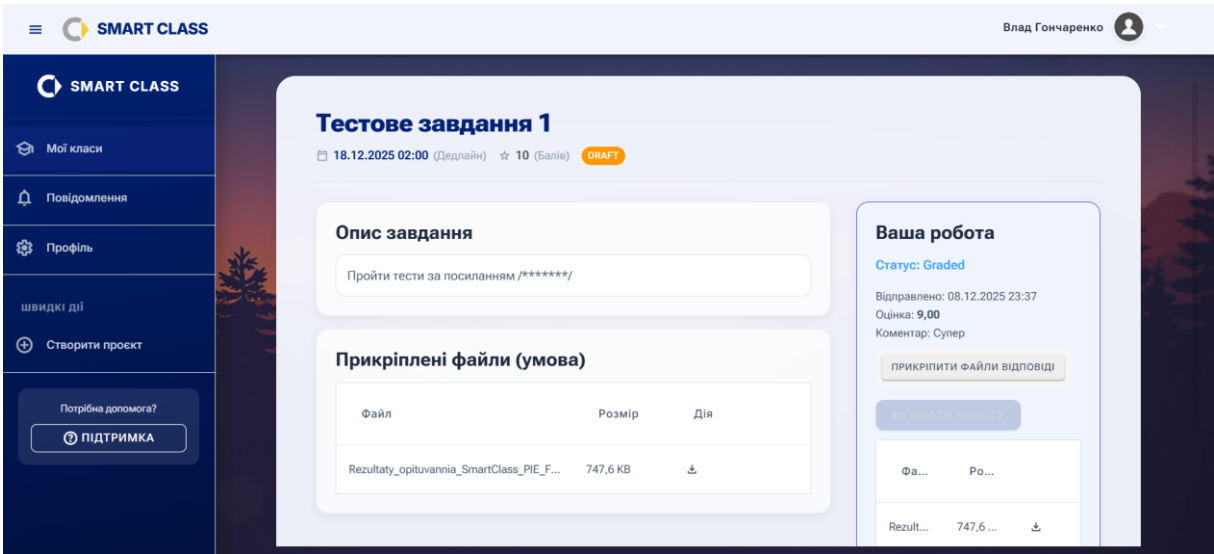


Рисунок Б.10. Сторінка з завданнями

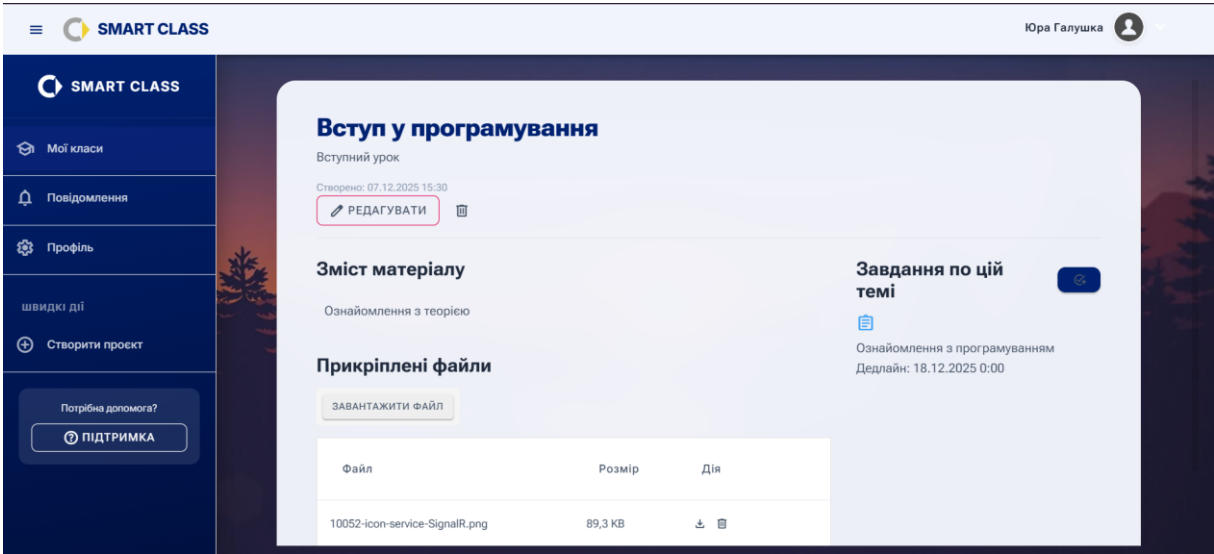


Рисунок Б.11. Сторінка з матеріалом

АНКЕТА ДЛЯ ОЦІНЮВАННЯ РОБОТИ ВЕБЗАСТОСУНКУ SMARTCLASS

Прошу Вас оцінити роботу вебзастосунку SmartClass за основними функціональними та інтерфейсними характеристиками. Оцінка продукції за чотирирівневою шкалою: «дуже добре», «добре», «середнє», «погано».

1. Оцініть зручність інтерфейсу вебзастосунку SmartClass
 - a. дуже добре
 - b. добре
 - c. посередньо
 - d. погано

2. Оцініть логічність структури меню та навігації
 - a. дуже добре
 - b. добре
 - c. посередньо
 - d. погано

3. Оцініть зручність роботи з навчальними матеріалами
 - a. дуже добре
 - b. добре
 - c. посередньо
 - d. погано

4. Оцініть зручність виконання та надсилання завдань
 - a. дуже добре
 - b. добре
 - c. посередньо
 - d. погано

5. Оцініть зрозумілість та прозорість системи оцінювання
- a. дуже добре
 - b. добре
 - c. посередньо
 - d. погано
6. Оцініть зручність використання чату для комунікації
- a. дуже добре
 - b. добре
 - c. посередньо
 - d. погано
7. Оцініть стабільність роботи вебзастосунку
- a. дуже добре
 - b. добре
 - c. посередньо
 - d. погано
8. Оцініть швидкість завантаження сторінок вебзастосунку
- a. дуже добре
 - b. добре
 - c. посередньо
