

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВОЛИНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ЛЕСІ УКРАЇНКИ
Кафедра комп'ютерних наук та кібербезпеки

На правах рукопису

САМОХ ПАВЛО ІГОРОВИЧ
**ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ
СУПРОВОДУ ОСВІТНЬОГО ПРОЦЕСУ ЗАКЛАДУ ЗАГАЛЬНОЇ
СЕРЕДНЬОЇ ОСВІТИ**

Спеціальність: 122 Комп'ютерні науки
Освітньо-професійна програма: Комп'ютерні науки та інформаційні технології
Робота на здобуття освітнього ступеня «магістр»

Науковий керівник:
ГЛИНЧУК ЛЮДМИЛА ЯРОСЛАВІВНА
кандидат фізико-математичних наук, доцент
кафедри комп'ютерних наук та кібербезпеки

РЕКОМЕНДОВАНО ДО ЗАХИСТУ
Протокол № _____
засідання кафедри комп'ютерних наук
та кібербезпеки
від _____ 2025 р.
Завідувач кафедри
(_____) Гришанович Т. О.

ЛУЦЬК – 2025

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ СУПРОВОДУ ОСВІТНЬОГО ПРОЦЕСУ ЗЗСО	6
1.1 Сучасні тенденції цифрової трансформації закладів загальної середньої освіти	6
1.2 Інформаційні системи для управління освітнім процесом.....	11
1.3 Вимоги до інформаційної системи супроводу освітнього процесу ЗЗСО .	17
1.4 Огляд та аналіз існуючих програмних рішень.....	20
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ СУПРОВОДУ ОСВІТНЬОГО ПРОЦЕСУ ЗЗСО	29
2.1 Постановка задачі, призначення та вимоги до розробки.....	29
2.2 Вибір моделі розробки програмного засобу	30
2.3 Опис проєкту програмного засобу	31
2.4 Обґрунтування вибору інструментальних засобів розробки.....	33
2.5 Програмна реалізація та її особливості	35
2.6 Тестування та налагодження програмного засобу	47
2.7 Рекомендації по впровадженню та використанню програмного засобу	47
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТКИ.....	55

ВСТУП

Актуальність теми. Розвиток загальної середньої освіти в Україні відбувається в умовах активної цифрової трансформації, що включає впровадження електронних журналів і щоденників, систем управління навчанням, хмарних сервісів для обміну матеріалами та інструментів онлайн-взаємодії між учасниками освітнього процесу. Проте цифрова інфраструктура більшості закладів освіти залишається фрагментованою: школи використовують окремі сервіси для розкладу, завдань, оцінювання, комунікації чи аналітики, які не інтегровані між собою. Це призводить до дублювання даних, ускладнює їх узгодження, створює ризики помилок і унеможливорює комплексний моніторинг навчального процесу.

Проблема стала особливо гострою внаслідок повномасштабної збройної агресії росії. Цілеспрямовані удари по енергетичній інфраструктурі спричиняють тривалі відключення електроенергії, перебої інтернет-зв'язку, що ускладнює доступ до навчальних даних. Постійна загроза ракетних ударів та повітряні тривоги призводять до переривання занять, зміни режимів роботи шкіл та тимчасового переміщення учнів і вчителів. За таких умов відсутність інтегрованої інформаційної системи значно знижує стабільність освітнього процесу.

Отже, виникає потреба у створенні єдиного цифрового рішення, яке забезпечить узгоджене управління освітніми процесами, централізоване зберігання й обробку даних, ефективну комунікацію та підтримку безперервності навчання навіть в умовах зовнішніх загроз і технічних обмежень.

Мета роботи полягає у дослідженні та створенні інформаційної системи супроводу освітнього процесу закладу загальної середньої освіти, здатної забезпечити цілісне, узгоджене й надійне управління ключовими навчально-організаційними процесами.

Завдання дослідження:

- проаналізувати сучасні тенденції цифрової трансформації закладів загальної середньої освіти;
- визначити функціональні та нефункціональні вимоги до інформаційної системи супроводу освітнього процесу ЗЗСО;
- дослідити та порівняти існуючі інформаційні системи управління освітнім процесом;
- сформулювати концептуальну модель, архітектуру та основні підсистеми розроблюваної інформаційної системи;
- обґрунтувати вибір моделі та інструментальних засобів розробки програмного забезпечення;
- реалізувати інформаційну систему відповідно до визначених вимог;
- провести тестування програмного забезпечення та оцінити якість його роботи;
- надати рекомендації щодо впровадження та подальшого розвитку системи в ЗЗСО.

Об’єкт дослідження – інформаційні системи та цифрові інфраструктури, що підтримують організацію та супровід освітнього процесу в закладах загальної середньої освіти.

Предмет дослідження – методи, моделі, архітектурні рішення та технології проєктування і реалізації єдиної інформаційної системи супроводу освітнього процесу ЗЗСО.

Практичне значення полягає у створенні програмного рішення, яке може бути впроваджене в школах для автоматизації навчально-організаційних процесів і забезпечення зменшення кількості ручних операцій, підвищення точності обліку та аналітики, централізований доступ до навчальних даних і покращення керованості освітнім процесом.

Публікації:

1. Самох П. І. Порівняльний аналіз інформаційних систем для супроводу освітнього процесу в закладах загальної середньої освіти. Проблеми

комп'ютерних наук, програмного моделювання та безпеки цифрових систем – 2025: матеріали II міжнар. наук.-практ. конференції. (Світязь–Луцьк, Україна, 9–11 червня 2025 р.). Луцьк, 2025 р. С. 218–219.

2. Глинчук Л.Я., Самох П.І. Особливості розробки інформаційної системи супроводу освітнього процесу закладу загальної середньої освіти. Інформаційні технології і автоматизація – 2025: матеріали XVIII міжн. наук.-практ. конференції. (Одеса, 30 - 31 жовтня 2025 р.). Одеса, Видавництво ОНТУ, 2025 р. С. 521-523.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ СУПРОВОДУ ОСВІТНЬОГО ПРОЦЕСУ ЗЗСО

1.1 Сучасні тенденції цифрової трансформації закладів загальної середньої освіти

Розвиток загальної середньої освіти в Україні відбувається в умовах, коли цифрові технології перестають бути додатковим інструментом і перетворюються на базовий компонент функціонування шкільної системи. Сучасні умови та реалії життя українського суспільства позначені стрімкою динамікою розвитку технологій і значними масштабами їх впровадження в різні галузі діяльності, зокрема у сферу освіти. Теперішній етап функціонування й розвитку української системи освіти характеризується інтенсивною імплементацією різноманітних цифрових технологій з метою оптимальної реалізації її цілей і завдань, забезпечення доступності і неперервності, підвищення якості й ефективності [9]. У цьому контексті цифрова трансформація набуває статусу стратегічного напрямку освітньої політики, а не сукупності несистемних нововведень.

Дослідження науково-методичного забезпечення цифровізації освіти підкреслюють, що цифрові зміни торкаються не лише інструментарію, а й самої інфосфери, у якій існує школа. «Мережа Інтернету утворила нове соціокультурне довкілля, яке все більшою мірою і динамікою впливає на людей. Розвиток такої інфосфери не може відбуватися поза освітньою діяльністю, відокремлено від сфери освіти» [5]. Звідси випливає, що заклад загальної середньої освіти має не тільки адаптуватися до зовнішнього цифрового середовища, а й цілеспрямовано проєктувати власне інформаційно-цифрове середовище, яке інтегрується в ширший суспільний контекст і одночасно забезпечує безпеку, якість та інклюзивність навчання.

Ключовим поняттям сучасних досліджень стає інформаційно-цифрове навчальне середовище закладу загальної середньої освіти. На основі аналізу різних підходів воно розглядається як цілісна система, що поєднує інформаційні

ресурси, технічну інфраструктуру, методичне забезпечення, організаційні процедури й управлінські механізми. Вітчизняні дослідники наголошують на тому, що інформаційно-цифрове навчальне середовище закладу загальної середньої освіти слід розглядати як системно організовану сукупність інформаційного, технічного, навчально-методичного забезпечення, спрямованих на організацію взаємодії учнів, вчителів, керівників шкіл та громадськості, а також на здійснення навчально-виховних впливів, що підтримуються цифровими засобами збору та передачі даних, апаратно-програмним та навчально-методичним забезпеченням [6]. Таке розуміння переносить акцент з окремих інструментів на архітектуру середовища в цілому, де всі компоненти мають бути взаємопов'язаними й узгодженими з освітніми цілями.

У дослідженнях принципів побудови сучасного цифрового освітнього середовища закладу загальної середньої освіти пропонується розглядати його як складну систему, що функціонує за низкою базових принципів: цілісності, системності, модульності, інтегрованості, відкритості, технологічної насиченості, орієнтації на потреби користувачів. Принцип цілісності вимагає, щоб усі цифрові сервіси школи були підпорядковані спільним завданням підтримки освітнього процесу, а не розвивалися як роз'єднані елементи. Принцип системності передбачає організацію цифрового середовища як взаємопов'язаної сукупності ресурсів, інструментів навчання, засобів управління, де зміни в одному компоненті неминуче породжують трансформації в інших. Принцип модульності орієнтує на структурування середовища за функціональними блоками, відповідальними за інформаційне забезпечення, організацію навчання, контроль результатів тощо, що полегшує адаптацію системи до змін потреб і умов.

Розгортання цифрового освітнього середовища має багатовимірний характер. На інфраструктурному рівні йдеться про забезпечення надійного підключення до Інтернету, доступу до сучасних пристроїв, використання хмарних сервісів, засобів збереження та резервування даних. На педагогічному рівні відбувається інтеграція цифрових ресурсів у навчальні програми, розвиток

змішаних та дистанційних моделей навчання, використання інтерактивних засобів і адаптивних платформ. На управлінському рівні цифрова трансформація проявляється через впровадження електронного документообігу, автоматизацію розкладу, обліку відвідування й оцінювання, побудову аналітичних панелей. На комунікаційному рівні зростає роль електронних журналів і щоденників, порталів школи, мобільних застосунків для учнів і батьків, інтегрованих засобів оповіщення. Окремий вимір становлять безпека та стійкість, які охоплюють захист персональних даних, кібербезпеку, відмовостійкість систем та підтримку безперервності навчання в кризових умовах.

Структурні зв'язки між цими вимірами формують модель цифрової зрілості закладу освіти, у якій інфраструктурні інвестиції мають бути узгоджені з педагогічними інноваціями, управлінськими практиками, культурою використання даних і процедурами безпеки. Узагальнене уявлення про взаємодію цих компонентів доцільно подано у вигляді схеми, що демонструє цифрове освітнє середовище як ядро, пов'язане з інфраструктурою, управлінням, навчанням, аналітикою і безпекою (рис. 1.1).



Рисунок 1.1 – Структурні виміри цифрової трансформації закладу загальної середньої освіти

Особливістю українського контексту є поєднання стратегічного курсу на цифровізацію з глибокою нерівністю умов і ресурсів між різними закладами та громадами. Аналіз даних міжнародного дослідження PISA та національної статистики показує суттєві відмінності між міськими й сільськими школами, закладами з різним соціально-економічним статусом, а також значний дефіцит навчальних і цифрових ресурсів. «Опитування керівників шкіл у межах дослідження PISA 2022 виявило відсутність цифрових ресурсів (комп'ютерів, доступу до Інтернету, системи управління навчанням або шкільних навчальних платформ), а також їх неналежну або низьку якість, що заважає їхнім закладам реалізовувати свій потенціал у здійсненні освітнього процесу. В Україні вагомою є проблема браку цифрових ресурсів та їх низької якості у закладах загальної середньої освіти» [3]. Така ситуація загострює розрив між політичними деклараціями та реальними можливостями шкіл, особливо в умовах воєнного стану.

На мікрорівні закладів освіти ці дисбаланси проявляються в неоднорідності розвитку цифрового освітнього середовища. Частина шкіл інтегрує електронні журнали й щоденники, використовує платформи дистанційного навчання, хмарні сховища, засоби відеоконференцій, має налаштовану внутрішню мережеву інфраструктуру. Інша частина функціонує в умовах нестабільного інтернет-з'єднання, обмеженої кількості пристроїв, відсутності технічної підтримки. У цьому контексті цифрове освітнє середовище може допомогти подолати географічні бар'єри, надавши доступ до якісної освіти учням із віддалених населених пунктів [20]. Однак реалізація цього потенціалу потребує не лише технічних рішень, а й цілеспрямованих управлінських стратегій, підготовки педагогів та участі місцевих громад.

Повномасштабне впровадження додатково змінює виміри цифрової трансформації, перетворюючи її з інструменту модернізації на засіб забезпечення життєздатності освітньої системи. Цифрове освітнє середовище має підтримувати швидкий перехід між очним, дистанційним і змішаним навчанням, зберігати цілісність освітніх даних у хмарних сховищах,

забезпечувати асинхронний доступ до матеріалів у разі відключень електроенергії та повітряних тривог, дозволяти гнучко змінювати розклад і форми взаємодії. У дослідженнях цифровізації освіти у воєнний час підкреслюється, що стійкість освітнього процесу безпосередньо залежить від того, наскільки школа здатна спиратися на інтегровану інформаційну систему, а не на набір розрізнених сервісів.

Цифрова трансформація неможлива без переосмислення ролі педагога та управлінської команди. У працях, присвячених опису цифрової компетентності педагогічного працівника, розроблено структурну модель, яка включає компоненти діяльності в цифровому суспільстві, професійний розвиток, використання цифрових ресурсів, організацію навчання й оцінювання, формування цифрових компетентностей учнів. Уточнюються рівні володіння цифровими технологіями, пов'язані не лише з технічними навичками, а й з педагогічним дизайном, критичною оцінкою інформації, дотриманням етичних і правових норм. Дослідження цифрової компетентності педагогів свідчать, що без системних програм підвищення кваліфікації та підтримки професійних спільнот нерівномірність цифрової готовності вчителів залишається одним із головних бар'єрів цифрової трансформації шкільної освіти.

У підсумку цифрова трансформація ЗЗСО в сучасних умовах розглядається як багаторівневий процес, що поєднує зміну інфраструктури, педагогічних практик, моделей управління, способів комунікації та використання даних. У поєднанні з викликами війни цей процес стає критичним для забезпечення безперервності й якості освітнього процесу, а також для реалізації концепції безпечної, гнучкої та стійкої школи. На цій основі виникає потреба в інформаційній системі, здатній підтримувати всі зазначені виміри цифрового освітнього середовища, що стає предметом подальшого аналізу в наступних підрозділах.

1.2 Інформаційні системи для управління освітнім процесом

Інформаційні системи управління освітнім процесом посідають центральне місце в цифровій трансформації закладів загальної середньої освіти, оскільки забезпечують підтримку основних адміністративних, навчальних, комунікаційних та аналітичних процесів школи. У загальному організаційному контексті вони спираються на класичне розуміння інформаційної системи як сукупності взаємопов'язаних компонентів, що збирають, опрацьовують, зберігають і розподіляють інформацію для підтримки ухвалення рішень та контролю в організації. На рівні ЗЗСО такі системи мають не лише реєструвати дані, а й забезпечувати їхню цілісність, доступність для різних категорій користувачів, інтеграцію з надшкільними інформаційними ресурсами та можливість використання даних для освітньої аналітики.

У сучасних дослідженнях виокремлюють кілька основних класів інформаційних систем, що безпосередньо пов'язані з управлінням освітнім процесом у школі.

Перший клас становлять системи управління школою (SMIS), які фокусуються на адміністративних задачах: ведення електронних карток учнів і працівників, облік відвідування, оцінювання, формування розкладу, облік руху контингенту, підтримка кадрових і ресурсних рішень, підготовка звітності. Ці системи розглядаються як інструмент переходу від паперових і розрізнених електронних форм обліку до єдиного джерела достовірних даних для керівництва закладу і засновника. У контексті зростання обсягів освітніх даних підкреслюється, що школи повинні базувати освітнє ухвалення рішень на точних, своєчасних та якісних даних, поєднаних із відповідно підготовленим персоналом [34].

Другий ключовий клас формують системи управління навчанням (LMS). Вони орієнтовані на організацію та супровід навчального процесу: підтримку курсів, тестування, зворотний зв'язок, фіксацію прогресу учнів, роботу з навчальними матеріалами. У дослідженнях функціональності таких рішень для

шкіл показано, що критерії оцінювання їх можливостей доцільно групувати у три макрокатегорії: управління навчанням, управління навчальним контентом та інструменти комунікації й колаборації, при цьому запропоновані критерії можуть використовуватися для оцінки функціональності будь-якої платформи електронного навчання і дають змогу вчителям та адміністраторам робити усвідомлений вибір високоякісного рішення, а розробникам – удосконалювати функціональність платформ [33]. Такі системи поступово стають ядром організації навчальної діяльності у цифровому середовищі школи.

Третій клас це платформи дистанційного навчання, які забезпечують комплексну підтримку змішаного й дистанційного навчання та роблять акцент на організації онлайн-взаємодії: проведенні синхронних занять (відеоконференцій), комунікації (чати, форуми, сповіщення), спільній роботі з матеріалами й документами та координації навчальної діяльності. Такі платформи можуть застосовуватися як самостійно, так і в інтеграції з LMS, доповнюючи їхні можливості інструментами комунікації та колаборації.

Четвертий клас охоплює інтегровані освітні середовища, які поєднують адміністративні й навчальні функції. У національній практиці це можуть бути автоматизовані системи управління закладами освіти та територіальні платформи, що інтегрують школи громади через єдині реєстри учнів, педагогів, навчальних планів, а також модулі аналітики й формування звітності. Науково-методичні матеріали з впровадження інформаційно-комунікаційної автоматизованої системи для шкіл підкреслюють, що електронні журнали й щоденники мають розглядатися не як ізольовані продукти, а як компоненти комплексної системи, яка забезпечує цілісність та оперативність даних для управлінських рішень і одночасно створює середовище прозорості взаємодії між школою, учнями та батьками.

Окремий напрям пов'язаний із національними та регіональними інформаційними системами управління освітою, які забезпечують збір, валідацію й аналіз даних з усіх закладів освіти для потреб державної політики, фінансування, прогнозування мережі, оцінювання результатів реформ. В

аналітичних матеріалах Інституту освітньої аналітики деталізується роль централізованих інформаційно-аналітичних систем, які акумулюють дані про контингент здобувачів освіти, кадровий склад, результати зовнішнього оцінювання, інфраструктуру, фінансові показники, дозволяючи переходити до моделі державного управління, ґрунтованого на доказах. Для шкіл це означає потребу у стандартизованих форматах даних, інтеграції локальних систем з надшкільними платформами, узгодженні структур довідників і ідентифікаторів.

Шосту групу становлять спеціалізовані модулі й сервіси: електронні журнали, щоденники, системи реєстрації вступу й переведення, кабінети батьків, модулі побудови індивідуальних освітніх траєкторій, засоби електронного документообігу. Їхнє поширення особливо помітне в українських ЗЗСО, де впровадження електронного класного журналу часто стає першим кроком до цифровізації управління. У методичних рекомендаціях наголошується, що такі інструменти мають слугувати засобом підтримки управлінської діяльності закладу освіти та окремого педагога, полегшувати доступ до інформації, підвищувати прозорість і зменшувати дублювання паперової звітності. Практика показує, що впровадження лише окремих модулів без інтегрованої системи нерідко призводить до паралельного ведення паперових і електронних форм, надмірного навантаження на вчителів і суперечностей у даних.

Функціональні можливості інформаційних систем управління освітнім процесом доцільно описувати через типові групи задач, які вони розв'язують. Для адміністративного рівня це реєстрація й ведення облікових записів учнів і працівників, управління навчальним навантаженням і розкладом, облік відвідування, оцінювання, формування внутрішніх і зовнішніх звітів, управління ресурсами. Для навчального рівня – управління курсами й навчальними планами, публікація та структуризація навчальних матеріалів, організація виконання завдань і проєктів, формування електронних портфоліо. Для комунікаційного рівня – повідомлення, сповіщення, форуми, відеоконференції, електронні щоденники, особисті кабінети батьків і учнів. Для аналітичного рівня – побудова

звітів, дашбордів, індикаторів досягнень, виявлення ризиків, підтримка внутрішньої системи забезпечення якості освіти.

У дослідженнях, присвячених практиці використання систем управління навчанням, підкреслюється, що такі системи фактично поєднують функції навчальної та управлінської платформи. Зокрема наголошується, що в межах однієї системи вчителі можуть керувати своїми учнями, адмініструвати тести, забезпечувати обговорення в класі та надавати навчальні матеріали за допомогою цього інструмента [11]. Тим самим інформаційна система стає місцем, де сходяться управління доступом до курсів, оцінювання, комунікація, аналіз активності учнів.

