

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Програмний модуль для автоматичного генерування
навчальної відомості студентів на основі даних з електронного
журналу успішності»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

	Анжеліка ГАВРИЛЮК
(підпис)	Ім'я, ПРІЗВИЩЕ здобувача

Виконав:	здобувачка вищої освіти гр. ІСД-43
	Анжеліка ГАВРИЛЮК
	Ім'я, ПРІЗВИЩЕ

Керівник:	
науковий ступінь,	к. т. н. Андрій АРОНОВ
вчене звання	Ім'я, ПРІЗВИЩЕ

Рецензент:	
науковий ступінь,	
вчене звання	Ім'я, ПРІЗВИЩЕ

Київ 2025

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій
Ступінь вищої освіти бакалавр
Спеціальність 126 Інформаційні системи та технології
Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедрою ІСТ

_____ Каміла СТОРЧАК

« ____ » _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Гаврилюк Анжеліка Русланівна

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Програмний модуль для автоматичного генерування навчальної відомості студентів на основі даних з електронного журналу успішності

керівник кваліфікаційної роботи Андрій АРОНОВ, к. т. н.,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій
від «24» лютого 2025р. № 56

2. Строк подання кваліфікаційної роботи «30» травня 2025 р.
3. Вихідні дані до кваліфікаційної роботи: офіційна технічна документація технологій та інструментів розробки, відкриті дані наукових досліджень із діджиталізації навчального процесу, відкриті дані українських та міжнародних навчальних закладів про архітектуру платформ електронного обліку навчальної успішності.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Огляд технологій та існуючих рішень електронного обліку успішності студентів
2. Аналіз ефективності існуючої системи обліку успішності
3. Розробка модулю автоматичної генерації навчальної відомості
5. Перелік ілюстративного матеріалу: *презентація*
6. Дата видачі завдання «24» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	01.03.2025 р.	
2	Дослідження існуючих систем електронного обліку навчальної успішності у закладах освіти	15.03.2025 р.	
3	Аналіз ефективності існуючої системи обліку успішності	17.03.2025 р.	
4	Формулювання технічного завдання, функціональних та нефункціональних вимог на розробку модулю генерації навчальних відомостей	04.04.2025 р.	
5	Реалізація та тестування розробленого модулю	16.05.2025 р.	
6	Розробка демонстраційних матеріалів	26.05.2025 р.	
7	Попередній захист дипломної роботи	27.05.2025 р.	
8	Подання роботи в деканат	30.05.2025 р.	

Здобувач(ка) вищої освіти

(підпис)

Анжеліка ГАВРИЛЮК

(Ім'я, ПРІЗВИЩЕ)

Керівник

кваліфікаційної роботи

(підпис)

Андрій АРОНОВ

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 73 стор., 37 рис., 11 табл., 30 джерел.

Мета роботи – розробка програмного модуля для автоматичного формування навчальної відомості студента на основі даних з електронного журналу успішності з метою підвищення ефективності академічного документообігу, зменшення адміністративного навантаження та забезпечення точності й доступності звітності в цифровому середовищі закладу вищої освіти.

Об'єкт дослідження – інформаційна інфраструктура автоматизованого обліку успішності студентів у закладах вищої освіти, включаючи електронні журнали, інтерфейси доступу до освітніх даних, а також програмні компоненти, що забезпечують формування звітної документації.

Предмет дослідження – технології та програмні засоби автоматичного генерування навчальних відомостей на основі структурованих даних, зокрема JSON-формату, у рамках інтеграції з системами електронного журналу.

Короткий зміст роботи: у роботі досліджено структуру електронного журналу успішності, механізми представлення та обробки даних про результати навчання студентів. Проаналізовано наявні системи обліку успішності та виявлено обмеження щодо формування навчальних відомостей. Розроблено архітектуру серверного модуля мовою програмування Go з використанням вебфреймворку Fiber та бібліотеки Excelize для створення звітів. Реалізовано HTML-інтерфейс для інтеграції з особистим кабінетом студента. Проведено тестування функціональності модуля та обґрунтовано доцільність його впровадження в освітній процес.

КЛЮЧОВІ СЛОВА: АВТОМАТИЗАЦІЯ ДОКУМЕНТООБІГУ, ЕЛЕКТРОННИЙ ЖУРНАЛ УСПІШНОСТІ, НАВЧАЛЬНА ВІДОМІСТЬ, МОДУЛЬ ГЕНЕРАЦІЇ, ІНФОРМАЦІЙНА СИСТЕМА, GO, FIBER, EXCELIZE, ВЕБ-ІНТЕРФЕЙС.

ABSTRACT

Text part of the qualification work for the bachelor's degree: 73 pages, 37 pictures, 11 tables, 30 sources.

The purpose of the study is to develop a software module for the automatic generation of a student's academic record based on data from an electronic gradebook in order to increase the efficiency of academic workflow, reduce administrative burden and ensure the accuracy and availability of reporting in the digital environment of a higher education institution.

Object of research – information infrastructure for automated accounting of student performance in higher education institutions, including electronic journals, interfaces for accessing educational data, as well as software components that ensure the generation of reporting documentation.

The subject of the study is technologies and software tools for automatic generation of educational information based on structured data, in particular JSON format, within the framework of integration with electronic journal systems.

Summary of the work: The paper investigates the structure of the electronic gradebook, mechanisms for presenting and processing data on student learning outcomes. The existing systems of academic performance accounting are analyzed and limitations in the formation of academic data are identified. The architecture of the server module was developed in the Go programming language using the Fiber web framework and the Excelize library for generating reports. An HTML interface for integration with the student's personal account was implemented. The functionality of the module was tested and the feasibility of its implementation in the educational process was substantiated.

KEYWORDS: AUTOMATION OF DOCUMENT FLOW, ELECTRONIC JOURNAL OF ACADEMIC PERFORMANCE, STUDY SHEET, GENERATION MODULE, INFORMATION SYSTEM, GO, FIBER, EXCELIZE, WEB INTERFACE.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 ТЕХНОЛОГІЧНІ ОСОБЛИВОСТІ АВТОМАТИЗАЦІЇ ОБЛІКУ УСПІШНОСТІ СТУДЕНТІВ.....	13
1.1 Огляд сучасних систем електронного журналу успішності та їх можливостей щодо формування навчальної відомості.....	13
1.2 Методологія та досвід автоматизації документообігу в навчальних закладах України.....	19
1.3. Програмні методи генерації електронних документів обліку	23
РОЗДІЛ 2 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ІСНУЮЧИХ РІШЕНЬ АВТОМАТИЗАЦІЇ ОБЛІКУ УСПІШНОСТІ У ВИЩИХ НАВЧАЛЬНИХ ЗАКЛАДАХ	28
2.1 Огляд та аналіз недоліків популярних рішень електронного журналу в контексті формування навчальних відомостей.....	28
2.2 Дослідження потреб викладачів, адміністрації та студентів щодо створення та використання навчальної відомості.....	36
2.3 Технічні вимоги до розроблюваного модуля автоматичного генерування відомостей.....	44
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ МОДУЛЮ ГЕНЕРУВАННЯ НАВЧАЛЬНОЇ ВІДОМОСТІ.....	52
3.1 Проєктування архітектури системи	52
3.2 Проєктування користувацького інтерфейсу системи	58
3.3 Реалізація програмних модулів генерування відомостей	62
3.4 Тестування розробленого модуля та перспективи впровадження.....	66
ВИСНОВКИ.....	72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	74