 - d. погано
9. Оцініть коректність адаптації вебзастосунку до різних розширень екрану
(адаптивний дизайн)
- a. дуже добре
 - b. добре
 - c. посередньо

d. погано

10. Оцініть загальний рівень зручності використання вебзастосунку SmartClass

- a. дуже добре
- b. добре
- c. посередньо
- d. погано

11. Які функціональні можливості вебзастосунку Ви використовували найчастіше?

(можна обрати декілька варіантів)**

- a. перегляд навчальних матеріалів
- b. виконання та надсилання завдань
- c. перегляд оцінок
- d. робота з класами
- e. використання чату
- f. інше: _____

12. Які недоліки у роботі вебзастосунку Ви помітили?

13. Які функції, на Вашу думку, доцільно додати або вдосконалити?

РЕЗУЛЬТАТИ ОПИТУВАННЯ СТОСОВНО ОЦІНКИ РОБОТИ ВЕБЗАСТОСУНКУ SMARTCLASS

Під час тестування вебзастосунку SmartClass було проведено опитування користувачів для оцінки функціональності, зручності використання інтерфейсу, стабільності та адаптивності до різних розмірів екранів. В опитуванні взяли участь 18 респондентів. Частка «середніх» відповідей мінімальна, негативних оцінок немає.

Зручність інтерфейсу (n=18)

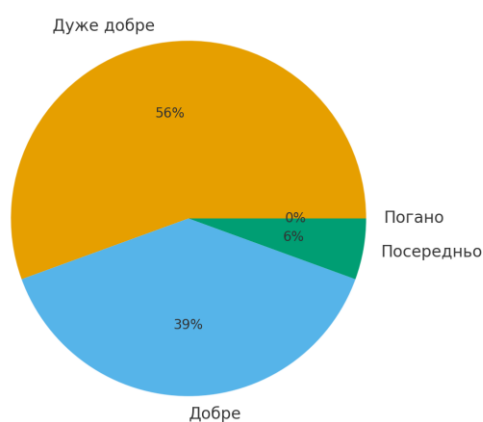


Рисунок Г.1. Результати анкетування (питання 1)

Інтерфейс отримав дуже високі оцінки, що підтверджує його інтуїтивно зрозумілий вигляд та простоту використання.