Сучасні тенденції розвитку шкільних інформаційних систем тісно пов'язуються з переходом до веборієнтованих і хмарних архітектур. У роботах, присвячених проектуванню шкільних інформаційних систем, наголошується, що врахування потреб учнів, батьків і вчителів робить такі інформаційні системи важливим інструментом підтримки прозорості, ефективності та точності інформації, яку надають школи, а веборієнтовані інформаційні системи стають суттєвим засобом забезпечення цих характеристик [25]. Для українських ЗЗСО це особливо актуально, оскільки доступ з різних пристроїв дозволяє підтримувати безперервність взаємодії навіть за умов переміщення учасників освітнього процесу та відключень електроенергії.

У зарубіжних і вітчизняних публікаціях дедалі більше уваги приділяється хмарним рішенням. Аналіз досліджень упровадження хмарних платформ показує, що значна кількість робіт фіксує зміни в організації та адмініструванні систем управління навчанням під впливом хмарних технологій, при цьому численні суттєві переваги хмарних систем управління навчанням сприяють поліпшенню освітнього досвіду та організаційної ефективності [15]. Для шкіл це означає можливість зменшити витрати на власну інфраструктуру, отримати масштабованість і гнучкість, але водночас це висуває підвищені вимоги до захисту даних.

З огляду на вказані функції та архітектурні особливості, класи інформаційних систем, що використовуються у ЗЗСО, можна узагальнити в таблиці 1.1.

Таблиця 1.1

Основні класи інформаційних систем для управління освітнім процесом у
ЗЗСО

Клас системи	Основне призначення	Типові модулі та сервіси
Системи управління школою (SMIS)	Адміністративне управління закладом, підтримка управлінських рішень	Реєстр учнів і персоналу, облік відвідування й оцінок, розклад, звітність, кадровий облік
Системи управління навчанням (LMS)	Організація й супровід навчального процесу	Курси й модулі, тести, трекінг прогресу, каталоги матеріалів, аналітика навчання
Платформи дистанційного навчання для шкіл	Комплексна підтримка змішаного й дистанційного навчання	Контент-менеджмент, відеоконференції, форуми, чати, спільні документи
Спеціалізовані модулі	Розв'язання окремих задач освітнього процесу й комунікації	Електронний журнал і щоденник, реєстрація вступу, кабінет батьків і учнів
Інтегровані освітні середовища	Поєднання управлінських, навчальних і комунікаційних функцій в єдиній системі	Єдині профілі користувачів, інтеграція SMIS і LMS, єдині довідники, наскрізна аналітика
Національні/регіональні інформаційні системи освіти	Збір і аналіз даних на рівні громади, регіону, держави	Централізовані бази даних, модулі звітності, інтеграція з реєстрами, аналітичні панелі

Логіка взаємодії між цими класами систем, ролями користувачів та потоками даних формує цілісну архітектуру керування освітнім процесом. На рівні закладу інтегрована інформаційна система забезпечує єдину точку доступу для адміністрації, педагогів, учнів і батьків, тоді як на рівні громади й держави відповідні інформаційно-аналітичні системи агрегують та узагальнюють дані для

стратегічних рішень. Схематично узагальнену структуру такої інформаційної системи управління освітнім процесом подано на рисунку 1.2.

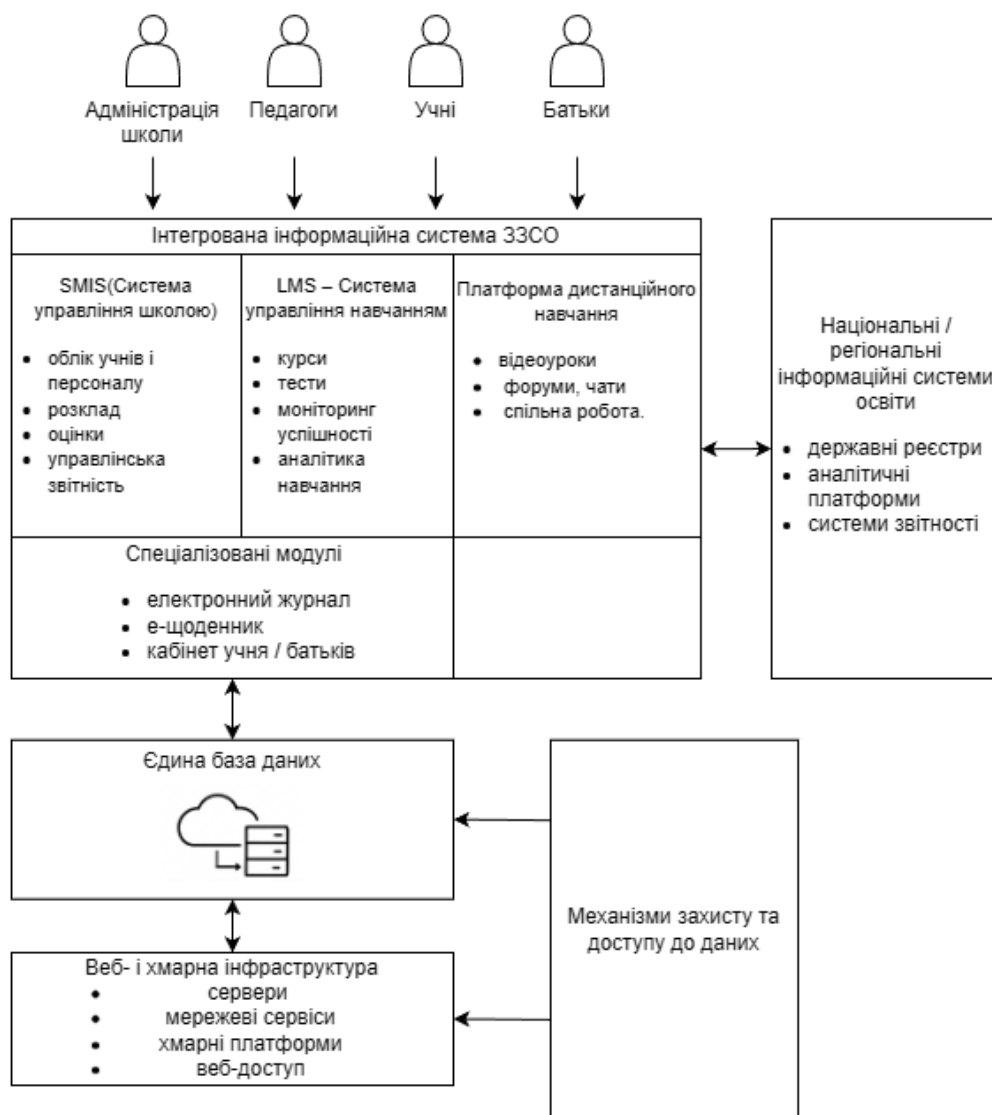


Рисунок 1.2 – Узагальнена структура інформаційної системи управління освітнім процесом закладу загальної середньої освіти

Не менш важливою характеристикою сучасних інформаційних систем є підтримка прийняття рішень на основі даних і можливість систематичного удосконалення. У дослідженнях оцінювання функціональності платформ електронного навчання показано, що система критеріїв, розроблена для українських шкільних платформ, дозволяє не лише порівнювати наявні рішення, а й виявляти прогалини у підтримці комунікації, колаборації та педагогічних моделей, необхідних для якісного дистанційного навчання [33].

У цілому в сучасній науковій та практичній літературі інформаційні системи для управління освітнім процесом розглядаються як багаторівнева сукупність взаємопов'язаних рішень – від локальних модулів електронного журналу до національних інформаційно-аналітичних платформ. Вони мають підтримувати прозорість, ефективність і точність освітніх процесів, створювати умови для змішаного й дистанційного навчання, забезпечувати інтеграцію з іншими цифровими сервісами та відповідати вимогам безпеки й захисту даних. Сучасні платформні підходи до проєктування освітніх інформаційних систем підкреслюють необхідність відкритих стандартів, доступності й орієнтації на користувача, що є вихідними вимогами для проєктування інформаційної системи супроводу освітнього процесу закладу загальної середньої освіти.

1.3 Вимоги до інформаційної системи супроводу освітнього процесу ЗЗСО

Функціональні вимоги до системи впливають із сукупності завдань, що визначаються дослідженнями цифрового освітнього середовища, інформаційних систем управління школою та платформ електронного навчання. У працях, присвячених інформаційно-цифровому середовищу ЗЗСО, підкреслюється необхідність створення відкритого, технологічно насиченого й педагогічно обґрунтованого середовища, де вчителі, учні та шкільна громада виступають активними суб'єктами навчання, спілкування та обміну досвідом. У дослідженнях цифрової компетентності педагогів і описі структури цифрової компетентності вчителя акцент робиться на здатності використовувати цифрові технології для планування, організації, реалізації й оцінювання навчального процесу, а також для комунікації та професійного розвитку. Це безпосередньо трансформується у вимоги до інтерфейсів і сервісів системи, які мають підтримувати педагогічні й управлінські функції без зайвого технічного навантаження.

Перший блок функціональних вимог стосується управління даними про учасників освітнього процесу та структурні характеристики закладу. Система повинна забезпечувати:

- ведення єдиного реєстру здобувачів освіти з фіксацією персональних даних, історії навчання, належності до класів, груп, профілів;
- облік педагогічних та інших працівників, їх посад, навантаження, участі в освітніх програмах;
- моделювання структури закладу, включно з паралелями, класами, групами, навчальними кабінетами та іншими ресурсами;
- фіксацію статусів учнів (переведення, відрахування, академічні відпустки, інклюзивні форми навчання).

Другий блок охоплює вимоги до підтримки навчального процесу. На основі аналізу існуючих систем, зокрема національної платформи «Єдина школа» та веборієнтованих шкільних інформаційних систем у зарубіжній практиці, можна виокремити такі групи функцій:

- планування навчання (формування навчальних планів, мережі класів, розкладу занять з урахуванням ресурсних обмежень і санітарних вимог);
- супровід уроку й навчальних курсів (публікація тем, матеріалів, завдань, посилок на зовнішні ресурси, організація практичних і проєктних видів діяльності);
- облік результатів навчання (тематичне, семестрове, річне оцінювання, формування електронного журналу, підтримка масштабованих шкал оцінювання);
- організація домашніх завдань із контрольними строками та можливістю прикріплення виконаних робіт, тестів, рефлексивних звітів;
- підтримка альтернативних форматів навчання (дистанційне, змішане, асинхронне), що особливо важливо в умовах повітряних тривог, відключення електроенергії та вимушеного переміщення.

Третій блок функціональних вимог пов'язаний із комунікацією та співпрацею. У дослідженнях цифрового освітнього середовища й електронних

журналів підкреслюється, що недостатній рівень цифрової комунікації між учителями, учнями й батьками є одним з ключових бар'єрів для ефективного використання цифрових технологій у школі. Система має підтримувати:

- канали індивідуальних і групових повідомлень між учителями, учнями та батьками з можливістю фільтрації за класом, предметом, подією;
- інформаційні стрічки і оголошення на рівні закладу, класу, окремого курсу;
- інструменти організації онлайн-засідань, консультацій, батьківських зборів на основі інтеграції з сервісами відеоконференцій;
- механізми збору й оброблення зворотного зв'язку, включно з опитуваннями, анкетами, рефлексивними формами.

Четвертий блок формують аналітичні функції та підтримка управлінських рішень. У роботах, присвячених шкільним інформаційним системам та управлінню базами даних в освіті, наголошується, що перехід до цифрових рішень має сенс лише за умови використання накопичених даних для планування, моніторингу та коригування освітнього процесу. Наголошується, що попри очевидні переваги шкільних інформаційних систем, концепція шкільної культури, заснованої на даних, не реалізувалася у багатьох навчальних закладах, зокрема через проблеми якості даних у контексті великих масивів інформації [34]. Цільова система повинна забезпечувати формування дашбордів для адміністрації, класних керівників і вчителів, інструменти аналізу успішності, відвідування, участі в онлайн-активностях, підтримку внутрішньої системи забезпечення якості освіти, а також підготовку регламентованої звітності для органів управління освітою.

П'ятий блок функціональних вимог стосується інтеграції з іншими елементами цифрової екосистеми. У роботах, присвячених національним та регіональним освітнім інформаційно-аналітичним системам, підкреслюється необхідність узгодження структури даних, сумісності та інтеграції локальних систем закладів освіти з державними реєстрами, підключення до централізованих сховищ. Аналіз функціональності платформ електронного

навчання показує, що для систематичного оцінювання їх можливостей використовуються комплексні набори критеріїв. В одному з досліджень сформовано 27 критеріїв оцінювання функціональності платформ електронного навчання, згрупованих у три макрогрупи, що охоплюють керування навчанням, керування контентом та засоби комунікації й співпраці. Ці критерії можуть бути використані як орієнтир для проєктування інтерфейсів інтеграції, експорту й імпорту даних, а також для узгодження навчальних і управлінських модулів.

1.4 Огляд та аналіз існуючих програмних рішень

Огляд програмних рішень, які вже використовуються в закладах загальної середньої освіти для супроводу освітнього процесу, дає можливість конкретизувати вимоги до проєктованої інформаційної системи. У практиці ЗЗСО найбільш поширеними є чотири платформи, що суттєво відрізняються за розробником, технологічним стеком, архітектурою та функціональністю: Google Classroom, Microsoft Teams, Moodle і Єдина школа. Вони реалізують різні моделі організації цифрового освітнього середовища, поєднуючи навчальні, комунікаційні та управлінські компоненти в різних пропорціях.

Для структурованого аналізу доцільно одразу узагальнити базові технічні характеристики і подати їх у вигляді таблиці 1.2.

Таблиця 1.2

Загальні технічні характеристики розглянутих інформаційних систем

Система	Розробник	Мови розробки	Тип архітектури
Google Classroom	Google LLC	Серверна частина на мовах сімейства Java, клієнтська – JavaScript/TypeScript	Хмарний багатокористувацький вебзастосунок, багаторівнева архітектура
Microsoft Teams	Microsoft Corporation	Серверна частина на платформі .NET (C#), клієнтська – JavaScript/TypeScript	Хмарний сервіс у складі Microsoft 365, мікросервісна архітектура
Moodle	Moodle HQ та спільнота з	Основна мова PHP, клієнтська частина –	Веборієнтована система з модульною

	відкритим кодом	JavaScript	структурою й плагінною архітектурою
Єдина школа	Національний проєкт за підтримки МОН України	Вебтехнології серверного боку та JavaScript на клієнтському боці	Централізована хмарна інформаційно-комунікаційна система

1.4.1 Google Classroom

Google Classroom позиціонується як хмарна платформа для організації роботи класів і курсів у межах екосистеми сервісів Google. Розробником виступає Google LLC, а система реалізована як багатокористувацький вебзастосунок із доступом через браузер і смартфони. Архітектурно це багаторівнева хмарна система, у якій серверна частина забезпечує зберігання даних, бізнес-логіку та інтеграцію з іншими сервісами, а клієнтська частина на JavaScript і TypeScript відповідає за інтерфейс користувача. Основні функціональні можливості включають створення та адміністрування курсів, формування й призначення завдань, перевірку робіт, виставлення оцінок, комунікацію в межах курсу, інтеграцію з хмарним сховищем, календарем та іншими сервісами [14].

Додатково платформа підтримує проведення онлайн-занять через інтеграцію з сервісами відеоконференцій, а також забезпечує спільну роботу над документами в межах навчальних проєктів. Для кожного курсу та учня автоматично створюється структура папок у Google Drive, що сприяє впорядкуванню навчальних матеріалів і результатів виконання завдань.

Інтерфейс Google Classroom побудований за принципом мінімалістичного робочого простору з трьома основними зонами: навігаційною панеллю, стрічкою оголошень і завдань та областю перегляду матеріалів конкретного курсу. Така організація знижує поріг входу для педагогів і учнів, які не мають глибоких цифрових навичок. Базовий робочий екран Google Classroom з основними елементами керування можна побачити на рисунку 1.3.

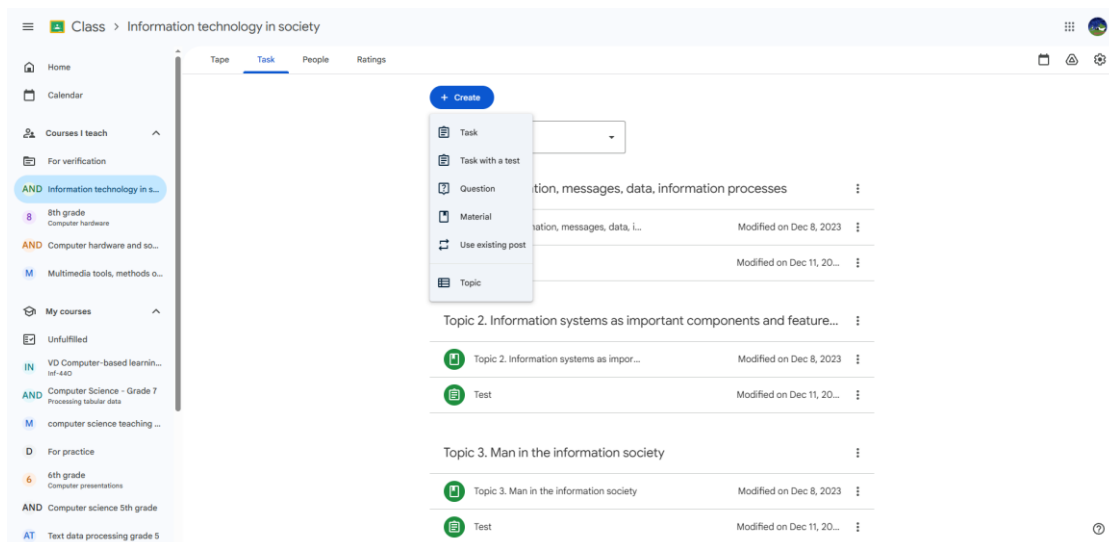


Рисунок 1.3 – Інтерфейс інформаційної системи Google Classroom з основними елементами керування курсом

До переваг Google Classroom належать простота розгортання (потрібен лише обліковий запис у екосистемі Google), зручна інтеграція з хмарними сховищами, календарем і засобами відеоконференцій, невисокі вимоги до апаратних ресурсів клієнтських пристроїв, зрозумілий інтерфейс для учнів. Значущим плюсом є централізоване зберігання всіх матеріалів і робіт у хмарі. До недоліків можна віднести обмежену гнучкість у налаштуванні структури курсів, відносно простий інструментарій аналітики, залежність від наявності стабільного інтернет-з'єднання та прив'язку до облікових записів і політик конфіденційності глобального постачальника.

1.4.2 Microsoft Teams

Microsoft Teams належить до іншого класу рішень: це комунікаційна й колаборативна платформа, інтегрована в екосистему Microsoft 365, яку Microsoft Corporation адаптувала для освітніх сценаріїв через спеціальні типи команд класу та модуль завдань. Серверна частина реалізована на платформі .NET з використанням мікросервісної архітектури й розгорнута в хмарній

інфраструктурі. Клієнтська частина побудована як веб і десктопні програми з використанням JavaScript і TypeScript, а також мобільні додатки [27].

Функціонально Microsoft Teams поєднує кілька рівнів: структурування навчання за командами й каналами, текстові чати, відеонаради, спільні файли, завдання з можливістю перевірки й оцінювання, інтеграцію з електронними нотатниками, календарями та іншими застосунками з каталогу. Для освітнього процесу це означає, що в межах однієї платформи вчитель може проводити онлайн-уроки, поширювати матеріали, давати завдання, коментувати роботи та комунікувати з учнями в синхронному й асинхронному режимах. У типовому режимі використання інтерфейс складається з бокової панелі навігації, списку команд, панелі каналів і центральної області вмісту, де відображаються чати, файли або завдання залежно від обраного режиму. Базовий робочий екран Microsoft Teams з основними елементами керування можна побачити на рисунку 1.4.