ВСТУП

У сучасних умовах цифрової трансформації освіти, що охоплює всі рівні навчального процесу – від організації електронного навчання до автоматизації академічного документообігу – особливого значення набуває питання ефективного управління освітніми даними. Перехід до електронного формату звітності, включаючи автоматичне формування навчальних відомостей, не лише відповідає загальносвітовим тенденціям paperless-документообігу, а й сприяє підвищенню прозорості, точності та оперативності в управлінні академічною інформацією. З огляду на зростання обсягів освітніх даних, а також потребу в індивідуалізації навчального процесу, питання автоматизованої генерації персоніфікованих документів стає надзвичайно актуальним.

Актуальність дослідження зумовлена низкою проблем, що характерні для сучасного стану академічного обліку в закладах вищої освіти України. Серед них – ручне або напівавтоматизоване формування навчальних відомостей, яке потребує значних часових і кадрових ресурсів; відсутність уніфікованих механізмів генерації офіційних звітів про успішність студентів; недостатній рівень інтеграції між інформаційними системами (електронний журнал, електронний деканат, системи аналітики).

Аналіз наявних програмних рішень – зокрема, Moodle, Google Classroom, локальні розробки ЗВО (наприклад, КПП «Електронний кампус», ПС «Журнал успішності» ЛНУ імені І. Франка) – засвідчує відсутність універсального рішення для автоматичного формування навчальних відомостей на основі даних електронного журналу. Більшість із наявних систем не підтримують вивантаження персоналізованих звітів у стандартному форматі, що відповідає вимогам вітчизняного електронного документообігу.

У контексті потреб вітчизняної освітньої системи та з урахуванням рекомендацій Міністерства освіти і науки України щодо впровадження цифрових рішень, запропонована тема кваліфікаційної роботи є актуальною, своєчасною та такою, що має значну соціальну й галузеву значущість.

Метою даної кваліфікаційної роботи є розробка програмного модуля автоматичного формування навчальної відомості студента на основі даних електронного журналу успішності, з урахуванням вимог до надійності, масштабованості, конфіденційності даних та сумісності з інформаційною інфраструктурою закладу вищої освіти.

Досягнення поставленої мети передбачає розв'язання таких завдань:

- здійснити критичний аналіз сучасних систем обліку успішності та виявити їх обмеження;
- дослідити програмні підходи до автоматизації формування електронних звітів;
- сформулювати технічні вимоги до модуля з урахуванням інтеграції з особистим кабінетом студента;
- спроектувати архітектуру серверної частини модуля з використанням Go і фреймворку Fiber;
- реалізувати HTML-інтерфейс для користувацької взаємодії з модулем;
- реалізувати логіку генерації навчальної відомості у форматі Excel;
- здійснити тестування модуля та оцінити його відповідність технічним і функціональним вимогам;
- обґрунтувати доцільність та шляхи впровадження модуля у практику роботи ЗВО.

Об'єктом дослідження є процес автоматизації академічного документообігу в системі вищої освіти України, зокрема цифрове формування документів обліку успішності студентів.

Предметом дослідження виступає програмний модуль автоматичного формування навчальної відомості, що взаємодіє з електронним журналом успішності, використовуючи сучасні вебтехнології для обробки та структурування освітніх даних.

У ході дослідження застосовувалися загальнонаукові та спеціальні методи. Зокрема, використовувалися методи аналізу й узагальнення для вивчення теоретичних основ автоматизації документообігу у сфері освіти. Моделювання та

проектно-конструкторський підхід застосовувалися під час побудови архітектури програмного модуля. Для перевірки працездатності розробленого рішення проводилося функціональне тестування в умовах, наближених до реального освітнього середовища.

Наукова новизна роботи полягає у запропонованій архітектурі програмного модуля, який функціонує без прямого доступу до бази даних електронного журналу, натомість працює із вхідним JSON-файлом, що дозволяє значно підвищити адаптивність до різних освітніх систем. Запропоноване рішення передбачає використання високопродуктивних інструментів (Go, Fiber, Excelize) та гнучкого HTML-інтерфейсу для формування індивідуальних звітів у форматі Excel.

Практична значущість полягає у можливості впровадження модуля як частини електронного журналу успішності в будь-якому закладі вищої освіти. Він забезпечує автоматизоване формування навчальних відомостей без участі адміністративного персоналу, знижує імовірність людських помилок, підвищує швидкість доступу до навчальних результатів, що особливо актуально в умовах віддаленого або гібридного навчання. Розробка може бути використана як на рівні окремого факультету, так і в межах централізованої університетської системи.

Апробація результатів бакалаврської роботи:

1. Гаврилюк А.Р. – «Перспективи використання технологій штучного інтелекту для адаптивного навчання у вищій освіті». Тези доповіді на II Всеукраїнській науково-технічній конференції «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу»

2. Гаврилюк А.Р. – «Технологія віртуальної та доповненої реальності в навчальних програмах: підвищення ефективності та мотивації студентів». Тези доповіді на II Всеукраїнській науково-технічній конференції «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу»

3. Гаврилюк А.Р., Дубенець С.В. – «Кібербезпека в архітектурі IoT для розумного міста: захист даних і приватності громадян». Тези доповіді на II

Всеукраїнській науково-технічній конференції «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу»

4. Гаврилюк А.Р., Ярмач Н.В. – «Особливості розробки мобільного застосунку для управління та оптимізації бізнес-процесів автомобільного салону: сучасні підходи та інструменти цифрової трансформації». Тези доповіді на II Всеукраїнській науково-технічній конференції «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу»

РОЗДІЛ 1 ТЕХНОЛОГІЧНІ ОСОБЛИВОСТІ АВТОМАТИЗАЦІЇ ОБЛІКУ УСПІШНОСТІ СТУДЕНТІВ

1.1 Огляд сучасних систем електронного журналу успішності та їх можливостей щодо формування навчальної відомості

В умовах цифрової трансформації вищої освіти автоматизація відстеження академічної успішності стала фундаментальним компонентом ефективного управління університетом. Системи електронних журналів забезпечують централізоване, структуроване та безпечне середовище для управління даними про успішність студентів. Серед впроваджень в українських вищих навчальних закладах, розглянемо системи, розроблені в Київському політехнічному інституті ім. Ігоря Сікорського (КПІ) та Львівському національному університеті ім. Івана Франка (ЛНУ).