Логічність навігації (n=18)

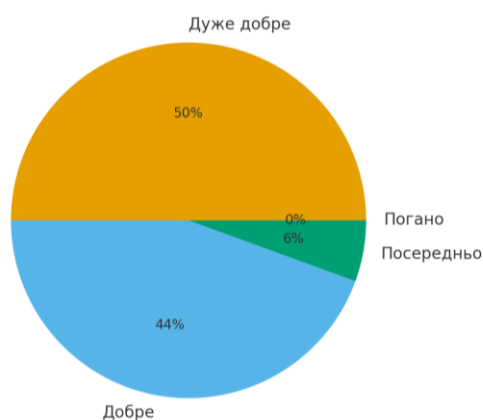


Рисунок Г.2. Результати анкетування (питання 2)

Структура меню та навігація логічні та зрозумілі для більшості користувачів.

Зручність роботи з матеріалами (n=18)

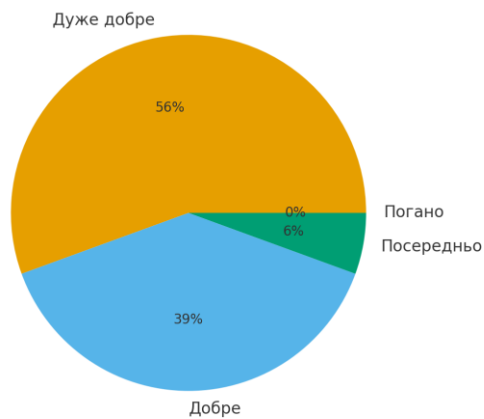


Рисунок Г.3. Результати анкетування (питання 3)

Користувачі відзначили високу зручність перегляду та використання навчальних матеріалів.

Зручність виконання завдань (n=18)

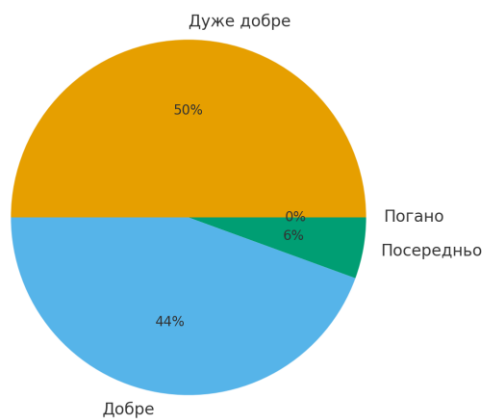


Рисунок Г.4. Результати анкетування (питання 4)

Процес виконання та подання завдань простий та зрозумілий.

Зрозумілість оцінювання (n=18)

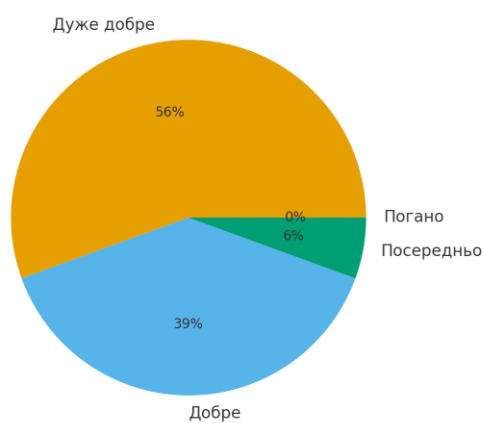


Рисунок Г.5. Результати анкетування (питання 5)

Система оцінювання прозора та зрозуміла для користувачів.

Зручність чату (n=18)

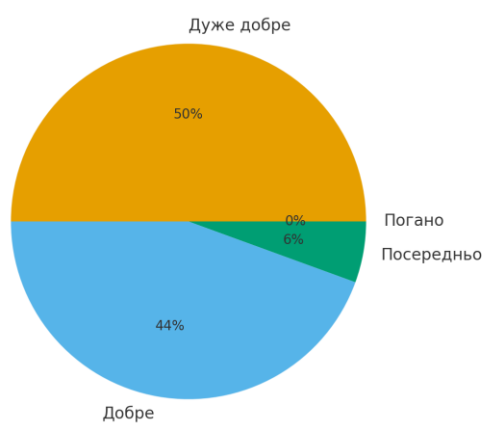


Рисунок Г.6. Результати анкетування (питання 6)

Функція чату забезпечує швидкий та зручний зворотний зв'язок.