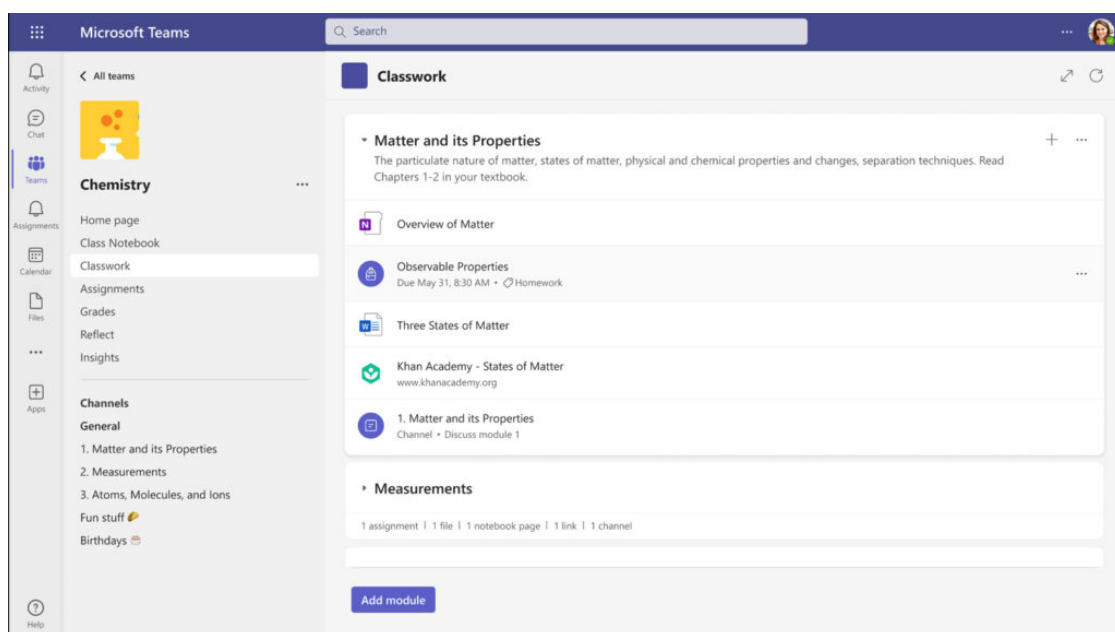


Рисунок 1.4 – Інтерфейс платформи Microsoft Teams під час роботи вчителя з командою класу

До переваг Microsoft Teams належить глибока інтеграція з іншими сервісами Microsoft 365, потужні засоби синхронної взаємодії (відеозв'язок, спільне редагування документів), розвинуті механізми управління доступом і

безпеки на рівні організації, можливість централізованої адміністрації середовища для великої кількості шкіл або цілої громади. Недоліками є високі вимоги до апаратних ресурсів клієнтських пристроїв, складність інтерфейсу для користувачів із низьким рівнем цифрової компетентності, значна залежність від корпоративної інфраструктури та підписки. Для багатьох ЗЗСО саме ресурсомісткість і складність адміністрування стають бар'єром для широкого використання.

1.4.3 Moodle

Moodle репрезентує класичну систему управління навчанням з відкритим кодом. Розробником є Moodle HQ у співпраці зі світовою спільнотою, а основна частина серверної логіки реалізована на мові PHP з активним використанням JavaScript на клієнтському боці. Архітектурно Moodle є веборієнтованою системою з модульною структурою й плагінною архітектурою: ядро містить базові функції, а додаткові модулі підключаються як розширення. Система розгортається на вебсервері закладу або в хмарній інфраструктурі постачальника послуг, що забезпечує гнучкість вибору моделі експлуатації [19].

Функціональні можливості Moodle охоплюють повний цикл роботи з курсом: створення й налаштування структури, додавання ресурсів різних типів, організацію практичної роботи, тестування, оцінювання, аналітику активності, управління ролями й правами доступу. Система підтримує великий спектр видів діяльності, зокрема завдання, тести, форуми, семінари, опитування, глосарії, вбудовані та зовнішні блоки інтеграції. Інтерфейс ядра Moodle історично орієнтований на відображення структури курсу у вигляді тематичних або тижневих блоків з переліком елементів, панеллю навігації та контекстними меню.

Базовий робочий екран Moodle з основними елементами керування можна побачити на рисунку 1.5.

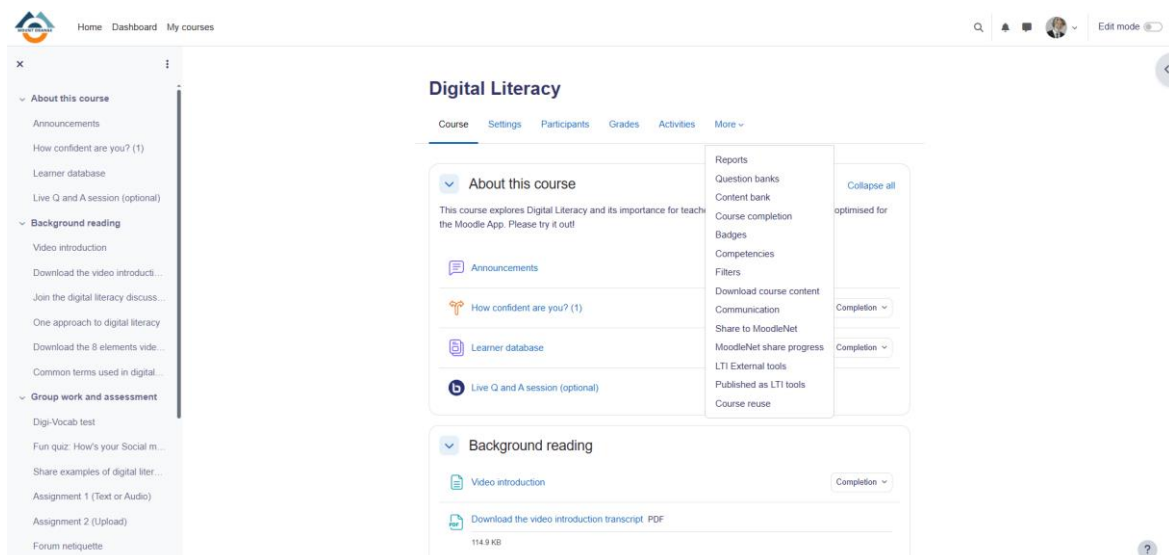


Рисунок 1.5 – Інтерфейс системи Moodle під час роботи з курсом у закладі загальної середньої освіти

До переваг Moodle належать відкритий код, можливість розгортання в інфраструктурі самого закладу, гнучкість у побудові курсів, розширюваність за рахунок плагінів, активна спільнота, наявність локалізацій. Це дозволяє адаптувати систему під специфіку окремої школи, впроваджувати власні модулі й сценарії навчання. Серед недоліків варто зазначити потребу в системному адмініструванні й технічній підтримці, залежність якості роботи від конфігурації серверного середовища, відносну складність інтерфейсу для частини користувачів, а також необхідність окремої інтеграції з електронними журналами, національними реєстрами та іншими інформаційними системами.

1.4.4 Єдина школа

Єдина школа є національною інформаційно-комунікаційною системою, що створена для підтримки освітнього процесу в українських закладах загальної середньої освіти з урахуванням вимог законодавства та управлінських процедур. Розробка й супровід здійснюються в межах державного проєкту за участю ІТ-команди, яка адмініструє платформу та забезпечує її інтеграцію з освітньою інфраструктурою. Система реалізована як централізований вебзастосунок із

серверною логікою на сучасних мовах програмування загального призначення та клієнтською частиною на JavaScript. Дані зберігаються в центральній базі, доступ до якої здійснюється через захищені канали [1].

Функціональність Єдиної школи орієнтована насамперед на управлінські та супровідні аспекти освітнього процесу: ведення електронних журналів і щоденників, формування розкладу, облік відвідування й оцінок, роботу з контингентом учнів, формування звітності, організацію доступу батьків до інформації про успішність, підтримку онлайн-уроків і завдань, базову освітню аналітику. Для користувачів передбачено рольовий доступ: адміністратор закладу, вчителі, учні, батьки, представники органів управління освітою. Інтерфейс платформи структурується навколо електронного журналу, календаря, модулів управління класами і користувачами.

Базовий робочий екран Єдиної школи з основними елементами керування можна побачити на рисунку 1.6.

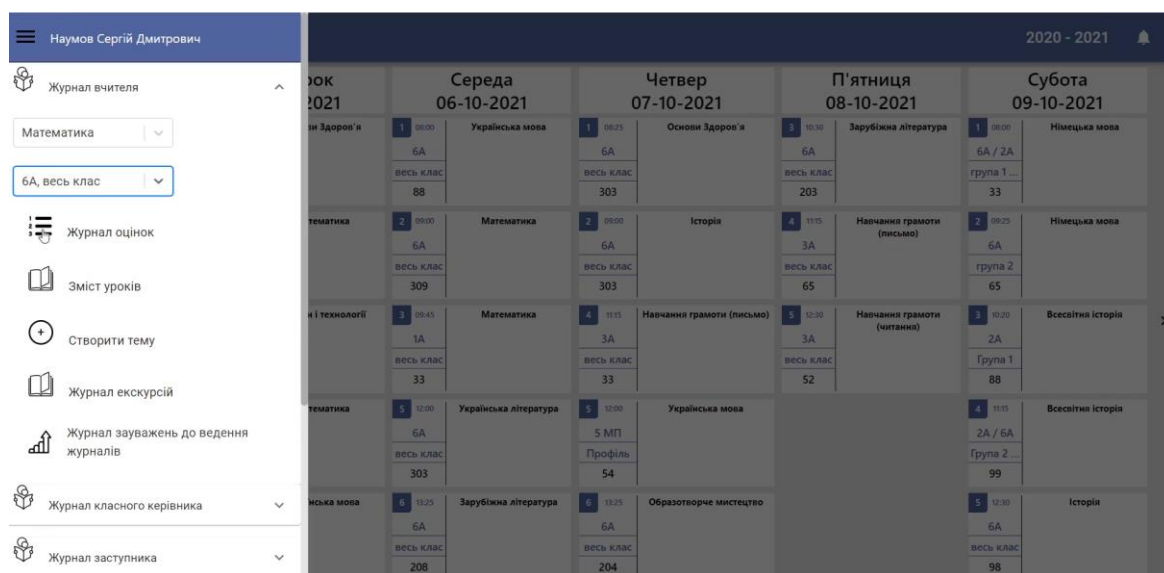


Рисунок 1.6 – Інтерфейс інформаційної системи «Єдина школа» під час роботи вчителя з електронним журналом та навчальними модулями

До переваг Єдиної школи належать відповідність нормативним вимогам до ведення документації, адаптація до української освітньої системи, використання державної термінології й класифікацій, підтримка ролей, пов'язаних із шкільним

управлінням, можливість централізованого адміністрування на рівні територіальних громад і органів управління освітою. Крім того, платформа підтримує базові сценарії організації дистанційного навчання, що важливо в умовах воєнного стану та непередбачуваних змін режиму роботи школи. До недоліків належать обмежена гнучкість у налаштуванні навчальних курсів та інтеграції з зовнішніми сервісами, залежність від стабільності центральної інфраструктури, проблеми сумісності з окремими браузерами, а також потреба у постійній методичній підтримці педагогів при переході до повномасштабного використання електронних журналів.

Для зручності подальшого проектування доцільно згрупувати ключові функціональні можливості, а також переваги й недоліки розглянутих систем у компактному порівняльному форматі у вигляді таблиці 1.3.

Таблиця 1.3

Функціональність, переваги та недоліки основних інформаційних систем

Система	Основна функціональність	Переваги	Недоліки
Google Classroom	Організація курсів і завдань, оцінювання, коментування робіт, інтеграція з хмарним сховищем і календарем	Простота інтерфейсу, швидке розгортання, невисокі вимоги до ресурсів, тісна інтеграція з сервісами	Обмежена гнучкість курсів, базова аналітика, залежність від хмарного постачальника
Microsoft Teams	Команди й канали, чати, відеонаради, завдання, спільні файли, інтеграція з Microsoft 365	Розвинена співпраця, потужні засоби безпеки, інтеграція з корпоративними сервісами	Висока ресурсомісткість, складність для новачків, залежність від організаційної ІТ-інфраструктури
Moodle	Курси, різні види діяльності, тести, журнали, аналітика, ролі й права доступу	Відкритий код, гнучкість, розширюваність, можливість локального розгортання	Потреба в адмініструванні, складніший інтерфейс, необхідність інтеграції з іншими системами
Єдина	Електронний журнал	Відповідність	Обмежена

школа	і щоденник, розклад, облік відвідування й оцінок, доступ батьків, звітність	нормативним вимогам, орієнтація на ЗЗСО, рольовий доступ, централізоване адміністрування	дидактична гнучкість, залежність від центральної інфраструктури, обмежені інтеграційні можливості
-------	---	--	---

Такий формат огляду показує, що кожна з платформ закриває певні сегменти цифрового освітнього середовища, але жодна з них не поєднує повною мірою розширені можливості супроводу навчання, управлінських процесів, аналітики й гнучкої адаптації до кризових умов роботи ЗЗСО.

РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ СУПРОВОДУ ОСВІТНЬОГО ПРОЦЕСУ ЗЗСО

2.1 Постановка задачі, призначення та вимоги до розробки

Призначення розроблюваного програмного засобу полягає в забезпеченні ведення структурних даних закладу, формування розкладу занять, організації роботи з навчальними матеріалами та завданнями, обліку оцінок і відвідуваності, а також у наданні користувачам зручного доступу до актуальної інформації через вебінтерфейс.

Постановка задачі полягає у створенні інформаційної системи супроводу освітнього процесу ЗЗСО, яка дозволить автоматизувати ключові елементи навчальної діяльності: планування, виконання, контроль та аналіз. Система повинна забезпечити зберігання та обробку навчальних даних, підтримку взаємодії між учасниками освітнього процесу та можливість розширення функціональності відповідно до потреб закладу.

Для досягнення поставленої мети система має реалізовувати такі основні вимоги:

- забезпечення реєстрації та автентифікації користувачів;
- підтримка довідників шкільних структур (класи, навчальні предмети, навчальні курси);
- формування та редагування розкладу занять;
- створення та структурування навчальних курсів за темами;
- додавання навчальних матеріалів у межах тем;
- створення завдань із можливістю додавання опису, терміну здачі та прикріплення файлів;
- подання учнями виконаних робіт та їх збереження в системі;
- перевірка поданих робіт та фіксація результатів оцінювання;
- ведення електронного журналу оцінок;
- облік та збереження відвідуваності занять;

- формування базових аналітичних показників щодо виконання завдань, оцінювання та відвідуваності;
- публікація новин та важливих оголошень закладу;
- забезпечення зручного та інтуїтивно зрозумілого інтерфейсу українською мовою;
- гарантування коректної роботи в сучасних браузерях та захисту персональних даних.

2.2 Вибір моделі розробки програмного засобу

У процесі розробки інформаційної системи супроводу освітнього процесу для закладу загальної середньої освіти було обрано та практично застосовано ітераційно-інкрементальну модель. Вибір цієї моделі був зумовлений структурою самої системи, яка складається з відносно незалежних функціональних модулів, а також необхідністю поетапної реалізації та перевірки працездатності кожної частини програмного засобу.

Розробка системи здійснювалася не абстрактно, а у вигляді конкретних послідовних ітерацій. Кожна ітерація завершувалася створенням працездатного інкременту програмного продукту, який можна було протестувати та за потреби доопрацювати перед переходом до наступного етапу.

У межах проєкту ітераційно-інкрементальна модель була реалізована таким чином:

- функціональність системи нарощувалася поетапно, починаючи з базових можливостей;
- після завершення кожної ітерації проводилося тестування реалізованого модуля;
- виявлені недоліки усувалися в межах наступної ітерації без порушення роботи інших компонентів;
- структура системи дозволяла додавати нові модулі без необхідності суттєвої переробки вже реалізованого коду.

На першій ітерації було сформовано базове ядро системи: реалізовано механізми автентифікації, визначено ролі користувачів і налаштовано базове керування обліковими записами. Це забезпечило контроль доступу та створило основу для подальшого підключення навчальних модулів.

Друга ітерація була присвячена формуванню логіки організації освітнього процесу. На цьому етапі реалізовано створення та адміністрування класів і курсів, налаштовано їх взаємозв'язки (закріплення курсів за відповідними класами та викладачами), а також механізми доступу користувачів до даних згідно з їхніми ролями. Окрему увагу приділено коректності структури збереження інформації та її відображення в інтерфейсі, оскільки зазначені сутності є базовими для подальшої роботи всієї системи.

Під час третьої ітерації реалізовано основний навчальний функціонал: роботу з навчальними матеріалами, завданнями, журналом оцінок і обліком відвідуваності. Кожен із цих компонентів перевірявся як окремо, так і в складі всієї системи, щоб забезпечити узгодженість між модулями.

2.3 Опис проєкту програмного засобу

Програмний продукт «Інформаційна система супроводу освітнього процесу закладу загальної середньої освіти» позиціонується як веборієнтований застосунок, призначений для підтримки щоденної роботи школи. Система забезпечує ведення структури закладу, формування розкладу занять, організацію навчальних курсів і завдань, облік оцінок та відвідуваності, а також публікацію новин.

Проєкт передбачає роботу користувачів із різними ролями, що взаємодіють із системою через єдиний вебінтерфейс. Логіка роботи реалізована за шаблоном MVC фреймворку Laravel. Контролери обробляють запити користувачів, моделі відображають таблиці бази даних, а Blade-шаблони відповідають за формування інтерфейсу.

Програмний засіб призначений для автоматизації освітнього процесу та передбачає такі функціональні можливості:

- управління користувачами, що забезпечуватиме створення, редагування та видалення облікових записів, збереження даних у базі даних і розподіл користувачів за ролями;
- управління класами, яке передбачає ведення переліку класів із можливістю закріплення за ними учнів та призначення класних керівників;
- управління курсами та темами, що забезпечуватиме створення навчальних курсів для окремих предметів з прив'язкою до класів і вчителів, а також формування структури курсу у вигляді набору тем;
- формування розкладу занять, яке дозволить створювати розклад уроків із прив'язкою до дати, часу, класу, курсу та вчителя, а також переглядати його для різних категорій користувачів;
- роботу з навчальними матеріалами, що передбачає можливість додавання та зберігання матеріалів до кожної теми курсу з метою забезпечення централізованого доступу учнів;
- створення завдань і подання робіт, яке надаватиме вчителям можливість формувати завдання із зазначенням основних параметрів, а учням — подавати виконані роботи через вебінтерфейс;
- перевірку завдань та оцінювання, що забезпечуватиме перегляд поданих робіт і виставлення оцінок із подальшим формуванням електронного журналу;
- облік відвідуваності, який передбачає фіксацію присутності учнів на заняттях та можливість перегляду історії відвідування за вибраний період;
- ведення електронного журналу оцінок, що зберігатиме результати навчання з прив'язкою до учнів, курсів і дат та відображатиметься у вигляді таблиці з можливістю фільтрації;

- публікацію новин та оголошень, яка забезпечуватиме інформування учнів і вчителів про актуальні події навчального закладу;
- відображення аналітичної інформації у вигляді дашборду з основними показниками роботи системи, такими як середній бал, кількість оцінок і зафіксованих відвідувань.

Логічну структуру взаємодії користувачів із системою подано в додатку Г на рисунку Г.1 у вигляді діаграми варіантів використання UML. На ній відображено основні актори та типові сценарії їхньої взаємодії зі системою: авторизація, робота з курсами й розкладом, організація та виконання завдань, перегляд оцінок і відвідуваності, робота з новинами.

Структуру основних об'єктів предметної області та їхні взаємозв'язки відображено в додатку Г на діаграмі класів UML (рис. Г.2).

База даних програмного засобу побудована на основі цих сутностей і містить, зокрема, таблиці users, school_classes, courses, course_user, schedules, topics, assignments, materials, submissions, grades, attendance, news. Така структура забезпечує нормалізоване зберігання даних, підтримку необхідних зв'язків між об'єктами та можливість подальшого розширення системи.

2.4 Обґрунтування вибору інструментальних засобів розробки

Під час розробки інформаційної системи супроводу освітнього процесу ЗЗСО важливо було обрати інструментальні засоби, які забезпечують швидке створення вебзастосунку, зрозумілу архітектуру, підтримку безпеки, роботу з базою даних, а також зручний інтерфейс адміністративної частини. Був обраний стек Laravel + PHP + AdminLTE/Bootstrap + jQuery + Chart.js. Він дозволяє реалізувати функціональність системи із достатнім запасом для розширення.

Laravel обрано як основний фреймворк серверної частини через його орієнтацію на швидку та структуровану розробку вебзастосунків, наявність готових механізмів маршрутизації, проміжного ПЗ, шаблонізатора та ORM.

Також Laravel надає вбудовані підходи до Authentication/Authorization, що важливо для системи школи, де доступ до даних має бути контрольованим [21].