Система електронних журналів КПІ ім. Ігоря Сікорського. Система електронних залікових книжок, розроблена в КПІ, демонструє комплексний інтерфейс для викладачів для запису та управління показниками успішності студентів. Як показано на рисунку 1.1, платформа пропонує табличну структуру, де кожен рядок відповідає студенту, а стовпці представляють завдання курсу, результати іспитів або модульні оцінки. Структура оптимізована для зручності читання даних та ефективного введення нових записів. Функціонал включає фільтрацію за студентською групою, предметом та семестром, що значно спрощує адміністративну роботу під час екзаменаційної сесії.

Ілюстрація внутрішньої логіки управління оцінками, де механізми перевірки вхідних даних забезпечують узгодженість та усувають помилки при введенні оцінок (рис. 1.2). Система КРІ інтегрується безпосередньо з інституційними базами даних і може експортувати дані у структуровані формати, такі як CSV або XLSX, які згодом можуть бути використані для ручного складання академічних довідок. Однак, незважаючи на потужний функціонал для відстеження успішності, системі

бракує власної підтримки для автоматизованого створення повних академічних довідок або форм, що відповідають стандартам Міністерства освіти.

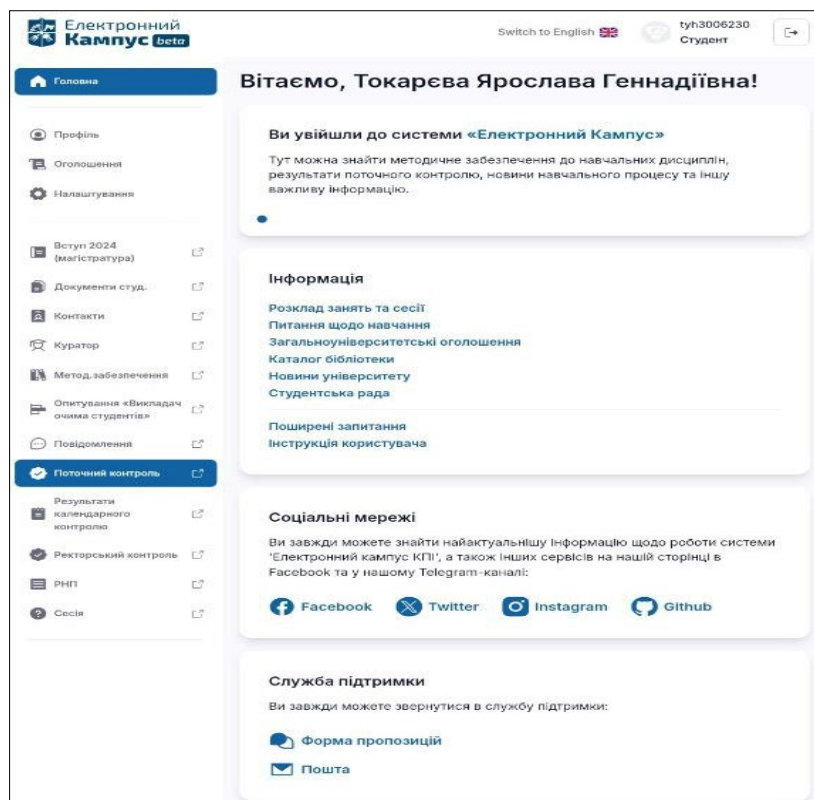


Рис. 1.1 Кабінет студента КПІ [27]

Вітаємо, Токарева Ярослава Геннадіївна! Вийти

Результати календарного контролю

Контакти	Дисципліна	Викладач	1 семестр Атестація №1	1 семестр Атестація №2	2 семестр Атестація №1	2 семестр Атестація №2
Повідомлення	Практичний курс: іноземні мови. Частина 2	Тихоненко М. А.	а	а	а	
Куратор	Стилі в образотворчому мистецтві	Олійник Е. В.	н/в	н/в		
РНП	Основи підприємницької діяльності	Косовиченко D. O.	н/в	н/в		
Метод. забезпечення	Силові види спорту	Сиринська О. К.	н/в	н/в		
Поточний контроль	Етика і психологія інтелекту	Касюкова М. А.	н/в	н/в		
Ректорський контроль	Математична логіка і теорія алгоритмів. Частина 2. Теорія аксіоматик і формальних мов, мови Петрі	Стус Д. В.	а	а		
Сесія	Ігрові види спорту	Ал. Івасюк Л. В.	н/в	н/в		
Вступ 2024 (магістратура)	Теорія ймовірностей	Васильчук О. І.	а	а		
Документи (студ.)	Диференціальні рівняння	Зеленська І. О.	а	а	н/в	
■ Результати атестації	Еристика	Кайс З. В.	а	а		
	Естетика промислового дизайну	Кузін О. Ю.	н/в	н/в		
	Логіка	Поташук О. О.	н/в	н/в		
	Об'єкти-орієнтоване програмування	Куква Ю. В.	а	а		
	Фізика, Частина 2. Електрика та магнетизм	Спарський А. О.	а	а		
	Математичний аналіз. Частина 3. Криволінійні та поверхневі інтеграли	Поліщук Н. Д.	а	а		
	Основи академічної доброчесності	Кулик Є. В.	н/в	н/в		
	Психологія конфлікту	Хіліна С. О.			н/в	
	Політична наука: конфліктологічний підхід	Скверніченко Д. Г.			а	
	Чисельні методи	Дмитрієва О. А.			а	
	Психологія	Хіліна С. О.			н/в	
	Вступ до філософії	Поташук О. О.			а	
	Теорія стійкості та варіаційне числення	Зеленська І. О.			а	
	Економічна психологія	Чуликов В. О.			н/в	
	Громадянська відповідальність та політична участь: трансформаційні виклики для України	Пашов Р. І.			н/в	
	Математична статистика	Салобідинова І. Р.			а	
	Застос. персональні дані: стандарти GDPR та Роді Сарала	Семеницька О. А.			н/в	
	Операційні системи	Коваленко А. Є.			а	
	Архітектура обчислювальних систем і комп'ютерних мереж	Артемчук Я. Г.			а	
	Основи енергозбереження та енергооптимізація	Чеснаков Г. І.			н/в	
	Гармонічний аналіз та операційне числення	Попова Н. Д.			а	

Рис. 1.2 Результати студента КПІ за поточний семестр [27]

Система електронного журналу ЛНУ ім. Івана Франка. На противагу цьому, система ЛНУ робить значний акцент на автоматизації адміністративної звітності. Як показано на рисунку 1.3, на інформаційній панелі представлені як узагальнені дані про успішність, так і посилання на завантажувану документацію про успішність. Платформа покликана зменшити залежність від ручної компіляції даних, тим самим підвищуючи точність та ефективність академічних операцій.