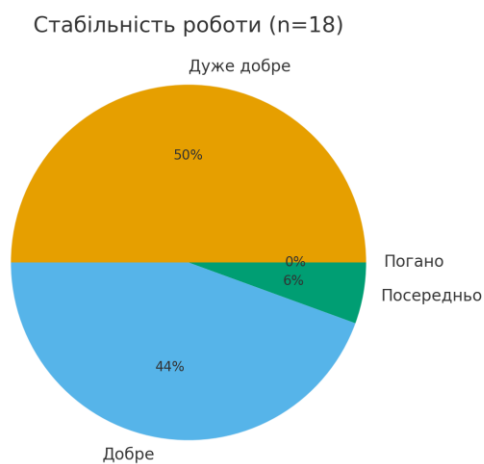


Рисунок Г.7. Результати анкетування (питання 7)

Вебдодаток працює стабільно без суттєвих технічних збоїв.

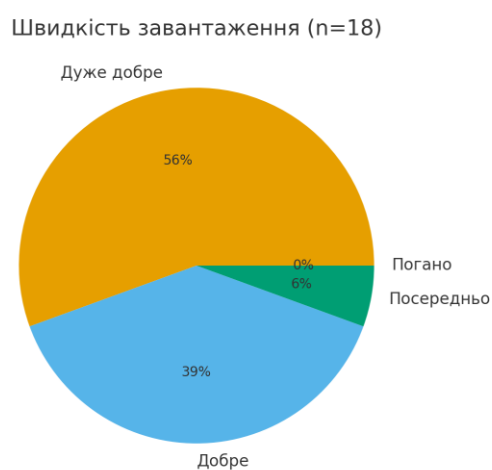


Рисунок Г.8. Результати анкетування (питання 8)

Швидкості системи достатньо для комфортної роботи.

Адаптація до різних розширень екрану (n=18)

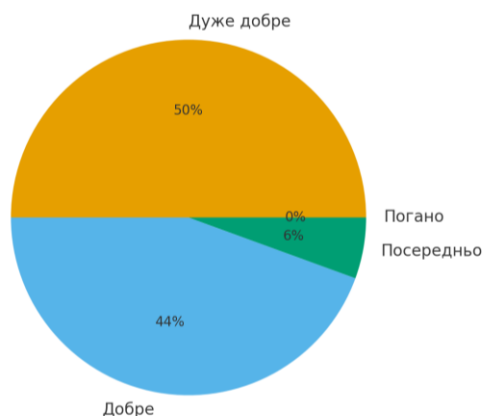


Рисунок Г.9. Результати анкетування (питання 9)

Адаптивний дизайн гарантує коректне відображення вебзастосунку на різних пристроях.

Загальна зручність (n=18)

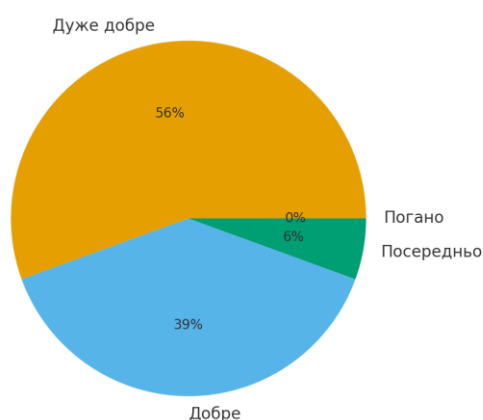


Рисунок Г.10. Результати анкетування (питання 10)

Загальна оцінка зручності використання SmartClass дуже висока.

Загальні висновки за результатами апробації

Узагальнені результати тестування підтверджують дуже високий рівень зручності, стабільності та функціональної придатності вебзастосунку SmartClass. Переважна більшість користувачів оцінили всі ключові функції на рівні «дуже добре» та «добре», що свідчить про якісну реалізацію інтерфейсу, логіку роботи системи та її адаптивність.