Ключові переваги Laravel для цього проєкту:

- MVC-архітектура та структурованість коду, що полегшує підтримку модулів (розклад, курси, завдання, журнал, відвідуваність);
- Eloquent ORM для зручної роботи з реляційною БД через моделі, що прискорює розробку та зменшує кількість ручного SQL-коду;
- Blade-шаблонізатор, який компілюється в PHP і кешується, забезпечуючи мінімальні накладні витрати та зручне формування інтерфейсу;
- готова екосистема (міграції, валідація, інструменти для форм, сесій, безпеки), що дає змогу зосередитись на логіці предметної області школи.

PHP обрано як базову мову серверної розробки, оскільки вона є стандартом для веброботи в середовищі хостингів, має велику екосистему бібліотек та напряду підтримується Laravel. Окремо важливим є те, що PHP надає вбудовані механізми безпечного хешування паролів через `password_hash()` і перевірки `password_verify()`, що використовується для захисту облікових записів у системі [29].

Переваги PHP у межах цього проєкту:

- швидке створення CRUD-логіки та інтеграція з Laravel;
- простота розгортання в типовому вебсередовищі навчальних закладів;
- достатня продуктивність для задач школи.

Bootstrap 4.6 використано як основу стилізації та компоновання інтерфейсу. Це обґрунтовано наявністю адаптивної сітки (grid system), утилітарних класів для швидкої верстки, а також готових компонентів форм, кнопок, навігації тощо. У контексті шкільної системи це зменшує час на розробку UI та забезпечує коректне відображення на різних пристроях[23].

AdminLTE v3.x обрано як готовий шаблон для адміністративної частини системи, оскільки він побудований на Bootstrap і надає типові елементи адмін-

інтерфейсу: навігаційне меню, сторінки форм, таблиць, віджетів, базові компоненти для «dashboard» [22].

Переваги AdminLTE для цього проєкту:

- швидке отримання професійного вигляду адмін-панелі без розробки дизайну з нуля;
- стандартні UI-патерни для інформаційних систем (меню, картки, таблиці, форми);
- добре підходить до типових задач школи: керування довідниками, списками, журналами, розкладом.

jQuery використано для реалізації клієнтських інтерактивних елементів, обробників подій та асинхронних запитів. Офіційна документація підкреслює, що jQuery спрощує маніпуляції DOM, обробку подій та Аjax-запити. Для типових сценаріїв системи (динамічні форми, швидкі оновлення списків, фільтрація, AJAX-завантаження даних) це практичне рішення, особливо з урахуванням того, що AdminLTE у класичному використанні часто комбінується саме з jQuery [24].

Chart.js обрано для побудови графіків у модулі аналітики/дашборду, оскільки бібліотека дає змогу швидко створювати діаграми на основі HTML5 canvas і за замовчуванням підтримує адаптивність (responsive charts). Документація також описує механізми оновлення графіків при зміні даних, що корисно для динамічного відображення статистики (наприклад, середній бал, відвідуваність, виконання завдань) [13].

2.5 Програмна реалізація та її особливості

Програмна реалізація інформаційної системи супроводу освітнього процесу ЗЗСО виконана як вебзастосунок з чітким розподілом відповідальностей між рівнями обробки запитів, бізнес-логіки, доступу до даних та представлення інтерфейсу. Основний робочий цикл користувача полягає у переході по сторінках системи, ініціюванні дій, обробці цих дій серверною частиною та

відображенні результату у вигляді вебсторінок або оновлених фрагментів інтерфейсу.

Ключовою особливістю реалізації є те, що логіка додатку організована навколо маршрутів, які виступають «точками входу» для всіх дій користувача. Архітектура фреймворку передбачає, що кожен HTTP-запит потрапляє на відповідний маршрут і лише після цього передається у контролер, де виконується бізнес-операція.

На рисунку 2.1 показано фрагмент конфігурації маршрутів, у якому зосереджено базові переходи й дії системи.

```

1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  use App\Http\Controllers\PageController;
6  use App\Http\Controllers\AuthController;
7  use App\Http\Controllers\ProfileController;
8  use App\Http\Controllers\UserController;
9  use App\Http\Controllers\ScheduleController;
10 use App\Http\Controllers\GradeController;
11 use App\Http\Controllers\CourseController;
12 use App\Http\Controllers\AssignmentController;
13 use App\Http\Controllers\TopicController;
14 use App\Http\Controllers\AttendanceController;
15 use App\Http\Controllers\MaterialController;
16 use App\Http\Controllers\SubmissionController;
17 use App\Http\Controllers\StudentSubmissionController;
18 use App\Http\Controllers\Admin\ClassController as AdminClassController;
19 use App\Http\Controllers\NewsController;
20 use App\Http\Controllers\Admin\NewsController as AdminNewsController;
21
22 Route::get('/', [PageController::class, 'home'])->name('home');
23
24 Route::get('registration', [AuthController::class, 'showRegistrationForm'])->name('registration.form');
25 Route::post('registration', [AuthController::class, 'register'])->name('registration');
26
27 Route::get('login', [AuthController::class, 'showLoginForm'])->name('login.form');
28 Route::post('login', [AuthController::class, 'login'])->name('login');
29 Route::post('logout', [AuthController::class, 'logout'])->name('logout');
30
31 Route::get('profile', [ProfileController::class, 'show'])->name('profile');

```

Рисунок 2.1 – Фрагмент маршрутів авторизації та базової навігації

Доступ до функцій системи пов'язаний із перевіркою особи користувача та коректністю його облікових даних. Вхід реалізовано через контролер, який отримує дані форми, виконує перевірку та створює сесію. На рисунку 2.2 наведено фрагмент програмного коду, що відповідає за виконання операції авторизації.

```

43 public function showLoginForm()
44 {
45     return view('auth.login');
46 }
47
48
49 public function login(Request $request)
50 {
51
52     $request->validate([
53         'username' => 'required',
54         'password' => 'required',
55     ]);
56
57     if (Auth::attempt(['username' => $request->username, 'password' => $request->password])) {
58         return redirect()->route('profile');
59     } else {
60         return back()->withErrors(['username' => 'Невірний логін чи пароль']);
61     }
62 }
63
64
65 public function logout()
66 {
67     Auth::logout();
68     return redirect('/');
69 }
70 }

```

Рисунок 2.2 – Обробка входу користувача

Інтерфейси користувача та повні коди роботи наведено в додатку Б і В.

З точки зору користувача цей процес проявляється у вигляді сторінки входу з формою для введення даних. На рисунку Б.2 показано інтерфейс сторінки авторизації, через яку здійснюється доступ до системи.

Після авторизації система не лише дозволяє користувачу входити, а й обмежує функції відповідно до ролей. Це реалізовано через механізм визначення дозволів, які описують, які дії дозволені різним категоріям користувачів. На рисунку 2.3 показано фрагмент визначення прав доступу, де описані правила для адміністративних дій.

```

17 public function boot(): void
18 {
19     Gate::define('manage-system', function (User $user) {
20         return $user->role === 'адміністратор';
21     });
22     Gate::define('manage-schedules', function (User $user) {
23         return $user->role === 'адміністратор';
24     });
25     Gate::define('view-news', fn(User $user = null) => true);
26
27     Gate::define('create-news', function (User $user) {
28         return in_array($user->role, ['вчитель', 'адміністратор']);

```

Рисунок 2.3 – Визначення правил доступу

Результат такого підходу відображається у поведінці інтерфейсу: користувач бачить лише ті пункти меню, на які він має права. На рисунку 2.4 показано, як у шаблоні інтерфейсу реалізовано умови відображення меню для різних ролей.

```

61     <div class="sidebar">
62         <nav class="mt-2">
63             <ul class="nav nav-pills nav-sidebar flex-column"
64                 data-widget="treeview" role="menu">
65                 @can('manage-students')
66                 <li class="nav-item">
67                     <a href="{{ route('admin.user-list') }}" class="nav-link">
68                         <i class="nav-icon fas fa-users"></i>
69                         <p>Користувачі</p>
70                     </a>
71                 </li>
72                 @endcan
73                 @can('manage-classes')
74                 <li class="nav-item">
75                     <a href="{{ route('classes.index') }}" class="nav-link">
76                         <i class="nav-icon fas fa-school"></i>
77                         <p>Класи</p>
78                     </a>
79                 </li>
80                 @endcan
81                 @can('manage-attendance')
82                 <li class="nav-item">
83                     <a href="{{ route('attendance.index') }}" class="nav-link">
84                         <i class="nav-icon fas fa-calendar-check"></i>
85                         <p>Відвідуваність</p>
86                     </a>
87                 </li>
88                 @endcan
89                 <li class="nav-item">
90                     <a href="{{ route('attendance.my') }}" class="nav-link">
91                         <i class="nav-icon fas fa-calendar-check"></i>
92                         <p>Моя відвідуваність</p>
93                     </a>
94                 </li>

```

Рисунок 2.4 – Рольова навігація в інтерфейсі

Після успішного входу користувач потрапляє до робочого середовища, де зосереджені основні модулі: курси, завдання, матеріали, оцінки, відвідування, розклад і аналітика. Вигляд головного робочого екрану демонструє, що інтерфейс побудовано як єдиний простір доступу до всіх функцій, а навігація – як структура модулів освітнього процесу. Це відображено на рисунку Б.3, де показано інтерфейс домашньої сторінки користувача з меню модулів. Навігація відрізняється залежно від ролі. Різницю можна побачити на рисунку Б.5.

Подальша робота системи спирається на збереження та обробку даних у реляційній базі. Структура бази даних визначається міграціями, що робить її відтворюваною та контрольованою. На рисунку 2.5 наведено приклад міграції таблиці користувачів, яка містить ключові поля (ідентифікатор, дані входу, роль).

```

10     public function up(): void
11     {
12         Schema::create('users', function (Blueprint $table) {
13             $table->id();
14             $table->string('last_name');
15             $table->string('first_name');
16             $table->string('middle_name')->nullable();
17             $table->string('username')->unique();
18             $table->string('email')->nullable()->unique();
19             $table->string('password');
20             $table->enum('role', ['учень', 'вчитель', 'адміністратор'])->default('учень');
21             $table->foreignId('class_id')->nullable()->constrained('school_classes')->onDelete('cascade');
22             $table->rememberToken();
23             $table->timestamps();
24         });

```

Рисунок 2.5 – Міграція таблиці користувачів

Адміністративна частина системи спрямована на керування структурою школи, зокрема класами та контингентом учнів. Важливою функцією є імпорт учнів зі списку, що дозволяє швидко заповнювати базу без ручного введення десятків записів. На рисунку 2.6 показано фрагмент коду імпорту учнів. Повний код можна побачити в додатку В.2.

```

90     public function importCSV(Request $request)
91     {
92         $request->validate([
93             'csv' => 'required|file|mimes:csv,txt'
94         ]);
95
96         $file = fopen($request->file('csv')->getRealPath(), 'r');
97         $header = fgetcsv($file);
98
99         $required = [
100             'grade', 'letter', 'academic_year',
101             'profile', 'description', 'active',
102             'class_teacher_username'
103         ];
104

```

Рисунок 2.6 – Імпорт учнів із CSV

Інтерфейс керування класом, де відображається список учнів і доступний імпорт, показано на рисунку Б.8.

З навчальної точки зору ядром системи виступає модуль курсів, оскільки саме курс об'єднує теми, матеріали, завдання, а також учасників. Логіка роботи цього модуля реалізується через контролер курсів, який відповідає за створення, оновлення та отримання даних для відображення. На рисунку 2.7 показано фрагмент коду створення/оновлення курсу, що демонструє типову реалізацію бізнес-операції через контролер.

```

96     public function store(Request $request)
97     {
98         $this->ensureTeacherOrAdmin();
99
100        $data = $request->validate([
101            'name' => 'required|string|max:255',
102            'subject_name' => 'required|string|max:255',
103            'description' => 'nullable|string',
104            'class_id' => 'required|exists:school_classes,id',
105        ]);
106
107        $course = Course::create([
108            'name' => $data['name'],
109            'subject_name' => $data['subject_name'],
110            'description' => $data['description'] ?? null,
111            'class_id' => $data['class_id'],
112            'teacher_id' => $this->currentUser()->id,
113        ]);
114
115        return redirect()
116            ->route('courses.manage', $course->id)
117            ->with('success', 'Курс створено');
118    }
119
120    public function destroy(Course $course)
121    {
122        $this->ensureCourseOwner($course);
123
124        $course->delete();
125
126        return redirect()
127            ->route('courses.index')
128            ->with('success', 'Курс видалено');
129    }

```

Рисунок 2.7 – Створення і редагування курсу

На рисунку Б.10 показано сторінку курсів, де користувач може переглядати доступні курси та переходити до керування конкретним курсом.

В межах одного курсу реалізована структура навчального наповнення: теми, матеріали та завдання. Інтерфейс керування курсом об'єднує ці компоненти у вигляді сторінки, де вчитель може додавати або редагувати навчальні елементи. Такий робочий простір показано на рисунку Б.15.

Завдання є ключовим елементом контролю навчальної діяльності. Їх створення реалізовано через окремий контролер, який виконує валідацію даних, збереження та прив'язку до курсу. На рисунку 2.8 показано фрагмент реалізації створення завдання.

```

29     public function store(Request $request, Course $course, Topic $topic)
30     {
31         $this->ensureTopicOwner($topic);
32
33         $data = $request->validate([
34             'title'      => 'required|string|max:255',
35             'description' => 'nullable|string',
36             'due_date'   => 'nullable|date',
37             'file'       => 'nullable|file|max:10240',
38         ]);
39
40         $assignment = new Assignment();
41         $assignment->topic_id = $topic->id;
42         $assignment->title = $data['title'];
43         $assignment->description = $data['description'] ?? null;
44         $assignment->due_date = $data['due_date'] ?? null;
45
46         if ($request->hasFile('file')) {
47             $assignment->file_path = $request->file('file')->store('assignments', 'public');
48         }
49
50         $assignment->save();
51
52         return back()->with('success', 'Завдання додано');
53     }
54
55     public function show(Assignment $assignment)
56     {
57         $assignment->load(['topic.course', 'submissions.student']);
58
59         $user = Auth::user();
60
61         return view('assignments.show', compact('assignment', 'user'));
62     }

```

Рисунок 2.8 – Створення завдання

З точки зору інтерфейсу створення завдання представлено як форма введення параметрів (назва, опис, дедлайн). На рисунку 2.9 показано відповідну сторінку або модальне вікно створення завдання.

Рисунок 2.9 – Інтерфейс створення завдання

Після публікації завдання учні отримують можливість завантажити виконану роботу. Реалізація задачі включає обробку файлу, його збереження у сховищі та фіксацію запису в базі даних. На рисунку 2.10 показано фрагмент коду, який приймає роботу учня та зберігає її.

```

13 public function store(Request $request, Assignment $assignment)
14 {
15     $user = Auth::user();
16
17     if ($user->role !== 'учень') {
18         abort(403, 'Тільки учні можуть здавати завдання');
19     }
20
21     $data = $request->validate([
22         'answer' => 'nullable|string',
23         'file'   => 'nullable|file|max:10240',
24     ]);
25
26     $submission = Submission::firstOrCreate([
27         'assignment_id' => $assignment->id,
28         'student_id'   => $user->id,
29     ]);
30
31     $submission->answer = $data['answer'] ?? null;
32
33     if ($request->hasFile('file')) {
34         if ($submission->file_path && Storage::disk('public')->exists($submission->file_path)) {
35             Storage::disk('public')->delete($submission->file_path);
36         }
37
38         $submission->file_path = $request->file('file')->store('submissions', 'public');
39     }
40
41     $submission->save();
42
43     return back()->with('success', 'Відповідь збережено');
44 }

```

Рисунок 2.10 – Задача роботи учня

У користувацькому інтерфейсі цей сценарій відображається як сторінка задачі з формою завантаження файлу. На рисунку Б.20 показано інтерфейс задачі роботи учнем.

Після надходження робіт викладач має інструмент перегляду та оцінювання. Це реалізовано як сторінка зі списком задач та можливістю виставити оцінку. Такий інтерфейс показано на рисунку Б.16.

Важливими складовими супроводу освітнього процесу є облік успішності та відвідування. Для оптимізації роботи вчителя система містить «живі» фільтри, які дозволяють оновлювати таблицю результатів без перезавантаження сторінки. На рисунку 2.11 показано серверний метод фільтрації відвідування, який повертає дані для таблиці у вигляді JSON.

```

96     public function filter(Request $request)
97     {
98         Gate::authorize('manage-attendance');
99
100        $query = Attendance::with(['user', 'schoolClass', 'teacher']);
101
102        if ($request->date) {
103            $query->whereDate('date', $request->date);
104        }
105
106        if ($request->class_id) {
107            $query->where('class_id', $request->class_id);
108        }
109
110        if ($request->subject) {
111            $query->where('subject_name', 'like', '%' . $request->subject . '%');
112        }
113
114        if ($request->status) {
115            $query->where('status', $request->status);
116        }
117
118        if ($request->search) {
119            $search = $request->search;
120            $query->whereHas('user', function ($q) use ($search) {
121                $q->where('last_name', 'like', '%' . $search . '%')
122                    ->orWhere('first_name', 'like', '%' . $search . '%');
123            });
124        }
125    }

```

Рисунок 2.11 – Фільтрація відвідування на сервері

Клієнтська частина динамічного оновлення реалізована у JavaScript-файлі: відправлення запиту, отримання HTML-фрагмента та підстановка його в таблицю. На рисунку 2.12 показано фрагмент цієї логіки.

```

194     document.addEventListener("DOMContentLoaded", () => {
195
196         async function loadAttendanceTable() {
197
198             const token = document
199                 .querySelector('meta[name="csrf-token"]')
200                 ?.getAttribute('content');
201
202             const payload = {
203                 date: document.getElementById('filter_date').value,
204                 class_id: document.getElementById('filter_class').value,
205                 subject: document.getElementById('filter_subject').value,
206                 status: document.getElementById('filter_status').value,
207                 search: document.getElementById('filter_search').value,
208             };
209
210             console.log("📤 Sending filter:", payload);
211
212             try {
213                 let res = await fetch("/attendance/filter", {
214                     method: "POST",
215                     headers: {
216                         "Content-Type": "application/json",
217                         "X-CSRF-TOKEN": token,
218                     },
219                     body: JSON.stringify(payload)
220                 });
221
222                 console.log("📥 Filter status:", res.status);
223
224                 let json = await res.json();
225
226                 document.getElementById('attendanceTableBody').innerHTML = json.html;
227
228             } catch (err) {
229                 console.error("❌ Filter error:", err);
230             }
231         }
232
233         ["filter_date", "filter_class", "filter_subject", "filter_status", "filter_search"]
234             .forEach(id => {
235                 let el = document.getElementById(id);
236                 el?.addEventListener("input", loadAttendanceTable);
237                 el?.addEventListener("change", loadAttendanceTable);
238             });
239
240     });
241

```

Рисунок 2.12 – Динамічне оновлення таблиці

Практичний результат цієї реалізації видно в інтерфейсі сторінки відвідування. На рисунку Б.17 показано сторінку обліку відвідування з фільтрами та таблицею.

Аналогічний підхід застосовано і для оцінок: сервер виконує вибірку за параметрами, а таблиця результатів оновлюється «на льоту». На рисунку Б.18 показано сторінку журналу оцінок з елементами швидкого редагування.

Організаційний аспект освітнього процесу реалізовано через модуль розкладу. Перегляд розкладу здійснюється через інтерфейс з перемиканням днів або тижнів. На рисунку Б.12 показано сторінку розкладу занять.

Для зменшення ручного введення в типовому шкільному сценарії реалізовано механізм створення розкладу на день (рис. 2.13). Повний код додаток В.3.

```

109 public function day(Request $request)
110 {
111     $date = $request->get('date', now()->toDateString());
112     $class = $request->get('class_name', '');
113
114     $query = Schedule::whereDate('date', $date)->orderBy('start_time');
115     if ($class !== '') {
116         $query->where('class_name', $class);
117     }
118     $schedules = $query->get();
119
120     $classes = Schedule::select('class_name')->distinct()->orderBy('class_name')->pluck('class_name');
121
122     return view('schedules.day', compact('date', 'class', 'schedules', 'classes'));
123 }
124

```

Рисунок 2.13 – Створення розкладу на день

Окремим блоком системи виступає аналітика, що дозволяє візуалізувати успішність та активність учнів за допомогою графіків. Дані аналітики формуються контролером у вигляді масивів значень і передаються у шаблон для подальшої візуалізації. На рисунку 2.14 наведено фрагмент програмного коду, який демонструє підготовку даних для побудови графіків. Повна реалізація відповідного методу контролера наведена у додатку В.1.

```

174 public function analytics(Course $course)
175 {
176     $this->ensureCourseOwner($course);
177
178     $course->load(['assignments.submissions.student', 'students']);
179
180     $assignments = $course->assignments()->with('submissions')->get();
181
182     // Статистика по завданнях
183     $assignmentStats = $assignments->map(function (Assignment $a) {
184         $grades = $a->submissions->pluck('grade')->filter();
185         return [
186             'id'          => $a->id,
187             'title'       => $a->title,
188             'submissions' => $a->submissions->count(),
189             'graded_count' => $grades->count(),
190             'avg_grade'   => $grades->count() ? round($grades->avg(), 2) : null,
191             'max_grade'   => $grades->count() ? $grades->max() : null,
192             'min_grade'   => $grades->count() ? $grades->min() : null,
193         ];
194     });
195
196     $students = $course->students()->get();
197
198     // Онлайн-оцінки (submissions)
199     $perStudentOnline = [];
200     foreach ($students as $s) {
201         $subs = Submission::whereIn('assignment_id', $assignments->pluck('id'))
202             ->where('student_id', $s->id)
203             ->get();
204
205         $grades = $subs->pluck('grade')->filter();
206

```

Рисунок 2.14 – Формування даних аналітики

Відображення аналітики в інтерфейсі виконується на сторінці, де виводяться графіки та таблиці. На рисунку 2.15 показано приклад сторінки аналітики курсу.