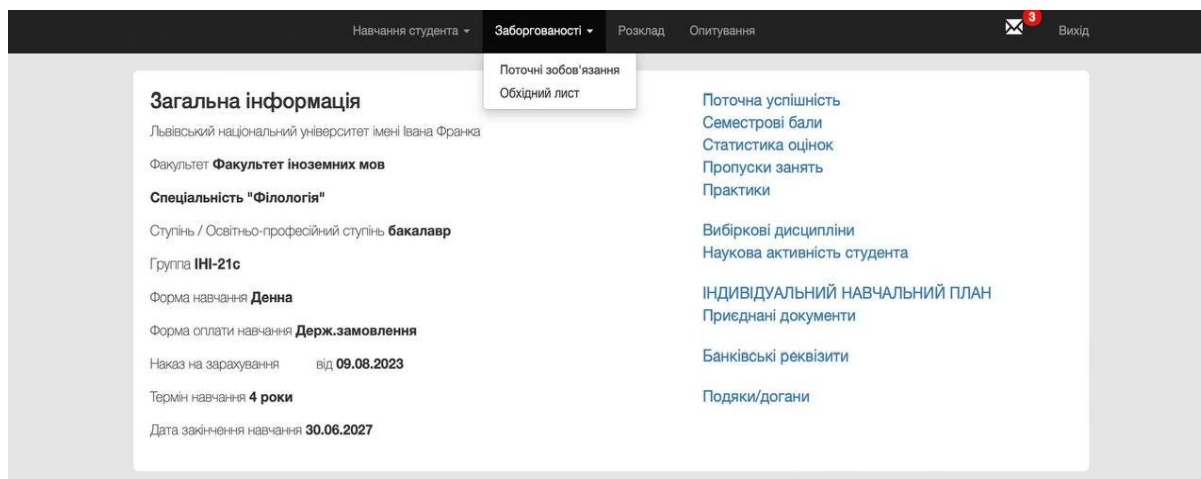


Рис. 1.3 Кабінет студента ЛНУ ім. Івана Франка

На рисунку 1.4 зображено інтерфейс розкладу студента, який є невід'ємним компонентом електронної системи ЛНУ. Цей модуль надає студентам доступ у режимі реального часу до їхніх персональних розкладів, включаючи детальну інформацію про назви курсів, викладачів, дати, час та місця проведення занять. Система динамічно генерує розклад на основі офіційних реєстраційних даних курсу, гарантуючи, що студенти будуть поінформовані про всю поточну академічну діяльність. Крім того, вона підтримує оновлення в режимі реального часу, які автоматично відображаються на студентському порталі в разі змін у розкладі. Ця функціональність значно підвищує прозорість організації та зменшує адміністративні затримки в поширенні інформації про розклад.

Розклад занять

з дати: 22.04.2025 до дати: 29.04.2025 Показати розклад на...
 ПІБ викладача Показати розклад занять

22.04.2025 Вівторок			24.04.2025 Четвер		
1	08:30 09:50	Історія світової літератури (Л) доцент Кушнір Ірина Богданівна Потік	1	08:30 09:50	
2	10:10 11:30	Лексикологія (іспанської мови) (Л) доцент Цимбалістий Ігор Юліанович	2	10:10 11:30	
3	11:50 13:10	Start-up менеджмент (Л) доцент Кохан Маріанна Остапівна Збірна група Креативне бізнес-мислення (Л) доцент Куліш Інна Михайлівна Збірна група 100/І Гендерні стереотипи та соціальна активність молоді (Л) доцент Корняк Віра Степанівна Збірна група Бізнес-аналітика засобами Excel та Power BI (ПрС) доцент Панчишин Тарас Володимирович Збірна група Астрономія для всіх (Л) доцент Стельмах Оксана Миколаївна Збірна група Юридичний захист прав водіїв (ПрС) асист. Заболотна Галина Ігорівна Збірна група 613/Г Навчальні географо-екологічні маршрути Карпатського регіону (Л) доцент Матвій Володимир Павлович Збірна група 7/В Юридичний захист прав водіїв (Л) професор Решета	3	11:50 13:10	
			4	13:30 14:50	Друга іноземна мова (англійська) (ПрС) доцент Сподарик Оксана Володимирівна Збірна група 421/А Друга іноземна мова (французька) (ПрС) асистент Миколаєвич-Джуман Наталія Богданівна Збірна група Друга іноземна мова (англійська) (ПрС) асистент Безугла Олена Володимирівна Збірна група 422/А
			5	15:05 16:25	Перша іноземна мова (іспанська) (ПрС) доцент Кабов Аркадій Вікторович Збірна група 418/А Перша іноземна мова (іспанська) (ПрС) доцент Сайфутдінова Олена Юріївна Збірна група 407/А Перша іноземна мова (іспанська) (ПрС) асистент Осечко Світлана Миколаївна Збірна група 408/А

Рис. 1.4 Розклад занять студента ЛНУ ім. Івана Франка

Основна перевага системи ЛНУ полягає в її здатності генерувати академічні довідки та звіти про успішність з мінімальним втручанням людини. Як показано на рисунку 1.5, платформа включає вбудований модуль для генерації транскриптів, який витягує дані з реєстру активних курсів і форматує їх відповідно до стандартизованих шаблонів. Ця функціональність дозволяє адміністраторам створювати придатні для друку академічні довідки для внутрішнього використання або подання зовнішнім зацікавленим сторонам.

Журнал поточної успішності

Оберіть дисципліну!!!
 Перічня ☐ перше ☐ друге ☒ тільки мої оцінки

з: dd.mm.yyyy по: dd.mm.yyyy Оберіть місяць Журнал Підсумки

Для перегляду журналу успішності необхідно обрати дисципліну, задати семестр, вказати діапазон дат або вибрати потрібний місяць і натиснути на кнопку "Журнал"
 Для перегляду підсумків з обраного предмету необхідно в полі вибору місяця обрати "За весь період" і натиснути на кнопку "Журнал" або натиснути на кнопку "Підсумки"

Рис. 1.5 Журнал поточної успішності ім. Івана Франка

Крім того, на рисунку 1.6 показано, як система підтримує пакетну обробку записів. Одночасно можна створювати декілька транскриптів для всієї групи студентів, що є суттєвим покращенням порівняно з системами, які вимагають збирання даних для кожного студента.

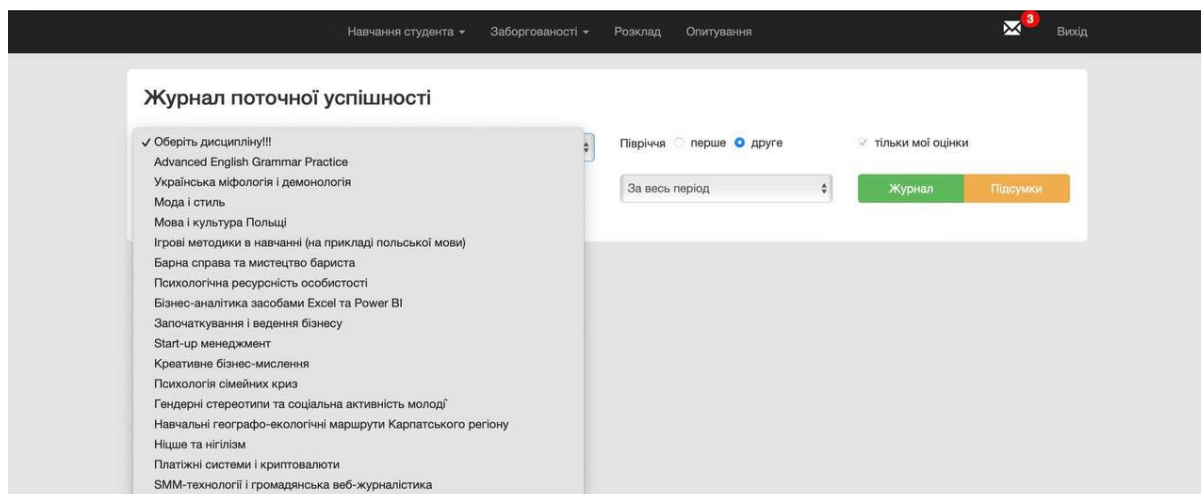


Рис. 1.6 Журнал поточної успішності ім. Івана Франка

На рисунку 1.7 представлено функціонал модуля звітності з відвідуваності, який забезпечує автоматизоване моніторинг та візуалізацію відвідування студентами запланованих академічних заходів. Система надає структуроване подання даних у табличному форматі, включаючи такі параметри: найменування навчальних дисциплін, відповідні дати проведення занять, а також статуси відвідування (присутній, відсутній, поважна причина).

Особливістю даного інтерфейсу є можливість оперативного введення інформації викладачами в режимі реального часу, що суттєво підвищує ефективність документування навчального процесу. Усі дані зберігаються в централізованій базі, що забезпечує їх подальше використання для: формування академічної звітності, проведення внутрішніх аудиторських перевірок, а також організації комунікації між викладачами та студентами щодо відвідування навчальних занять.

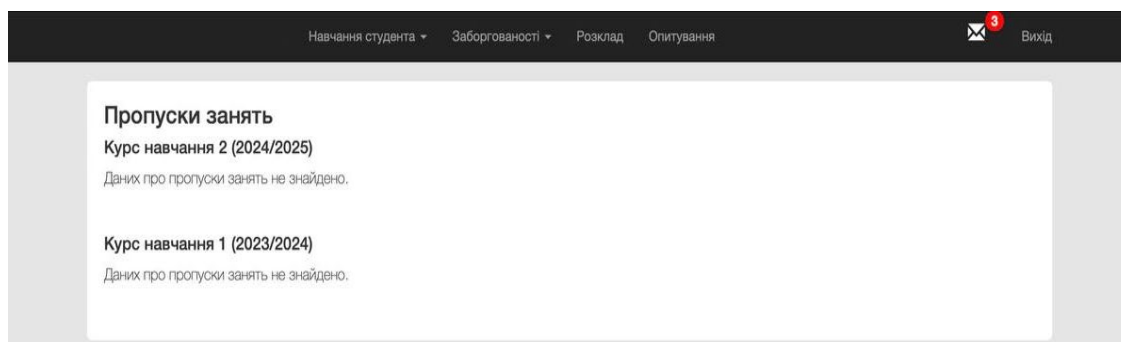


Рис. 1.7 Звітність щодо відвідуваності студента ЛНУ ім. Івана Франка

Обидві системи електронного журналу успішності – КПП та ЛНУ – включають інструменти для обліку академічних досягнень студентів, однак їхні можливості щодо автоматизованого формування офіційної навчальної відомості суттєво відрізняються за рівнем функціональності та ступенем автоматизації.

Система КПП, хоча й забезпечує ефективне ведення обліку оцінок, не має вбудованого або повністю автоматизованого механізму формування повної навчальної відомості. Викладачам доводиться вручну експортувати дані про успішність студентів та самостійно формувати відповідні офіційні форми за допомогою зовнішніх шаблонів або додаткових програм. Такий підхід, хоч і частково цифровізований, вимагає значних трудозатрат і підвищує ризик виникнення помилок, особливо в період сесії або масового оформлення звітності.

На відміну від цього, система ЛНУ демонструє вищий рівень автоматизації при формуванні навчальної відомості. На основі введених у систему даних – зокрема оцінок, відвідуваності та статусу виконання навчальних компонентів – платформа автоматично заповнює уніфіковані шаблони навчальної відомості. Ці шаблони відповідають як внутрішнім вимогам університету, так і стандартам Міністерства освіти, що усуває потребу в повторному ручному введенні однієї й тієї ж інформації. Крім того, підтримка пакетної генерації документів для цілих академічних груп значно підвищує масштабованість адміністративних процесів. Такий підхід свідчить про орієнтацію системи ЛНУ на сучасні вимоги цифрової документації та прозорості академічного процесу.

Таким чином, попри те, що обидві системи ефективно виконують базову функцію обліку успішності, лише рішення ЛНУ забезпечує безперервний та автоматизований процес формування навчальних відомостей безпосередньо з електронного журналу. Ця відмінність підкреслює необхідність подальшої розробки та модульного розширення систем на зразок тієї, що використовується в КПП, з метою задоволення зростаючих потреб навчальних закладів у комплексній академічній звітності.

1.2 Методологія та досвід автоматизації документообігу в навчальних закладах України

Сучасна вища освіта в Україні стрімко трансформується під впливом цифровізації, що зумовлює необхідність автоматизації внутрішніх управлінських процесів, зокрема документообігу. Однією з ключових складових цифрової трансформації є впровадження систем електронного документообігу (ЕДО), що дозволяють забезпечити оперативне зберігання, обробку, передачу та використання навчальної та адміністративної документації. Методологічна основа таких систем передбачає використання концепцій одноразової реєстрації документа, безперервності його руху в системі та неможливості дублювання, що особливо важливо для освітніх установ з великою кількістю документів [26, 20].

Електронний документообіг – це група операцій, які виконуються за допомогою комп'ютерних мереж і включають створення, збереження, обробку, передачу та знищення електронних документів, зберігаючи їх цілісність і законність [26].

Застосування систем ЕДО базується на таких концептуальних підходах на сучасному етапі розвитку української вищої освіти:

- документація зберігається в одному місці;
- фіксація кожного документа окремо;
- забезпечення нагляду за переміщенням документів;
- уникнення дублювання та втрати даних;
- взаємодія з інформаційними системами ЗВО.