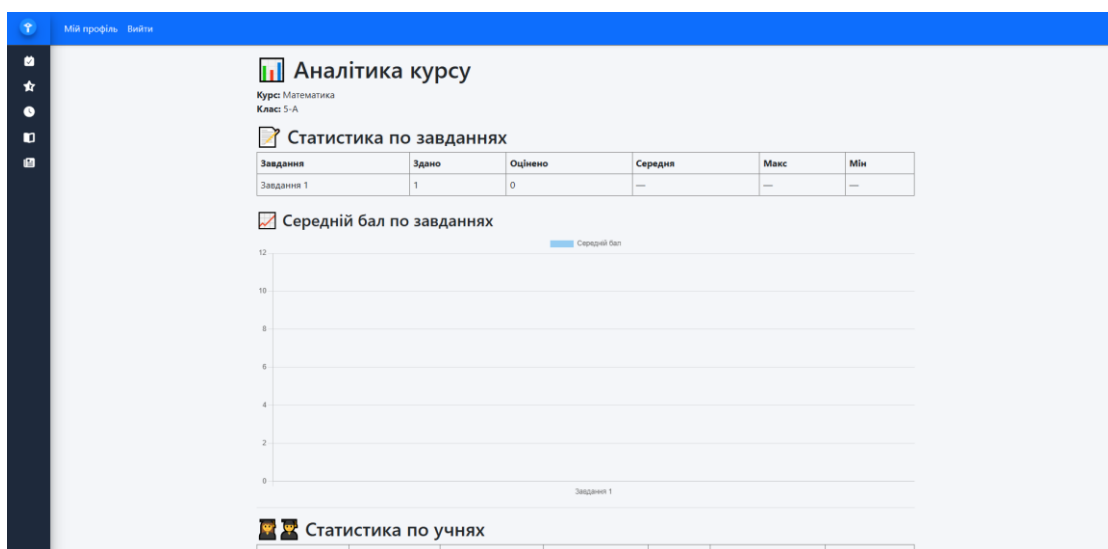


Рисунок 2.15 – Сторінка аналітики

2.6 Тестування та налагодження програмного засобу

У процесі розробки застосовувалося функціональне та інтеграційне тестування, яке проводилося поетапно разом із реалізацією основних модулів. Функціональне тестування включало перевірку базових сценаріїв роботи користувачів, зокрема авторизації, доступу до функцій відповідно до ролей, створення та перегляду курсів, роботи із завданнями, оцінками, відвідуванням і розкладом занять.

Інтеграційне тестування було спрямоване на перевірку коректності взаємодії між клієнтською та серверною частинами системи. Особлива увага приділялася правильності обробки HTTP-запитів, передачі даних між маршрутами, контролерами та представленнями, а також стабільності роботи при виконанні послідовних дій користувача.

Окремо проводилося мануальне тестування інтерфейсу користувача, під час якого перевірялася зручність навігації, коректність відображення даних та робота форм введення. За результатами тестування встановлено, що інтерфейс є інтуїтивно зрозумілим та не викликає труднощів під час роботи.

Налагодження програмного засобу включало виправлення логічних помилок, перевірку обробки некоректних або порожніх вхідних даних, а також усунення неузгодженостей між модулями. Проведені перевірки підтвердили стабільну роботу системи в сучасних веббраузерах та її готовність до практичного використання в закладі загальної середньої освіти.

2.7 Рекомендації по впровадженню та використанню програмного засобу

Розроблений програмний засіб призначений для використання в закладах загальної середньої освіти з метою автоматизації та спрощення супроводу освітнього процесу. Для успішного впровадження та ефективної експлуатації

системи доцільно дотримуватися ряду рекомендацій, що стосуються організаційних, технічних та експлуатаційних аспектів.

Перед впровадженням програмного засобу в навчальному закладі рекомендується провести попереднє тестування системи з використанням тестових облікових записів. Це дозволить перевірити коректність роботи основних функцій, зокрема авторизації користувачів, доступу до модулів відповідно до ролей, роботи з курсами, завданнями, оцінюванням та переглядом навчальної інформації.

Впровадження системи доцільно здійснювати поступово, починаючи з обмеженого кола користувачів (адміністрації або окремих педагогів). Такий підхід дозволяє адаптувати систему до конкретних умов навчального закладу, зібрати первинний зворотний зв'язок та, за необхідності, внести корективи перед повномасштабним використанням.

Для забезпечення ефективної роботи програмного засобу рекомендується організувати ознайомлення користувачів із функціональними можливостями системи. Це може бути короткий інструктаж або надання базової документації, що пояснює принципи роботи з інтерфейсом, порядок створення курсів, додавання матеріалів та виконання інших основних дій.

Особливу увагу слід приділити питанням безпеки та збереження даних. Рекомендується використовувати надійні облікові дані для входу в систему, обмежувати доступ до адміністративних функцій та регулярно перевіряти коректність роботи механізмів авторизації. Це дозволить забезпечити конфіденційність інформації та захист персональних даних учасників освітнього процесу.

Для подальшого розвитку програмного засобу доцільно здійснювати моніторинг його використання та збирати відгуки користувачів. Отримана інформація може бути використана для вдосконалення інтерфейсу, розширення функціональності або оптимізації окремих модулів системи.

Дотримання наведених рекомендацій сприятиме успішному впровадженню програмного засобу в освітній процес, підвищенню ефективності

його використання та забезпеченню стабільної роботи інформаційної системи в умовах закладу загальної середньої освіти.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було досліджено сучасний стан цифрової трансформації закладів загальної середньої освіти та обґрунтовано доцільність створення єдиної інформаційної системи для супроводу освітнього процесу. Аналіз показав, що використання розрізнених цифрових сервісів не забезпечує цілісності управління навчально-організаційними процесами й ускладнює доступ до актуальних даних, особливо в умовах зовнішніх загроз і нестабільної інфраструктури.

У результаті роботи було сформовано концептуальну модель та архітектуру інформаційної системи, визначено її функціональні й нефункціональні вимоги, обґрунтовано вибір моделі розробки та інструментальних засобів. На основі фреймворку Laravel реалізовано веборієнтований програмний продукт, який забезпечує управління користувачами й ролями, класами та курсами, формування розкладу, організацію навчальних матеріалів і завдань, облік оцінок і відвідуваності, а також інформування учасників освітнього процесу через систему новин.

Проведене тестування підтвердило коректність роботи основних функціональних модулів та відповідність програмного засобу поставленим вимогам. Розроблена інформаційна система може бути використана як готове рішення або прототип для впровадження в закладах загальної середньої освіти з метою підвищення керованості освітнього процесу, зменшення кількості ручних операцій і забезпечення безперервності навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Єдина Школа. Єдина Школа. URL: <https://eschool-ua.com/>
2. Канал "Єдина Школа". URL: <https://www.youtube.com/@eschool-ua/videos>
3. Концептуальні засади цифровізації освітнього середовища закладу загальної середньої освіти / О. Ляшенко та ін. *Information Technologies and Learning Tools*. 2024. Т. 102, № 4. С. 1–25. URL: <https://doi.org/10.33407/itlt.v102i4.5829>
4. Литвинова С., Мельник О., Сухіх А. Електронний журнал як інструмент цифрової освітньої комунікації: результати всеукраїнського опитування. *Information Technologies and Learning Tools*. 2024. Т. 100, № 2. С. 141–161. URL: <https://doi.org/10.33407/itlt.v100i2.5516>
5. Науково-методичне забезпечення цифровізації освіти України: стан, проблеми, перспективи / В. Г. Кремень та ін. *Herald of the National Academy of Educational Sciences of Ukraine*. 2022. Т. 4, № 2. С. 1–49. URL: <https://doi.org/10.37472/v.naes.2022.4223>
6. Організаційно-педагогічні умови використання інформаційно-цифрового середовища закладу загальної середньої освіти / О. В. Овчарук та ін. *Information Technologies and Learning Tools*. 2023. Т. 95, № 3. С. 41–57. URL: <https://doi.org/10.33407/itlt.v95i3.5186>
7. Серденко Т. В., Рейс Т. Т. Персоналізація навчання за допомогою інформаційних систем. *Вісник Херсонського національного технічного університету*. 2025. Т. 2, № 2(93). С. 333–338. URL: <https://doi.org/10.35546/kntu2078-4481.2025.2.2.41>
8. Степанівна С. О. Принципи побудови сучасного цифрового освітнього середовища. *Академічні візії*. 2024. № 27. С. 1–5. URL: <https://academy-vision.org/index.php/av/article/view/779/705>
9. Сторонська О., Воробель М. Професійна компетентність педагога в умовах цифровізації освіти. *Молодь і ринок*. 2024. № 10/218. С. 72–76. URL: <https://doi.org/10.24919/2308-4634.2023.290462>

10. Яновський А. Інформаційно-освітнє середовище в умовах дистанційного навчання. *Humanities science current issues*. 2020. Т. 4, № 30. С. 310–315. URL: <https://doi.org/10.24919/2308-4863.4/30.212627>
11. Abdulkadhim E. G., Abbas Z. M., Hayder M. A. E-School Information Management System (E-SIMS). *Central Asian Journal Of Mathematical Theory And Computer Sciences*. 2025. Vol. 6, no. 3. P. 263–278. URL: <https://cajmtcs.casjournal.org/index.php/CAJMTCS/article/view/746/749>
12. Bui T. T., Nguyen T. S. The Survey of Digital Transformation in Education: A Systematic Review. *International Journal of TESOL & Education*. 2023. Vol. 3, no. 4. P. 32–51. URL: <https://doi.org/10.54855/ijte.23343>
13. Chart.js | Chart.js. Chart.js | Open source HTML5 Charts for your website. URL: <https://www.chartjs.org/docs/latest/>
14. Classroom Management Tools & Resources - Google for Education. *Google for Education*. URL: <https://classroom.google.com/>
15. Cloud Computing in Education: A Revolution in Learning Management Systems / P. Sikarwar et al. *Innovare Journal of Education*. 2025. P. 10–17. URL: <https://doi.org/10.22159/ijoe.2025v13i5.54161>
16. Description of educator's digital competence / n. Morze et al. *Open educational e-environment of modern university*. 2019. Special edition. URL: <https://doi.org/10.28925/2414-0325.2019s39>
17. Designing Information System for Vocational School in Minahasa Regency. *Nanotechnology Perceptions*. 2024. Vol. 20, no. 6. URL: <https://doi.org/10.62441/nano-ntp.v20i6.105>
18. Digital transformation of education as a trend in the development of educational reforms and a process of social and cultural change / n. Metelenko et al. *Humanities studies*. 2023. Vol. 93, no. 16. P. 122–134. URL: <https://doi.org/10.32782/hst-2023-16-93-13>
19. Home | Moodle.org. Home | Moodle.org. URL: <https://moodle.org/?lang=id>
20. Hrytsai Y. Digital educational environment of general secondary education institutions. *Image Of The Modern Pedagogue*. 2024. Vol. 1, no. 4. P. 37–41.

URL: [https://doi.org/10.33272/2522-9729-2024-4\(217\)-37-41](https://doi.org/10.33272/2522-9729-2024-4(217)-37-41)

21. Installation - Laravel 12.x - The PHP Framework For Web Artisans. Laravel - The PHP Framework For Web Artisans. URL: <https://laravel.com/docs/12.x>

22. Introduction. AdminLTE v3.2 Documentation. URL: <https://adminlte.io/docs/3.2/>

23. Introduction. Bootstrap · The most popular HTML, CSS, and JS library in the world. URL: <https://getbootstrap.com/docs/4.6/getting-started/introduction/>

24. jQuery API Documentation. jQuery API Documentation. URL: <https://api.jquery.com/>

25. Lontaan R. J., Sinadia A. R. Design and Development of a Web-Based School Information System. CogITO Smart Journal. 2024. Vol. 10, no. 2. P. 593–606. URL: <https://doi.org/10.31154/cogito.v10i2.858.593-606>

26. Marginingsih P., Kusumaningsih W., Violinda Q. Digitalization Management in Schools : A Strategic Framework for Enhancing Academic Quality. Jurnal Paedagogy. 2025. Vol. 12, no. 2. P. 391. URL: <https://doi.org/10.33394/jp.v12i2.15039>

27. Microsoft Teams for Schools and Students | Microsoft Education. URL: <https://www.microsoft.com/en-us/education/products/teams>

28. OECD Digital Education Outlook 2023: Towards an Effective Digital Education Ecosystem. Organization for Economic Cooperation & Development, 2023.

29. PHP: PHP Manual - Manual. PHP. URL: <https://www.php.net/manual/en/>

30. Report scientific conference of the Institute for Digitalisation of Education of the NAES of Ukraine " Digital transformation of education in Ukraine under martial law": collection of materials. Report scientific conference of the Institute for Digitalisation of Education of the NAES of Ukraine " Digital transformation of education in Ukraine under martial law" 2023. URL: <https://doi.org/10.33407/lib.naes.735053>

31. The Usage of Automated Information Systems for Database Management in General Secondary Education Institutions of Ukraine / T. Kravchynska et al. IJCSNS International Journal of Computer Science and Network Security. 2022. Vol. 5, no. 22. P. 497–506. URL: <https://doi.org/10.22937/IJCSNS.2022.22.5.69>

32. Tolochko S. Digital competence of teachers in the conditions of digitalization of

educational establishments and distance learning. 2021.

URL: <https://doi.org/10.5281/zenodo.5077823>

33. Ukrainian E-Learning Platforms for Schools: Evaluation of Their Functionality / M. Zhenchenko et al. The International Review of Research in Open and Distributed Learning. 2022. Vol. 23, no. 2. P. 136–150.

URL: <https://doi.org/10.19173/irrodl.v23i2.5769>

34. Vivienne V. F. School Management Information Systems: Challenges to Educational Decision-Making in the Big Data ERA. 2019.

URL: <https://doi.org/10.5281/zenodo.2624163>

ДОДАТКИ

ДОДАТОК А

Технічне завдання на проєктування та розробку програмного засобу

1. ВСТУП

1.1 Назва програмного продукту

Інформаційна система супроводу освітнього процесу закладу загальної середньої освіти.

1.2 Коротка характеристика програмного продукту

Розроблений програмний засіб є веборієнтованою інформаційною системою, призначеною для автоматизації та оптимізації освітнього процесу в закладах загальної середньої освіти. Система забезпечує централізований доступ до навчальної інформації, підтримує взаємодію між учнями, вчителями та адміністрацією школи, а також дозволяє ефективно керувати навчальними матеріалами, курсами та результатами навчання.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

2.1 Документ, на підставі якого ведеться розробка

Розробка програмного засобу здійснюється відповідно до завдання на проєктування і розробку, виданого науковим керівником у межах виконання кваліфікаційної (дипломної) роботи.

2.2 Організація, що затвердила документ

Волинський національний університет імені Лесі Українки.

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Програмний продукт призначений для використання в закладах загальної середньої освіти з метою підтримки та автоматизації основних процесів освітньої діяльності. Система дозволяє організувати навчальний процес, забезпечити зручний доступ до навчальних матеріалів, здійснювати контроль за виконанням завдань та покращити комунікацію між учасниками освітнього процесу.

4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

4.1 Функціональні вимоги

Програмний засіб повинен забезпечувати виконання таких основних функцій:

- авторизацію та автентифікацію користувачів системи;
- підтримку різних ролей користувачів (адміністратор, вчитель, учень);
- керування навчальними курсами та предметами;
- додавання, редагування та перегляд навчальних матеріалів;
- взаємодію користувачів із навчальним контентом через вебінтерфейс;
- збереження та обробку даних у централізованій базі даних;
- коректну обробку помилок та повідомлення користувачів про стан виконання операцій.

4.2 Вимоги до інтерфейсу

Інтерфейс користувача повинен бути інтуїтивно зрозумілим, зручним у використанні та адаптованим для різних категорій користувачів. Навігація в системі має бути логічною та послідовною, а всі основні функції – доступними через веббраузер.

4.3 Вимоги до безпеки

Програмний продукт повинен забезпечувати захист персональних даних користувачів, реалізовувати механізми авторизації доступу відповідно до ролей та використовувати безпечні методи збереження облікових даних.

4.4 Вимоги до технічних засобів

Тип ЕОМ: персональні комп'ютери та сервери;

Операційні системи: Windows, macOS, Linux;

Мінімальний обсяг оперативної пам'яті: 4 ГБ;

Програмне забезпечення: сучасний веббраузер з підтримкою HTML5 та CSS3;

Доступ до мережі Інтернет.

5. ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

До програмного засобу повинна бути підготовлена документація, яка містить:

- загальний опис системи;
- функціональне призначення програмного продукту;
- умови встановлення та використання;
- опис роботи основних модулів;
- рекомендації щодо тестування та налагодження.

6. СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

Розробка програмного засобу передбачає виконання таких етапів:

- аналіз предметної області та існуючих програмних рішень;
- формування вимог до програмного продукту;
- проектування архітектури системи;
- розробка серверної та клієнтської частин;
- реалізація функціоналу системи;
- тестування та налагодження програмного засобу;
- підготовка програмної документації;
- формування рекомендацій щодо впровадження та використання.

7. ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Програмний продукт вважається прийнятим до експлуатації за умови виконання таких вимог:

- система коректно виконує авторизацію користувачів;
- функціонал відповідає визначеним ролям користувачів;
- дані коректно зберігаються та обробляються;
- інтерфейс працює стабільно та без помилок;
- програмний засіб забезпечує стабільну роботу при стандартному навантаженні.

ДОДАТОК Б

Інструкція користувача

Даний документ призначений для користувачів інформаційної системи супроводу освітнього процесу закладу загальної середньої освіти. Інструкція описує порядок роботи з вебінтерфейсом системи та основні сценарії її використання з точки зору кінцевого користувача.

Програмний засіб надає можливість централізованого зберігання та опрацювання інформації, пов'язаної з освітнім процесом, зокрема управління користувачами, навчальними даними та взаємодією між учасниками освітнього процесу.

Доступ до системи здійснюється через веббраузер. Користувачу не потрібно встановлювати додаткове програмне забезпечення або виконувати технічні налаштування – робота здійснюється виключно через графічний інтерфейс.

1. Початкова сторінка системи

Після переходу за URL користувач потрапляє на головну сторінку з коротким описом системи та навігаційними елементами. На рисунку Б.1 показано вигляд стартової сторінки та розташування кнопки входу.

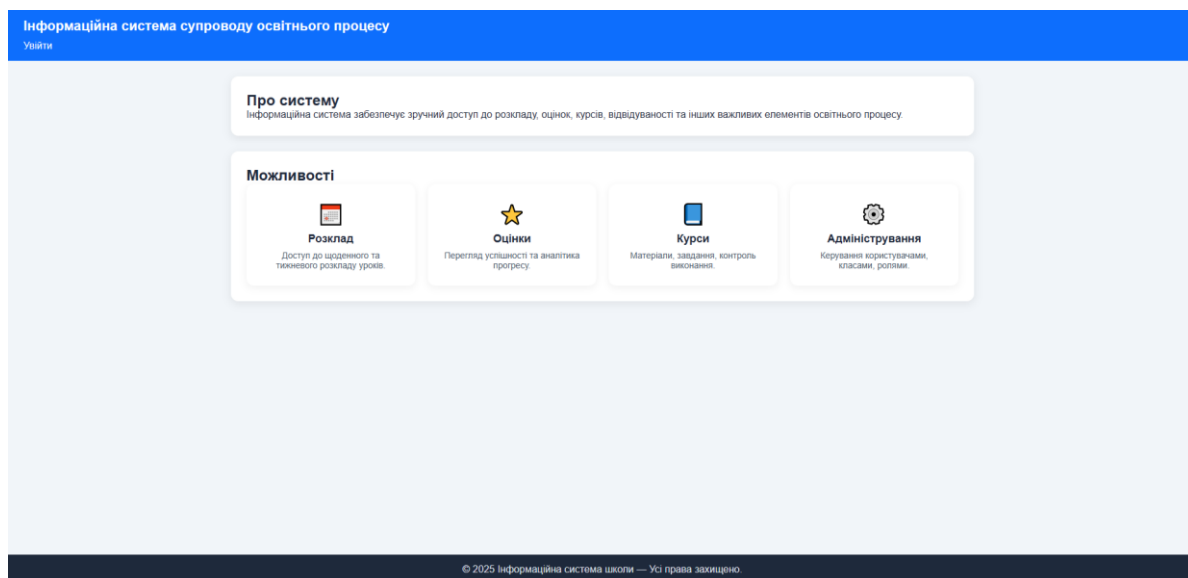


Рисунок Б.1 – Головна сторінка системи

Основні дії на головній сторінці:

- перейти до авторизації (кнопка/посилання «Увійти»);
- переглянути загальний опис призначення системи.

2. Вхід до системи та завершення роботи

2.1 Авторизація користувача

Щоб розпочати роботу, потрібно натиснути «Увійти» на головній сторінці (рис. Б.1). Відкривається форма входу (рис. Б.2).

Рисунок Б.2 – Сторінка входу

У формі входу користувач:

- 1) вводить логін, який надав адміністратор;
- 2) вводить пароль, отриманий від адміністратора;
- 3) натискає кнопку входу.

Важливо: самостійна реєстрація користувачів не використовується (облікові записи створює адміністратор).

Після успішного входу користувач переходить у свій робочий простір (профіль/панель роботи), вигляд якого залежить від ролі (адміністратор, вчитель, учень). Приклад сторінки після входу наведено на рис. Б.3.

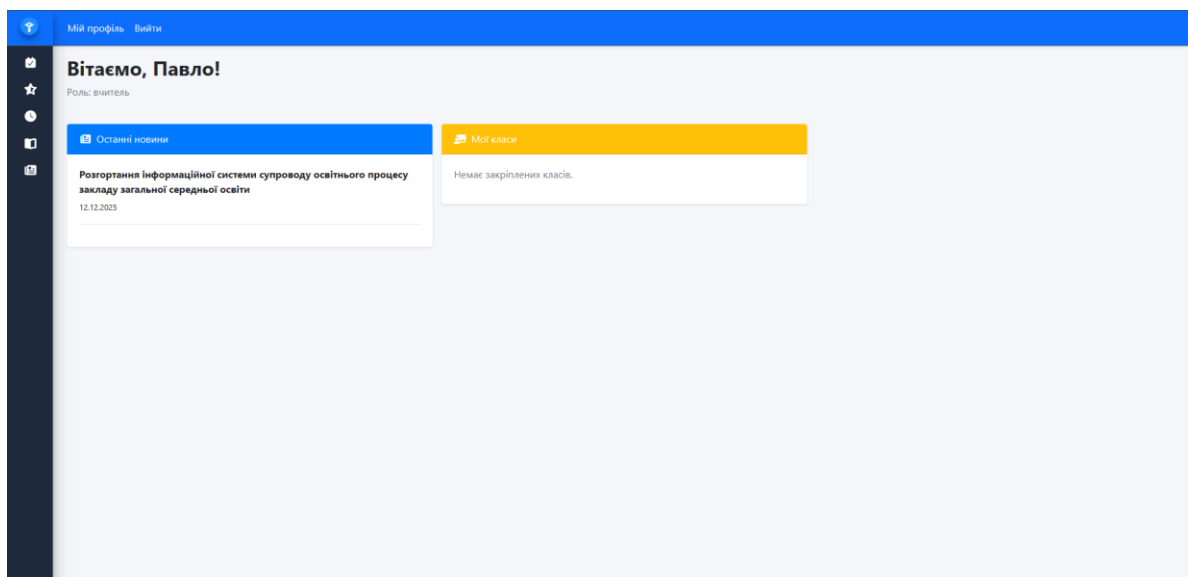


Рисунок Б.3 – Сторінка профілю після входу

2.2 Вихід із системи

Для завершення роботи необхідно натиснути «Вийти» у верхній панелі (рис. Б.4). Після виходу система повертає користувача на публічну частину або сторінку входу.

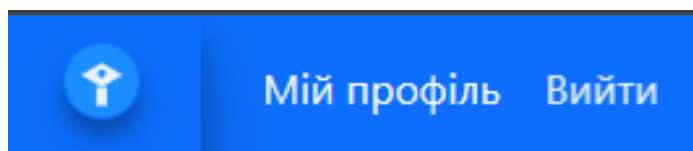


Рисунок Б.4 – Кнопка виходу з системи

3. Загальні елементи інтерфейсу

- Після входу користувач працює в інтерфейсі, що має типову структуру:
- верхня панель (перехід у профіль, вихід із системи);
- бокове меню з основними розділами;
- робоча область (таблиці, форми, сторінки розкладу, курси, оцінки тощо).

На рисунку Б.5 показано приклад розташування бокового меню та основних пунктів навігації.

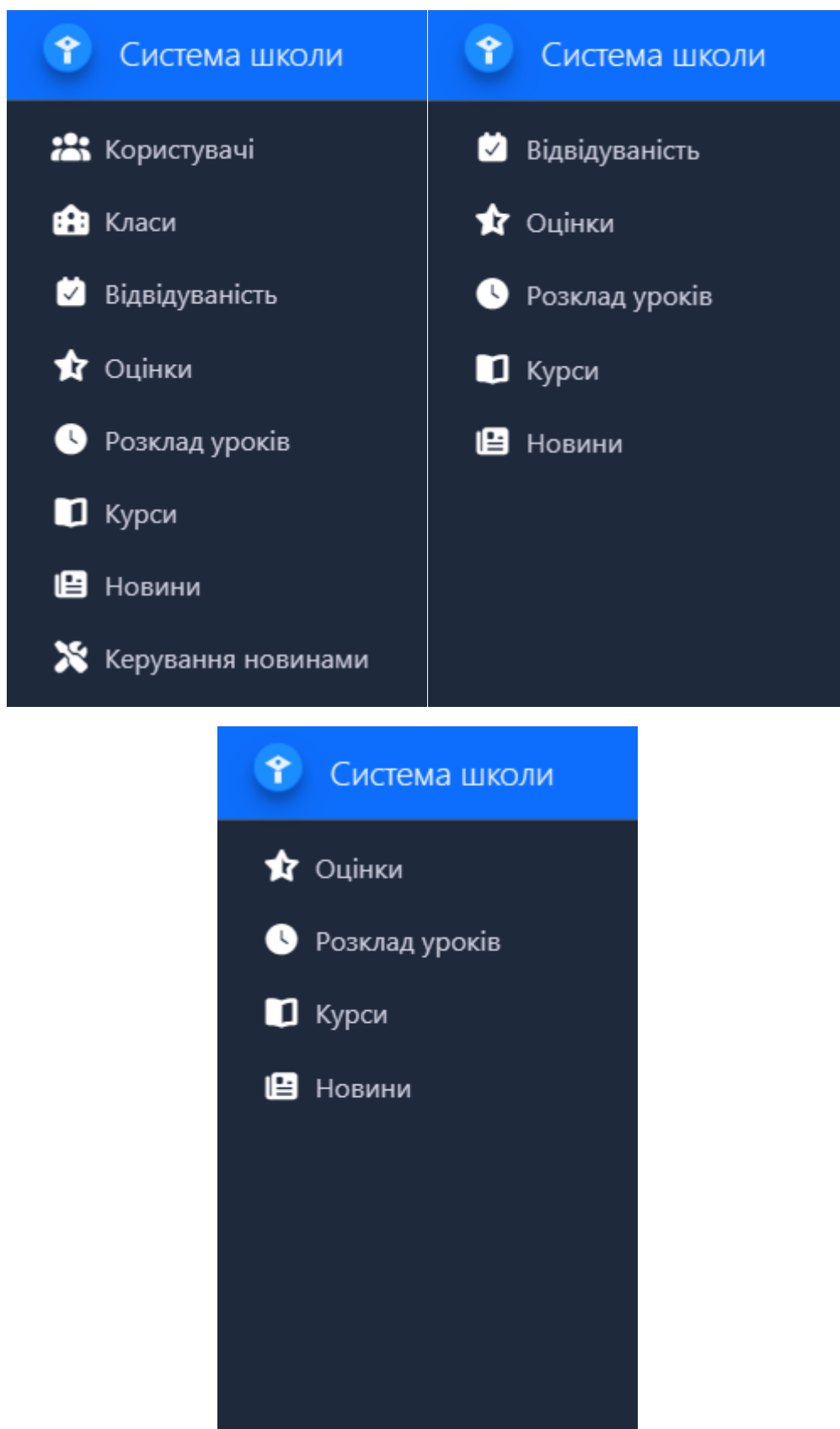


Рисунок Б.5 – Бокове меню та навігація для ролі адміністратора, вчителя і учня

Набір пунктів у меню може відрізнятися залежно від ролі користувача та наданих прав доступу.

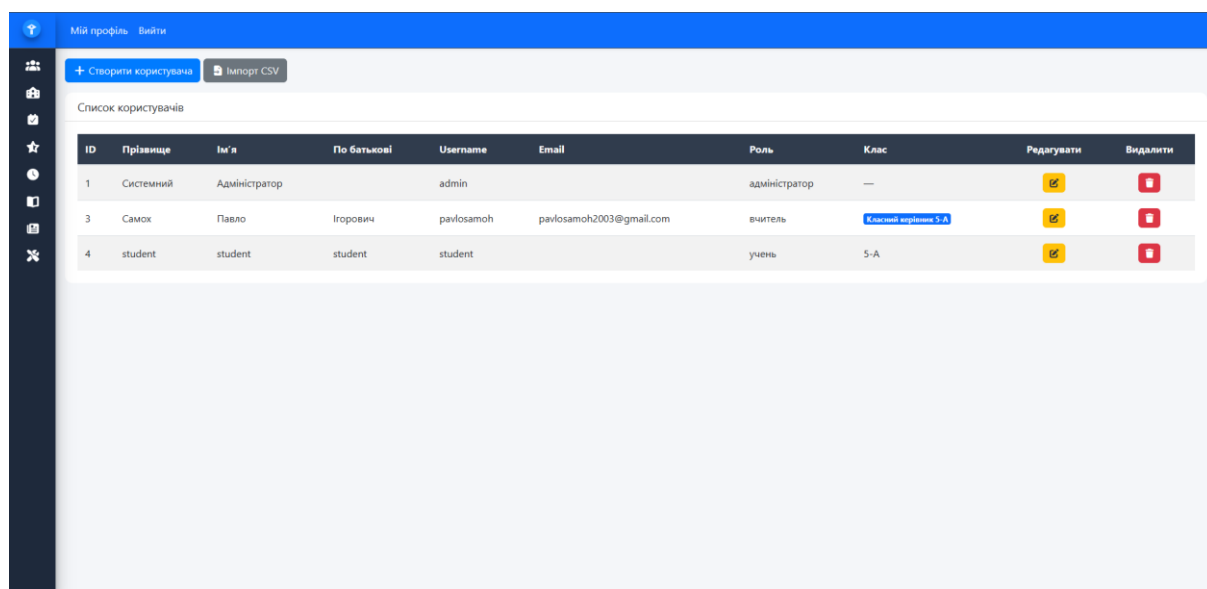
4. Сценарії роботи за ролями

4.1 Робота адміністратора

Адміністратор відповідає за наповнення системи довідниками та користувачами, а також за організаційні дані (класи, розклад, новини).

4.1.1 Управління користувачами

У боковому меню адміністратор обирає пункт «Користувачі». Відкривається сторінка зі списком користувачів (рис. Б.6), де доступні: перегляд, створення, редагування, видалення і імпорт в форматі csv.



ID	Прізвище	Ім'я	По батькові	Username	Email	Роль	Клас	Редагувати	Видалити
1	Системний	Адміністратор		admin		адміністратор	—		
3	Самох	Павло	Ігорович	pavlosamoh	pavlosamoh2003@gmail.com	вчитель	Класний керівник 5-А		
4	student	student	student	student		учень	5-А		

Рисунок Б.6 – Список користувачів

Для створення нового користувача адміністратор натискає «Додати», після чого відкривається форма (модальне вікно) створення (рис. Б.7). У формі заповнюються основні поля (ПІБ, email, пароль), а також обирається роль.

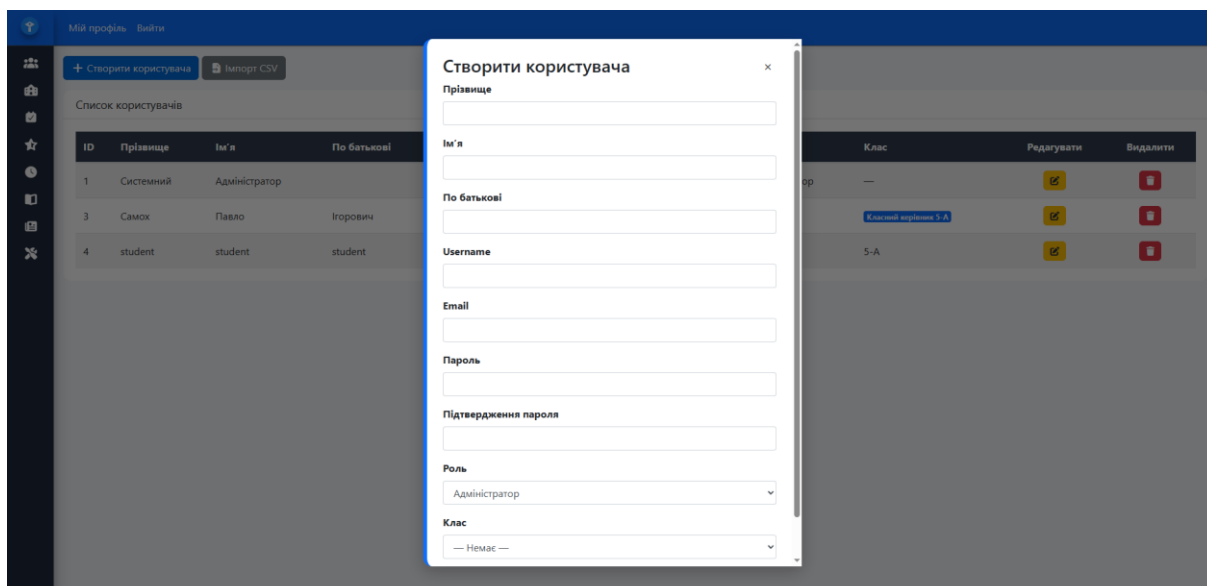


Рисунок Б.7 – Створення користувача

Після натискання «Зберегти» новий користувач з'являється у списку (рис. Б.6) та може виконати вхід за своїми даними.

Якщо для учня потрібно вказати прив'язку до класу (за логікою системи), адміністратор обирає відповідне значення у формі створення користувача (рис. Б.7).

4.1.2 Управління класами

Для створення/редагування класів адміністратор переходить у розділ «Класи». На сторінці відображається перелік класів та дії над ними (рис. Б.8).

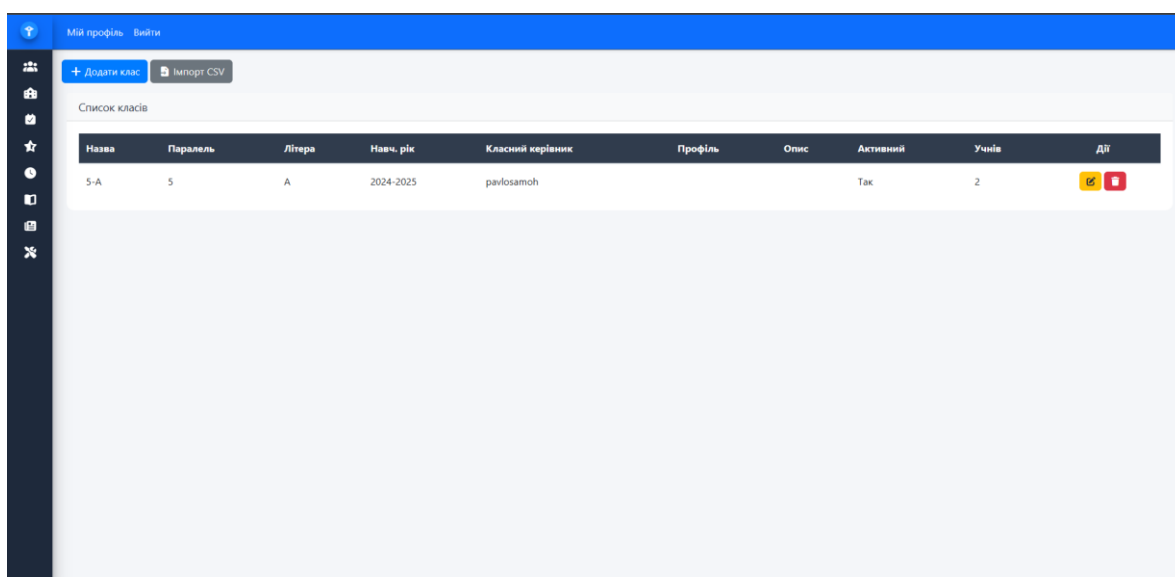


Рисунок Б.8 – Список класів

Під час створення класу (рис. Б.9) адміністратор задає параметри (наприклад, номер/літеру класу), а також може призначити класного керівника, якщо це передбачено інтерфейсом.

The screenshot shows a web application interface with a dark sidebar on the left containing icons for user profile, classes, subjects, and reports. The main content area is titled 'Мій профіль Вийти' and features a '+ Додати клас' button and an 'Імпорт CSV' button. Below these is a table titled 'Список класів' with columns: Назва, Паралель, Літера, Навч. рік. A modal window titled 'Додати клас' is open in the center, containing the following fields: 'Паралель' (text input), 'Літера' (text input), 'Назва (автоматично)' (text input), 'Навчальний рік' (text input with value '2024-2025'), 'Класний керівник' (dropdown menu with 'Немає' selected), 'Профіль' (text input), 'Опис' (text area), and 'Активний' (dropdown menu with 'Так' selected). A 'Створити' button is at the bottom of the modal.

Рисунок Б.9 – Створення класу

4.1.3 «Курси»

Розділ меню «Курси» відкриває сторінку керування курсами/предметами (рис. Б.10). Тут адміністратор може переглядати створені предмети та контролювати загальну структуру навчальних даних.

The screenshot shows the 'Курси' (Courses) management page. The sidebar is the same as in the previous image. The main content area has a blue header with 'Мій профіль Вийти'. Below the header is a section titled 'Курси' with a search bar 'Пошук за назвою курсу' and a 'Пошук' button. There is a '+ Створити курс' button. Below this is a 'Мої курси' section containing a card for 'Математика 7-Б' (Mathematics 7-B). The card displays 'Предмет: Математика', 'Клас: 5-А', and 'Викладач: Савен Павло'. At the bottom of the card are three buttons: 'Керувати' (Manage), 'Аналітика' (Analytics), and a red button with a minus sign.

Рисунок Б.10 – Розділ «Курси»

4.1.4 Розклад

Для перегляду або формування розкладу адміністратор переходить у «Розклад». На сторінці розкладу доступний вибір параметрів (наприклад, дата/клас), а також перегляд сформованих записів (рис. Б.11).