В Україні досвід автоматизації документообігу в навчальних закладах є поки що фрагментарним, однак існують успішні приклади. Одним із них є досвід Тернопільського державного медичного університету імені І. Я. Горбачевського (ТДМУ), де у 2013-2014 роках було впроваджено систему електронного документообігу Alfresco на базі відкритого програмного забезпечення. Основними критеріями її впровадження стали безкоштовна ліцензія, простота адміністрування та сумісність з існуючою ІТ-інфраструктурою університету [26, 20].

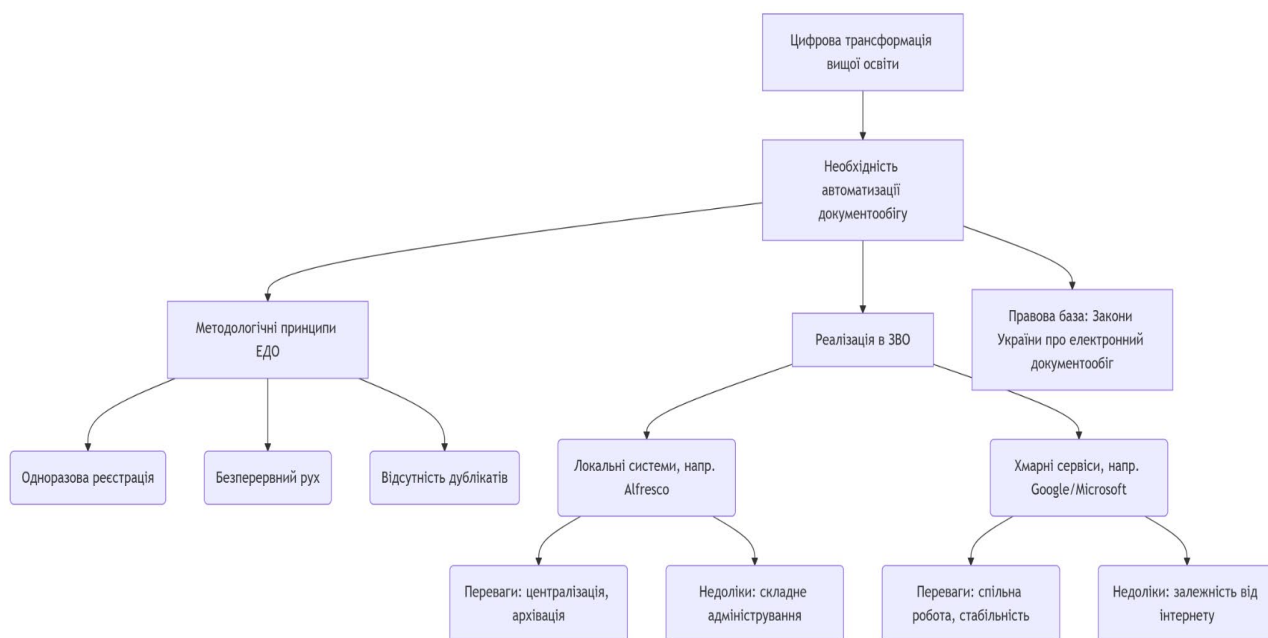


Рис. 1.8 Система електронного документообігу (ЕДО) у ЗВО України

В ході експлуатації системи було виявлено низку переваг: централізований доступ до документів, можливість розподілу завдань, використання календаря та повна архівація документації. Однак були зафіксовані й недоліки: складність у розширеному адмініструванні, висока чутливість до збоїв апаратного забезпечення, що змусило університет шукати альтернативи.

У зв'язку з цим, наступним логічним кроком стало впровадження хмарних сервісів Google Apps for Education та Microsoft Office 365 у практику освітньої діяльності. Зазначені платформи, розроблені спеціально для потреб навчальних закладів, надають безкоштовний доступ для акредитованих освітніх установ, що створює сприятливі умови для впровадження сучасних інформаційних технологій у навчальний процес. Обидві системи забезпечують широкі можливості для організації спільної роботи учасників освітнього процесу, зокрема підтримують редагування документів у реальному часі, обмін файлами, коментування та відстеження змін. Також до функціональних можливостей входить зберігання інформаційних ресурсів у хмарному середовищі з доступом з будь-якого пристрою, що має підключення до Інтернету. Інтегровані інструменти комунікації – такі як електронна пошта, календарі, відеозв'язок, сповіщення та спільні чати – значно розширюють можливості для оперативної взаємодії між викладачами, студентами

та адміністративним персоналом. Додатково слід зазначити, що обидві платформи пропонують розвинену систему адміністрування з можливістю гнучкого налаштування доступу до ресурсів, створення груп користувачів, моніторингу активності та впровадження політик безпеки. Таблиця 1.1 нижче ілюструє порівняння ключових можливостей зазначених хмарних сервісів як платформ для ЕДО в ЗВО.

Таблиця 1.1

Порівняльна характеристика хмарних сервісів для електронного документообігу

Критерій	Google Apps for Education	Microsoft Office 365
Модель доступу	Безкоштовна для ЗВО	Безкоштовна для ЗВО
Платформа зберігання	Google Drive	OneDrive
Спільна робота з документами	Google Docs, Sheets, Slides	Word Online, Excel Online, PowerPoint
Підтримка відеоконференцій	Google Meet	Microsoft Teams, Lync
Легкість адміністрування	Висока	Середня
Чутливість до перебоїв системи	Низька (через хмарну інфраструктуру)	Низька
Інтеграція з LMS	Так	Так

Варто також відзначити, що деякі українські ЗВО (наприклад, Університет "НТУ") впровадили власні адаптовані рішення на базі open-source систем або побудували корпоративні інформаційні середовища на платформі [26].

Методи автоматизації документообігу в освітніх установах України повинні базуватися на таких основних принципах:

- Принцип централізованого управління передбачає, що всі документи повинні зберігатися в єдиному цифровому сховищі, доступному для авторизованих користувачів. Централізоване зберігання дозволяє уникнути розпорошеності інформації, спрощує процес контролю, уніфікує підходи до класифікації та

архівування документів. Завдяки цьому забезпечується цілісність і структурованість документопотоку, а також спрощується адміністрування даних.

- Відмова від дублювання. Автоматизована система документообігу повинна запобігати створенню кількох копій одного й того ж документа. Це реалізується шляхом впровадження механізмів контролю унікальності, автоматичного визначення версій та перевірки відповідності нових документів до вже існуючих. Відсутність дублювання сприяє підвищенню достовірності інформації, зменшенню ризику помилок та покращенню ефективності пошуку.

- Безперервний рух документу полягає у забезпеченні автоматичного маршрутизації документів від одного відповідального працівника до іншого відповідно до встановлених регламентів або бізнес-процесів. Система має самостійно визначати, кому й коли має бути переданий документ для ознайомлення, погодження або підписання. Це дозволяє уникнути простоїв, забезпечити дотримання строків та підвищити загальну оперативність документообігу.