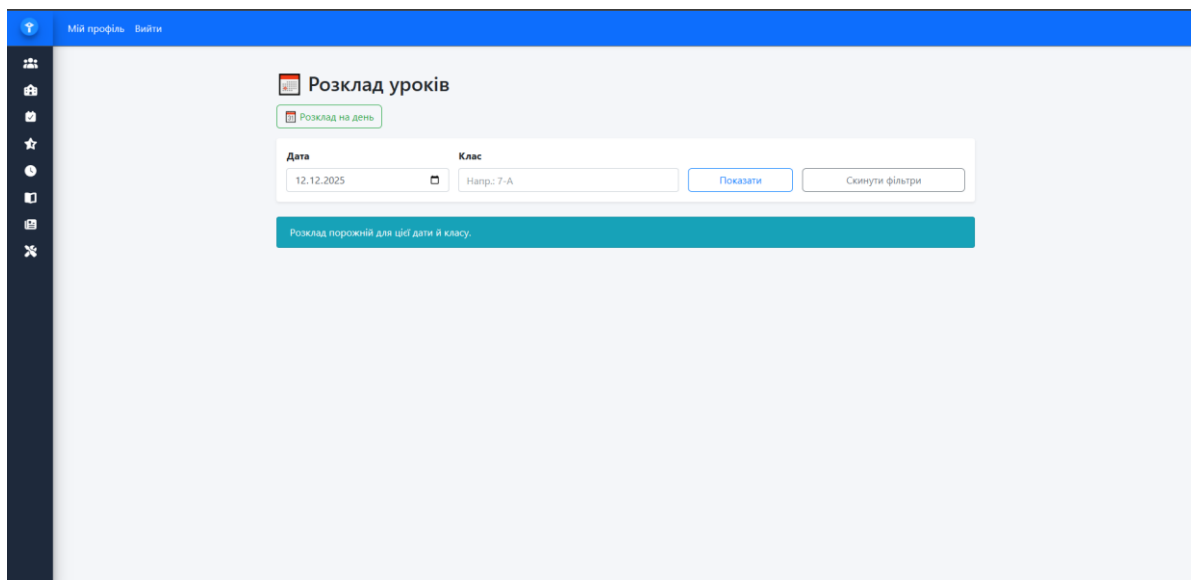


Рисунок Б.11 – Перегляд розкладу

Якщо в системі використовується додавання розкладу «на день», адміністратор відкриває відповідну форму та задає дату, клас і перелік уроків (рис. Б.12).

Початок	Кінець	Предмет	Вчитель
09:00	09:45	Алгебра	Іваненко І.М.
10:00	10:45	Історія України	Марченко М.В.
11:00	11:45	Англійська мова	Карп'юк П.П.
12:00	12:45	Геометрія	Самбір А.М.
--:--	--:--	Предмет	Вчитель
--:--	--:--	Предмет	Вчитель

Рисунок Б.12 – Додавання розкладу на день

4.1.5 Новини

Для інформування користувачів адміністратор може публікувати новини. У меню обирається «Керування новинами» (залежно від реалізації доступів). На рисунку Б.13 наведено приклад списку новин та переходу до створення.

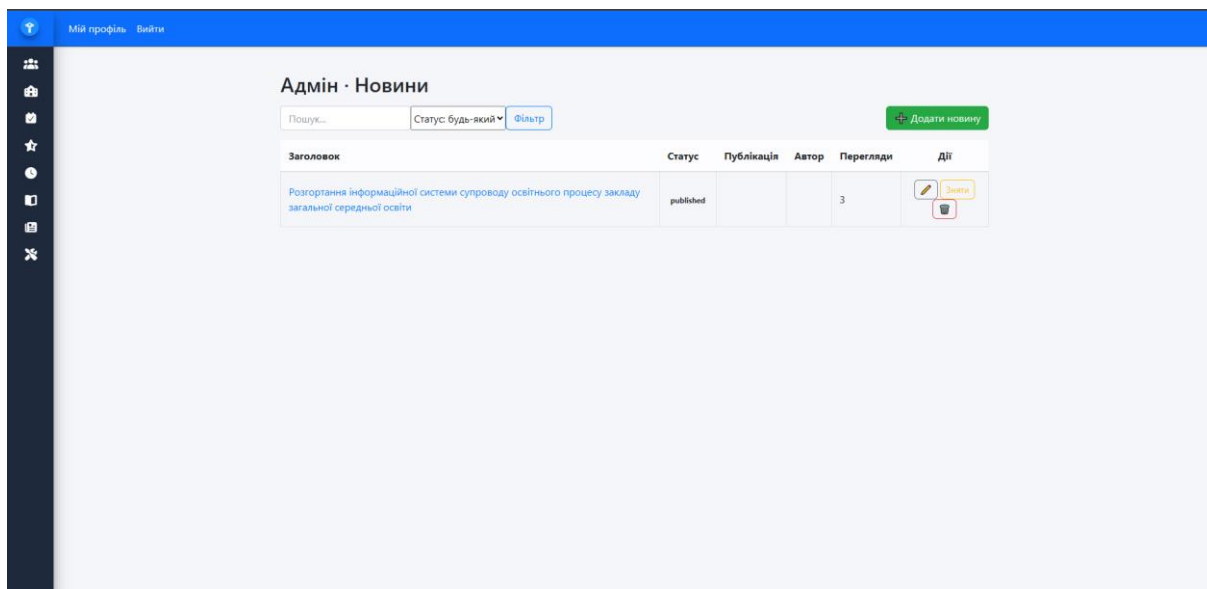


Рисунок Б.13 – Керування новинами

Створення новини виконується через форму введення заголовка та тексту (рис. Б.14), після чого новина стає доступною іншим користувачам у розділі НОВИН.

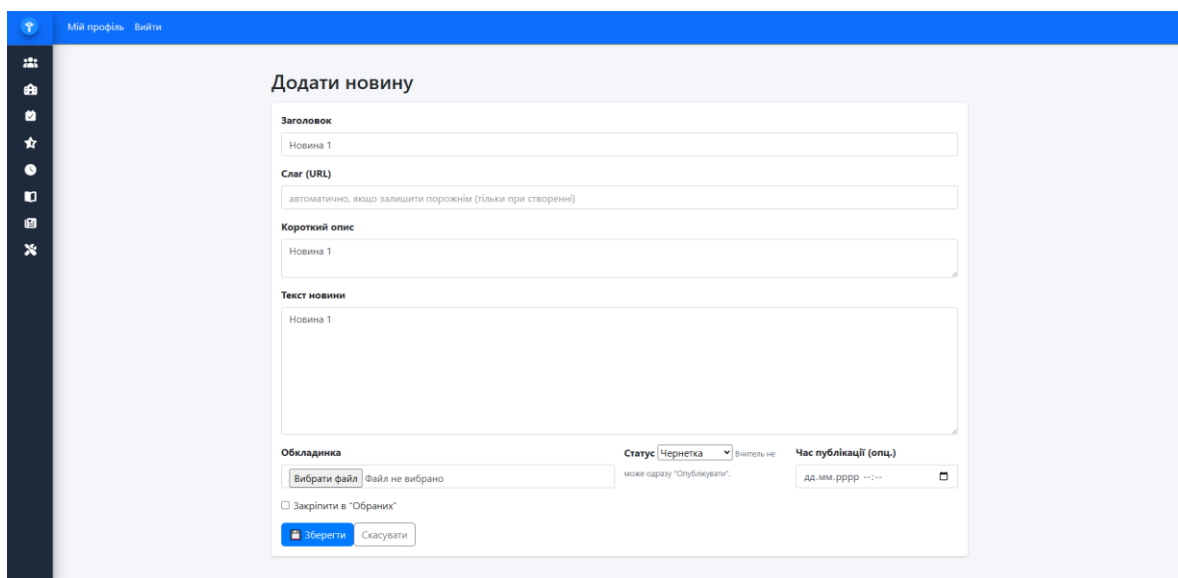


Рисунок Б.14 – Створення новини

4.2 Робота вчителя

Роль вчителя передбачає роботу з навчальними матеріалами (предметами/курсами), завданнями, оцінюванням та відвідуваністю.

4.2.1 Перегляд предметів і наповнення курсу

Вчитель відкриває «Курси» (рис. Б.10) та переходить у конкретний курс. Для наповнення курсу (теми, матеріали, завдання) використовується сторінка керування курсом (рис. Б.15).

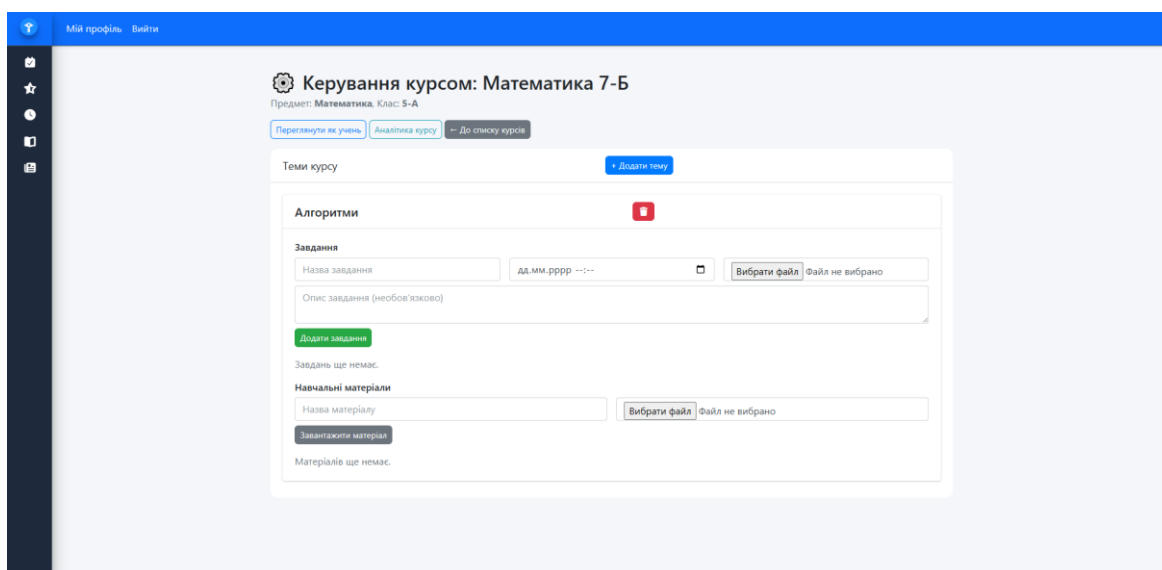


Рисунок Б.15 – Керування курсом вчителем

Логіка роботи зазвичай така:

створити тему (назва, опис/порядок);

додати матеріали (файли/посилання, якщо передбачено);

створити завдання (умова, дедлайн, прикріплення файлів, якщо передбачено).

4.2.2 Робота із завданнями та перевірка відповідей

Після публікації завдання учні надсилають свої роботи. Вчитель відкриває сторінку завдання та переглядає список відповідей/подань (рис. Б.16), перевіряє матеріали та виставляє оцінку.

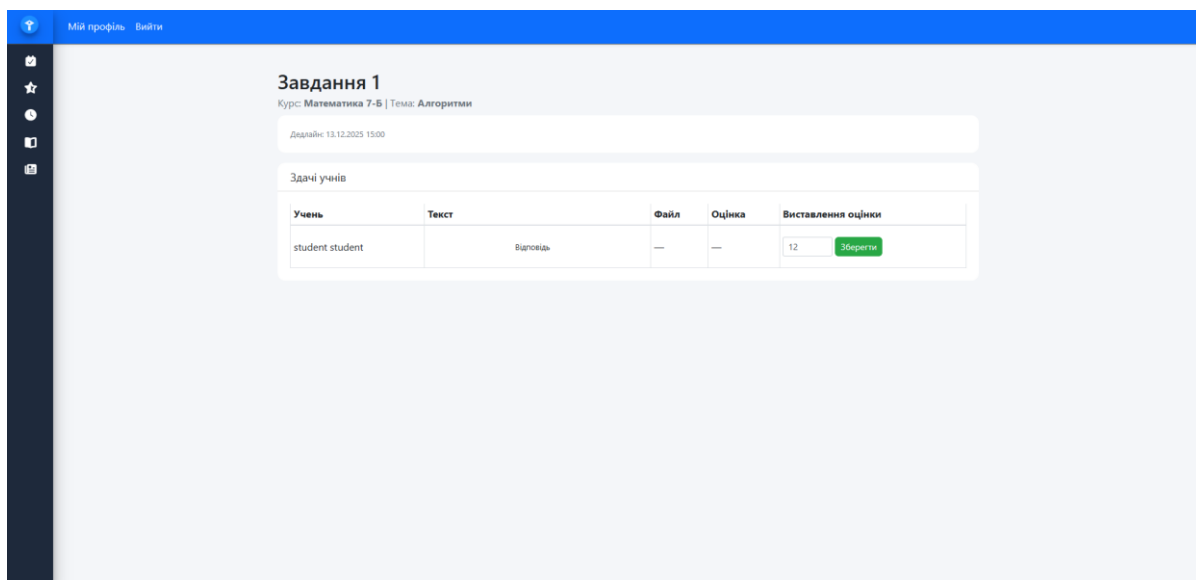


Рисунок Б.16 – Перегляд подань/відповідей на завдання

4.2.3 Відвідуваність та оцінки

У меню вчителя доступні розділи:

- «Відвідуваність» (фіксація присутності, перегляд записів);
- «Оцінки» (перегляд/аналіз оцінок, фільтрація).

Приклад сторінки відвідуваності показано на рис. Б.17, а сторінки оцінок – на рис. Б.18.

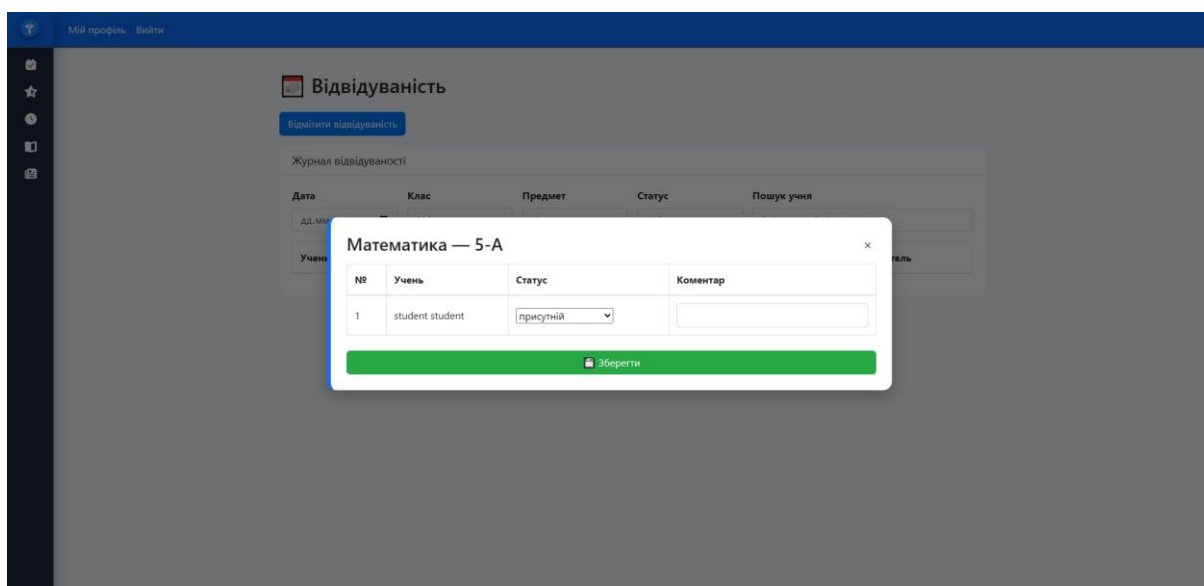


Рисунок Б.17 – Відвідуваність

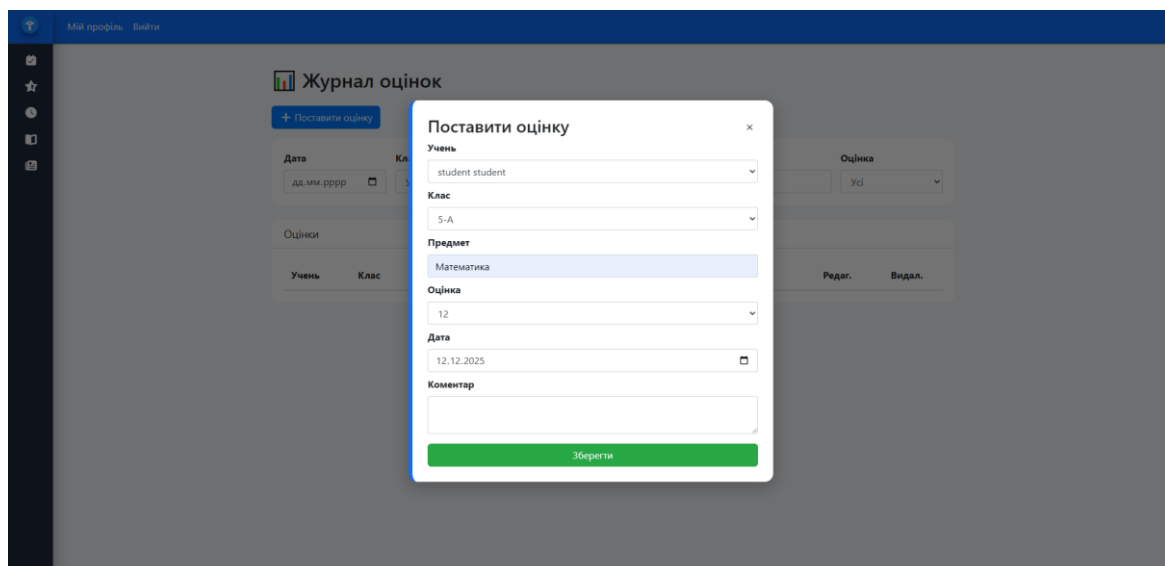


Рисунок Б.18 – Оцінки

4.3 Робота учня

Учень використовує систему для перегляду розкладу, доступу до матеріалів, виконання домашніх завдань, а також контролю оцінок і відвідуваності.

4.3.1 Навчальні матеріали

Для переходу до матеріалів учень відкриває «Предмети» (рис. Б.10) і обирає потрібний курс. На сторінці курсу (рис. Б.19) відображаються теми та пов'язані ресурси.

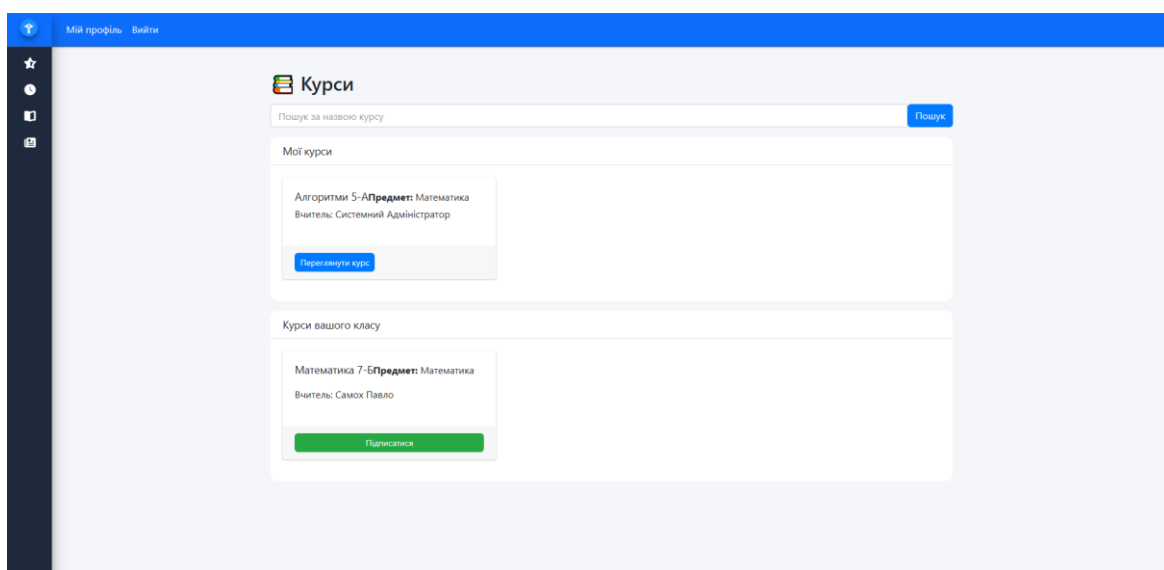


Рисунок Б.19 – Перегляд курсу учнем

4.3.2 Виконання домашнього завдання

Учень переходить у потрібне завдання зі сторінки курсу (рис. Б.19). На сторінці завдання відображається умова та форма подання відповіді (рис. Б.20). Після прикріплення файлу або введення відповіді учень надсилає роботу.

Завдання 1

Курс: Математика 7-8 | Тема: Алгоритми

Дедлайн: 13.12.2025 15:00

Моя відповідь

Текстова відповідь

Відповідь

Файл (необов'язково, перезавантажує старий)

Вибрати файл Файл не вибрано

Зберегти відповідь

Рисунок Б.20 – Сторінка завдання та подання відповіді

Після відправлення робота фіксується в системі, а статус/оцінка стають доступними після перевірки вчителем.

4.3.3 Перегляд оцінок і відвідуваності

Для контролю успішності учень відкриває «Оцінки» (рис. Б.21).

Журнал оцінок

Оцінки

Предмет	Дата	Оцінка	Коментар	Вчитель
Математика	2025-12-12	12		Самох Павло

Рисунок Б.21 – Перегляд оцінок за предметами/періодами

Для перегляду своєї відвідуваності учень повинен відкрити «Моя відвідуваність» (рис. Б.22).

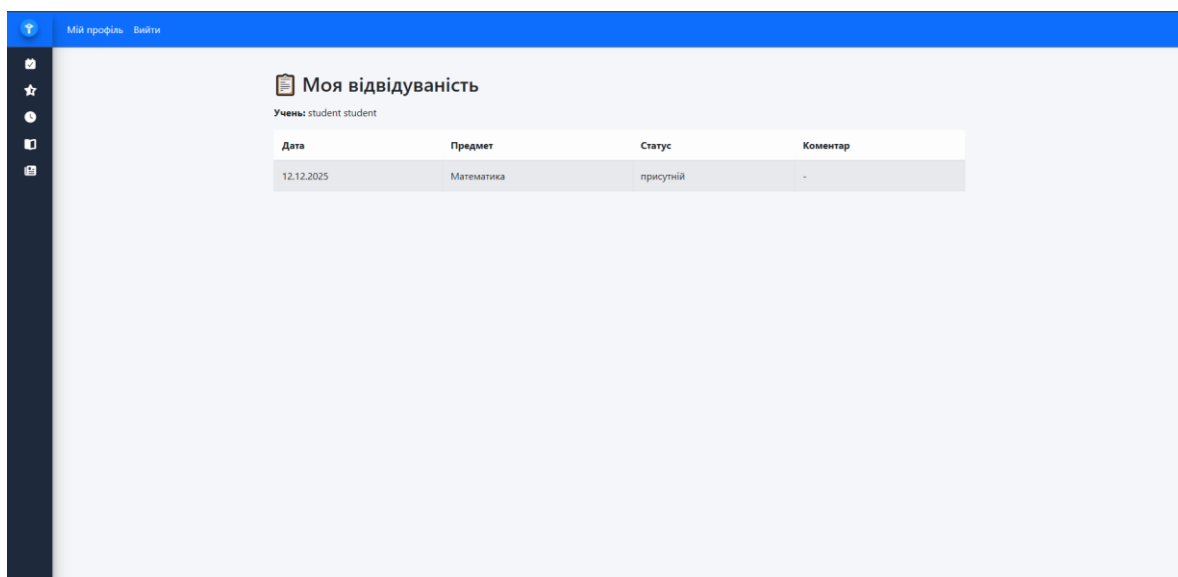


Рисунок Б.22 – Перегляд відвідуваності учнем

5. Типові ситуації та рекомендації

Не вдається увійти (помилка email/пароля). Перевірити правильність введення даних (без зайвих пробілів). Якщо пароль втрачено – звернутися до адміністратора для скидання/оновлення.

Не вдається прикріпити файл до завдання. Перевірити формат/розмір файлу. За потреби спробувати інший браузер або повторити дію після оновлення сторінки.

Дані не оновилися одразу. Оновити сторінку (F5) або повторно відкрити розділ через бокове меню.

ДОДАТОК В

Фрагменти програмного коду інформаційної системи

В.1 Реалізація формування даних аналітики

```

public function analytics(Course $course)
{
    $this->ensureCourseOwner($course);

    $course->load(['assignments.submissions.student', 'students']);

    $assignments = $course->assignments()->with('submissions')->get();

    $assignmentStats = $assignments->map(function (Assignment $a) {
        $grades = $a->submissions->pluck('grade')->filter();
        return [
            'id'      => $a->id,
            'title'   => $a->title,
            'submissions' => $a->submissions->count(),
            'graded_count' => $grades->count(),
            'avg_grade'  => $grades->count() ? round($grades->avg(), 2) : null,
            'max_grade'  => $grades->count() ? $grades->max() : null,
            'min_grade'  => $grades->count() ? $grades->min() : null,
        ];
    });

    $students = $course->students()->get();

    $perStudentOnline = [];
    foreach ($students as $s) {
        $subs = Submission::whereIn('assignment_id', $assignments->pluck('id'))
            ->where('student_id', $s->id)
            ->get();

        $grades = $subs->pluck('grade')->filter();
    }
}

```

```

$perStudentOnline[$s->id] = [
    'student'      => $s,
    'submissions'  => $subs->count(),
    'graded_count' => $grades->count(),
    'avg_online'   => $grades->count() ? round($grades->avg(), 2) : null,
    'missing_count' => max(0, $assignments->count() - $subs->count()),
];
}

```

```

$offlineQuery = Grade::query()
->where('class_id', $course->class_id)
->where('subject_name', $course->subject_name);

```

```

$offlineByStudent = $offlineQuery->get()->groupBy('student_id')->map(function
($items) {
    return [
        'count'      => $items->count(),
        'avg_grade' => round($items->avg('value'), 2),
    ];
});

```

```

$attendanceQuery = Attendance::query()
->where('class_id', $course->class_id)
->where('subject_name', $course->subject_name);

```

```

$attendanceByStudent = $attendanceQuery->get()->groupBy('student_id')-
>map(function ($items) {
    $absent = $items->whereIn('status', ['відсутній', 'поважна причина'])->count();
    return [
        'total'      => $items->count(),
        'absent'     => $absent,
        'absence_rate' => $items->count()
            ? round($absent / $items->count() * 100, 1)
            : null,
    ];
});

```

```

    ];
  });

  $chartAssignmentsLabels = $assignmentStats->pluck('title');
  $chartAssignmentsAvg    = $assignmentStats->pluck('avg_grade')->map(fn($v) => $v
  ?? 0);

  $chartStudentsLabels = $students->map(fn($s) => $s->last_name . ' ' . $s->first_name);
  $chartStudentsAvg    = $students->map(function ($s) use ($perStudentOnline) {
    return $perStudentOnline[$s->id]['avg_online'] ?? 0;
  });

  return view('courses.analytics', compact(
    'course',
    'assignmentStats',
    'students',
    'perStudentOnline',
    'offlineByStudent',
    'attendanceByStudent',
    'chartAssignmentsLabels',
    'chartAssignmentsAvg',
    'chartStudentsLabels',
    'chartStudentsAvg',
  ));
}

```

B.2 Реалізація імпорту класів із CSV-файлу

```

public function importCSV(Request $request)
{
  $request->validate([
    'csv' => 'required|file|mimes:csv,txt'
  ]);

  $file = fopen($request->file('csv')->getRealPath(), 'r');

```

```

$header = fgetcsv($file);

$required = [
    'grade', 'letter', 'academic_year',
    'profile', 'description', 'active',
    'class_teacher_username'
];

if ($header !== $required) {
    return back()->with('csv_errors', [
        'Невірний формат CSV. Має бути: ' . implode(', ', $required)
    ]);
}

$errors = [];
$imported = 0;
$rowNumber = 1;

while (($row = fgetcsv($file)) !== false) {
    $rowNumber++;

    $data = array_combine($required, $row);

    $name = "{${data['grade']}}-".mb_strtoupper(${data['letter']});

    $teacherId = null;
    if (!empty(${data['class_teacher_username']})) {
        $teacher = User::where('username', ${data['class_teacher_username']})->first();

        if (!$teacher) {
            $errors[] = "Рядок {$rowNumber}: Вчитель '{${data['class_teacher_username']}'
не знайдений";
            continue;
        }
    }
}

```

```

        if ($teacher->role !== 'вчитель') {
            $errors[] = "Рядок {$rowNumber}: '{$data['class_teacher_username']}' не є
вчителем";

            continue;
        }

        $teacherId = $teacher->id;
    }

    try {
        SchoolClass::create([
            'grade' => (int)$data['grade'],
            'letter' => $data['letter'],
            'name' => $name,
            'academic_year' => $data['academic_year'],
            'profile' => $data['profile'] ?: null,
            'description' => $data['description'] ?: null,
            'active' => (int)$data['active'],
            'class_teacher_id' => $teacherId,
        ]);

        $imported++;

    } catch (\Exception $e) {
        $errors[] = "Рядок {$rowNumber}: помилка ({$e->getMessage()})";
    }
}

fclose($file);

return back()
    ->with('success', "Імпортовано {$imported} класів")
    ->with('csv_errors', $errors);
}

```

B.3 Реалізація керування розкладом занять на рівні дня

```

public function day(Request $request)
{
    $date = $request->get('date', now()->toDateString());
    $class = $request->get('class_name', "");

    $query = Schedule::whereDate('date', $date)->orderBy('start_time');
    if ($class !== "") {
        $query->where('class_name', $class);
    }
    $schedules = $query->get();

    $classes = Schedule::select('class_name')->distinct()->orderBy('class_name')-
    >pluck('class_name');

    return view('schedules.day', compact('date', 'class', 'schedules', 'classes'));
}

public function dayStore(Request $request)
{
    $request->validate([
        'date' => 'required|date',
        'class_name' => 'required|string|max:50',
        'lessons' => 'required|array|min:1',
    ]);

    $date = $request->date;
    $className = $request->class_name;

    $lessons = collect($request->lessons)
        ->filter(fn($l) =>
            !empty($l['start_time']) &&
            !empty($l['subject_name']) &&

```

```

        !empty($l['teacher_name'])
    )
    ->map(fn($l) => [
        'date' => $date,
        'start_time' => $l['start_time'],
        'end_time' => $l['end_time'] ?? null,
        'subject_name' => $l['subject_name'],
        'teacher_name' => $l['teacher_name'],
        'class_name' => $className,
        'created_at' => now(),
        'updated_at' => now(),
    ])
    ->values();

    if ($lessons->isEmpty()) {
        return back()->withErrors(['msg' => 'Не вказано жодного повного уроку!'])-
>withInput();
    }
    foreach ($lessons as $l) {
        $start = $l['start_time'];
        $end = $l['end_time'] ?? $l['start_time'];

        $conflict = Schedule::whereDate('date', $date)
            ->where(function($q) use ($l, $className) {
                $q->where('class_name', $className)
                ->orWhere('teacher_name', $l['teacher_name']);
            })
            ->where(function($q) use ($start, $end) {
                $q->whereBetween('start_time', [$start, $end])
                ->orWhereBetween('end_time', [$start, $end])
                ->orWhere(function($qq) use ($start, $end) {
                    $qq->where('start_time', '<=', $start)
                    ->where('end_time', '>=', $end);
                });
            });
    }
}

```

```

->exists();

if ($conflict) {
    return back()->withErrors([
        'msg' => "Конфлікт: {$l['teacher_name']} або клас {$className} вже зайняті
{$start}-{$l['end_time']}.",
    ])->withInput();
}
}

Schedule::insert($lessons->all());

return redirect()->route('schedules.day', [
    'date' => $date,
    'class_name' => $className
])->with('success', 'Розклад збережено!');
}

```

UML-діаграми інформаційної системи

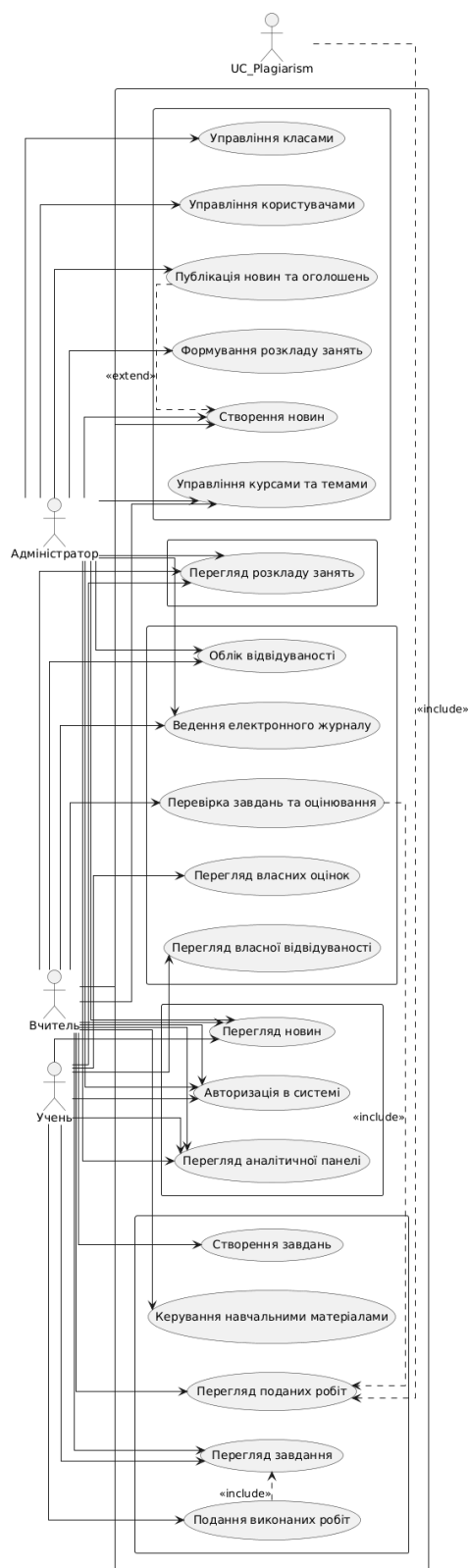


Рисунок Г.1 – Діаграма варіантів використання інформаційної системи супроводу освітнього процесу ЗЗСО

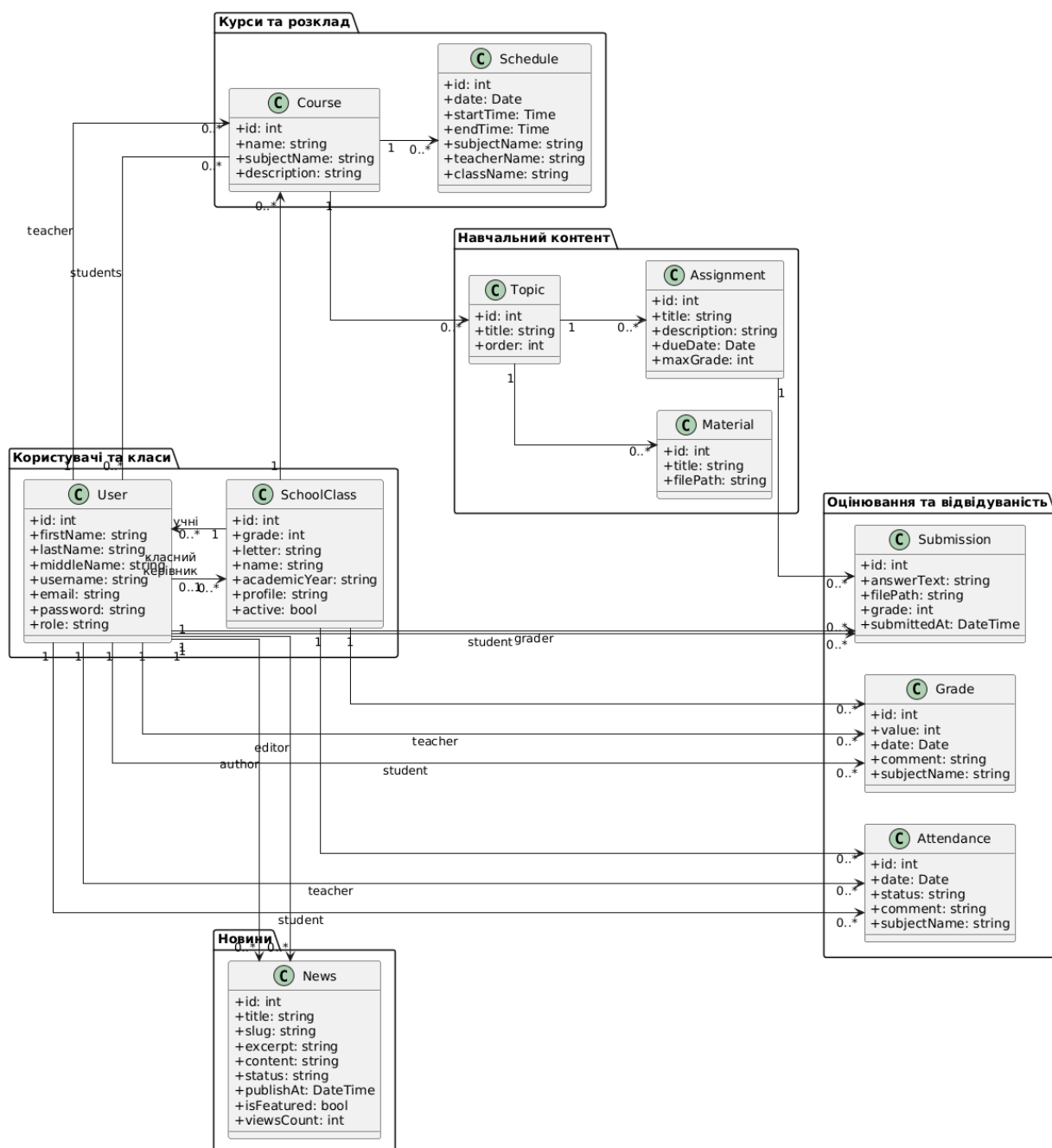


Рисунок Г.2 – Діаграма класів інформаційної системи супроводу освітнього процесу ЗЗСО

АНОТАЦІЯ

Самох П.І. – Проектування та розробка інформаційної системи супроводу освітнього процесу закладу загальної середньої освіти – Рукопис.

Кваліфікаційна робота за спеціальністю 122 Комп'ютерні науки. – Волинський національний університет імені Лесі Українки, Луцьк. – 2025 р.

Кваліфікаційна робота присвячена проектуванню та розробці інформаційної системи супроводу освітнього процесу закладу загальної середньої освіти. У роботі здійснено аналіз сучасних підходів до цифровізації освітнього середовища, розглянуто особливості побудови веборієнтованих інформаційних систем та їх роль у підвищенні ефективності управління навчальним процесом.

У межах дослідження проаналізовано нормативні та методичні вимоги до організації освітнього процесу в закладах загальної середньої освіти, а також існуючі програмні рішення, що застосовуються для автоматизації навчальної діяльності. На основі проведеного аналізу сформовано вимоги до функціональних можливостей інформаційної системи, зокрема щодо управління користувачами, навчальними класами, предметами, розкладом занять, навчальними матеріалами, завданнями, оцінками та відвідуваністю.

Особливу увагу приділено процесу проектування архітектури програмного засобу, логіці взаємодії між компонентами системи та реалізації рольової моделі доступу, яка забезпечує розмежування функціональних можливостей між адміністраторами, вчителями та учнями. Реалізована система дозволяє централізовано зберігати та опрацьовувати навчальну інформацію, забезпечуючи зручний доступ до неї через вебінтерфейс.

У роботі описано етапи програмної реалізації інформаційної системи з використанням сучасних вебтехнологій та фреймворків, а також розглянуто питання тестування та налагодження програмного засобу. Проведене тестування підтвердило коректність роботи основних функціональних модулів і стабільність системи під час взаємодії з користувачами.

Результатом виконання кваліфікаційної роботи є повнофункціональна інформаційна система, яка може бути використана як основа для впровадження цифрових рішень у закладах загальної середньої освіти з метою підвищення ефективності організації та супроводу освітнього процесу.

Ключові слова: інформаційна система, освітній процес, заклад загальної середньої освіти, вебзастосунок, автоматизація, управління навчальними даними, Laravel, PHP, HTML, CSS, JavaScript.

ABSTRACT

Samokh P. I. – **Design and development of an information system for supporting the educational process of a general secondary education institution** – Manuscript.

Qualification thesis in the speciality 122 Computer Sciences. – Lesya Ukrainka Volyn National University, Lutsk. – 2025.

The qualification thesis is devoted to the design and development of an information system for supporting the educational process of a general secondary education institution. The paper analyzes modern approaches to the digitalization of the educational environment and examines the features of building web-oriented information systems and their role in improving the efficiency of educational process management.

The study analyzes regulatory and methodological requirements for organizing the educational process in general secondary education institutions, as well as existing software solutions used to automate educational activities. Based on the analysis, functional requirements for the information system were formulated, including user management, class and subject management, scheduling, learning materials, assignments, grades, and attendance tracking.

Special attention is paid to the system architecture design, the logic of interaction between system components, and the implementation of a role-based access model that ensures differentiation of functionality between administrators, teachers, and students.

The developed system provides centralized storage and processing of educational data with convenient access through a web-based interface.

The thesis describes the stages of software implementation using modern web technologies and frameworks, as well as testing and debugging procedures. The conducted testing confirmed the correctness of the main functional modules and the stability of the system during user interaction.

The result of the qualification work is a fully functional information system that can serve as a basis for implementing digital solutions in general secondary education institutions to enhance the organization and support of the educational process.

Keywords: information system, educational process, general secondary education institution, web application, automation, educational data management, Laravel, PHP, HTML, CSS, JavaScript.