- Аудит доступу є важливою функцією будь-якої системи електронного документообігу є ведення повної історії всіх дій з документом. Це включає фіксацію того, хто переглядав документ, вносив зміни, пересилав або завантажував його. Аудит дозволяє здійснювати контроль за дотриманням процедур, забезпечує прозорість роботи системи, а також може бути використаний для вирішення спірних ситуацій або виявлення порушень.

Відповідність чинному законодавству. Усі зазначені принципи реалізуються в межах правового поля, визначеного законодавством України, зокрема Законом України «Про електронні документи» та Законом України «Про електронний документообіг». Ці нормативно-правові акти визначають юридичний статус електронних документів, встановлюють порядок їх створення, обігу, зберігання та використання, а також регламентують права й обов'язки суб'єктів документообігу. Дотримання цих норм забезпечує легітимність та правову захищеність усіх дій, пов'язаних з обробкою електронних документів в освітніх установах [26].

Таблиця 1.2

Порівняння підходів до впровадження ЕДО в ЗВО

Параметр	Alfresco (локальна)	Google Apps for Education (хмарна)
Вартість	Безкоштовна	Безкоштовна
Простота розгортання	Середня	Висока
Потреба в технічному персоналі	Висока	Низька
Стабільність	Залежить від сервера	Висока
Адміністрування	Складне	Просте
Підтримка співпраці	Часткова	Повна
Можливість резервного копіювання	Обмежена	Вбудована

Таким чином, досвід, накопичений в Україні, свідчить про те, що використання хмарних сервісів і відкритих програмних рішень є дуже ефективним у сфері електронного навчання. Такі технології сприяють прозорості навчального процесу, пришвидшують обмін інформацією між підрозділами та гарантують надійний контроль за документообігом.

1.3 Програмні методи генерації електронних документів обліку

У сучасних закладах вищої освіти спостерігається стрімке впровадження інформаційних технологій у процеси управління освітньою, адміністративною та фінансовою діяльністю. Однією з ключових складових цифрової трансформації є автоматизація документообігу, яка передбачає використання програмних засобів для створення, обробки та зберігання електронних документів. Серед них особливе місце займають документи обліку – навчальні відомості, залікові відомості, аналітичні звіти, накази та довідки. Програмні методи генерації таких документів

дозволяють оптимізувати документообіг, зменшити навантаження на персонал і забезпечити відповідність нормативним вимогам [20, 28].



Рис. 1.9 Етапи обробки документу незалежно від обраного методу генерації

Від ефективності організації системи генерації електронних документів залежить якість управлінських рішень, швидкість адміністративних процедур, прозорість навчального процесу та відповідність внутрішньої документації вимогам Міністерства освіти і науки України [21, 24]. У зв'язку з цим постає потреба у дослідженні та класифікації основних методів, які застосовуються для формування електронних документів у вищих навчальних закладах [26, 28]. Виділяють такі ключові підходи до генерації документів:

1. Шаблонні системи документообігу

Цей метод передбачає використання стандартизованих форм документів (наприклад, DOCX, PDF, ODT), які автоматично заповнюються даними з баз даних. Технології злиття даних (mail merge) дозволяють масово створювати однотипні документи, що особливо корисно для формування навчальних відомостей [21, 28].

Перевагами є простота впровадження та можливість швидкої генерації документів, проте обмежена гнучкість при роботі зі складними формами документації [20].

2. Системи управління документами (DMS)

Сучасні DMS-платформи, такі як Alfresco, OpenText та DocuWare, пропонують розширені функції генерації документів:

- Інтеграція з академічними інформаційними системами
- Автоматичне формування наказів, довідок, звітів
- Підтримка версійності документів

Ці системи дозволяють централізовано керувати документообігом, забезпечуючи контроль за рухом документів і запобігання дублюванню даних [26, 30].

Таблиця 1.3

Вибір DMS-платформи для ЗВО

Критерій	Alfresco	Google Workspace	Microsoft 365
Вартість	Відкрита версія	Безкоштовно для ЗВО	Безкоштовно для ЗВО
Інтеграція з LMS	Через API	Google Classroom	Teams
Генерація PDF	Так	Через скрипт	Power Automate
Е-підпис	Потрібні додатки	Підтримка КЕП	Підтримка КЕП

3. Програмні інтерфейси (API) генерації

Для складних випадків застосовуються спеціалізовані API, які забезпечують високий рівень автоматизації:

- REST/SOAP API для інтеграції з ERP-системами
- Бібліотеки генерації PDF (iText, PDFKit)
- Підсистеми електронного підпису

Хоча впровадження цього методу потребує суттєвих технічних ресурсів, зокрема кваліфікованого персоналу та розробки індивідуальних інтеграційних рішень, він забезпечує високу адаптивність до специфічних потреб установи. API-рішення часто використовуються у випадках, коли необхідна інтеграція з кількома зовнішніми системами або коли стандартні шаблонні інструменти виявляються недостатніми. Крім того, такі інтерфейси відкривають можливості для подальшого розширення функціональності системи, зокрема впровадження аналітики, електронного документообігу та безпечного зберігання даних. [26, 20].

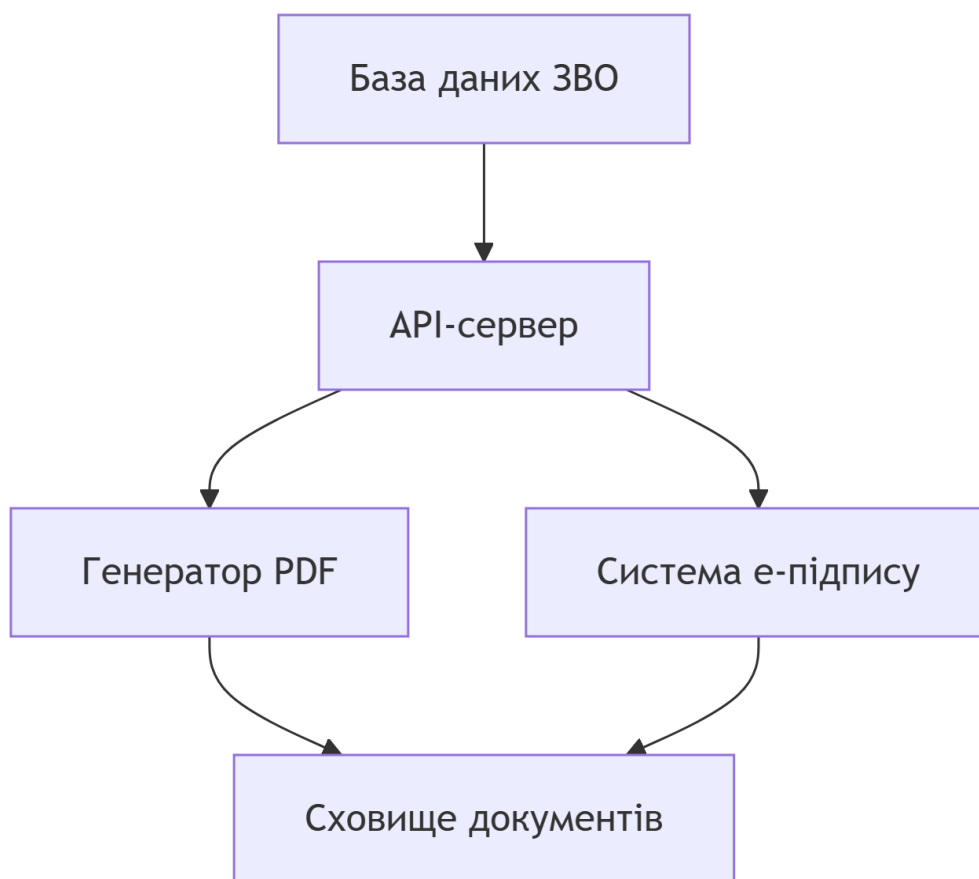


Рис. 1.10 Архітектура API-генерації документів

4. Low-code/No-code підходи

Сучасний тренд – використання платформ швидкої розробки, таких як Microsoft Power Apps та Google AppSheet. Вони дозволяють створювати додатки для генерації документів без глибоких технічних знань, використовуючи drag-and-drop конструктори форм та візуальне програмування бізнес-процесів [12, 5].

5. Інноваційні технології

Перспективними напрямками є:

- Використання блокчейн-технологій для забезпечення цілісності документів
- Застосування штучного інтелекту для автоматичного аналізу та класифікації
- Інтеграція NLP-технологій для обробки текстів

Ці технології формують основу наступного етапу цифрової трансформації документообігу у сфері освіти [28].

Таблиця 1.4

Порівняльна характеристика методів генерації документів

Метод	Переваги	Недоліки	Прикладність для ЗВО
Шаблонні системи	Простота, масове створення, низька вартість	Обмежена гнучкість, ручна обробка складних форм	Накази, довідки, заяви
DMS-платформи	Централізоване управління, версійність	Висока вартість, складність впровадження	Архіви, навчальні плани
API-генерація	Висока автоматизація, інтеграція з ERP	Вимагає технічних спеціалістів	Дипломні роботи, звітність
Low-code/No-code	Швидка розробка, мінімум кодування	Обмежені функціональні можливості	Внутрішні форми, анкети
Інноваційні технології	Автоматизація аналізу, захист даних	Дорогі впровадження, експериментальність	Наукові звіти, міжнародні документи

Аналіз програмних методів генерації електронних документів демонструє, що сучасні ЗВО мають широкий вибір інструментів – від простих шаблонних систем до складних API-рішень. Оптимальний вибір залежить від масштабу закладу, технічної інфраструктури та специфіки ділових процесів. Додатковим важливим фактором при виборі методів генерації електронних документів є рівень інтеграції із внутрішніми та зовнішніми цифровими системами, що використовуються у закладі вищої освіти. Крім того, варто враховувати перспективи подальшої еволюції інструментів – наприклад, впровадження модулів штучного інтелекту для автоматичної обробки, класифікації та аналізу документів. Саме тому оптимальне рішення повинно передбачати не лише поточні функціональні потреби, а й бути масштабованим та адаптивним до майбутніх технологічних змін у сфері освіти та електронного урядування. Найбільш ефективним на сучасному етапі є комбінований підхід, що поєднує гнучкість програмних інтерфейсів, зручність шаблонів і можливості хмарних сервісів.

РОЗДІЛ 2 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ІСНУЮЧИХ РІШЕНЬ АВТОМАТИЗАЦІЇ ОБЛІКУ УСПІШНОСТІ У ВИЩИХ НАВЧАЛЬНИХ ЗАКЛАДАХ

2.1 Огляд та аналіз недоліків популярних рішень електронного журналу в контексті формування навчальних відомостей

У сучасній системі вищої освіти електронні журнали успішності виступають базовим інструментом для фіксації та обробки результатів навчальної діяльності студентів. Їх активне впровадження зумовлено загальною тенденцією цифровізації освітнього середовища, необхідністю оптимізації адміністративного навантаження на викладачів та підвищення прозорості оцінювання. Проте, попри поширеність електронних журналів, актуальною залишається проблема ефективного формування навчальних відомостей на їх основі – особливо в умовах різномірних навчальних планів, модульно-рейтингових систем і вимог до офіційної звітності.

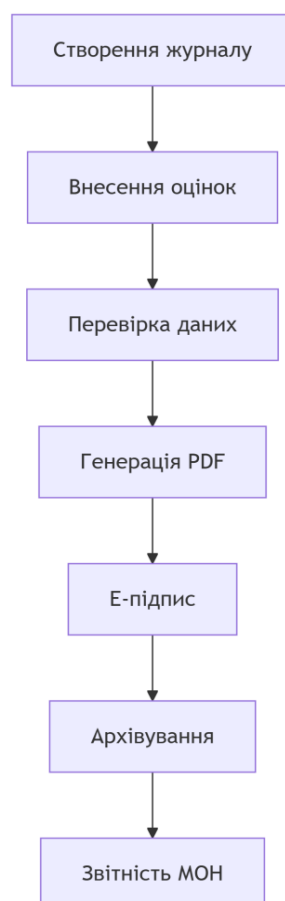


Рис. 2.1 Процес роботи з відомостями

Оцінювання ефективності електронних журналів успішності передбачає застосування чітко визначених критеріїв, що відображають як технічні, так і організаційно-правові аспекти функціонування систем. Такий підхід дозволяє здійснити об'єктивне порівняння різних рішень, визначити їх відповідність потребам закладів вищої освіти та виявити потенційні напрямки для вдосконалення. Основні критерії оцінювання:

- Користувацький досвід (UX): зручність інтерфейсу, наявність адаптивного доступу (через веб або мобільний додаток), інтуїтивність управління журналом.
- Функціональність: чи відповідають можливості системи вимогам ведення навчального процесу: підтримка різних типів дисциплін (лекції, практичні, лабораторні), модульно-рейтингової системи (МРС), індивідуальних навчальних планів.
- Інтеграція: здатність платформи до інтеграції з іншими системами університету (електронним деканатом, студентською інформаційною системою, системою ЕДЕБО), а також із зовнішніми сервісами (Google Workspace, Microsoft 365) [16].
- Організаційні та юридичні аспекти: відповідність системи положенням Закону «Про електронні документи та електронний документообіг» (наявність електронного підпису, зберігання оригіналів в електронній формі) [23], рівень захисту персональних даних, вимоги до навчального процесу (погодження з МОН).

Важливим є забезпечення функцій генерації звітів і відомостей згідно з державними стандартами: створення електронних відомостей про успішність (залікових книжок, додатків до диплома тощо), їх експорт і підпис згідно з законодавством [27].

З метою комплексного аналізу сучасних електронних платформ, що використовуються у закладах вищої освіти для ведення обліку успішності, доцільним є розгляд як міжнародних систем, так і локальних рішень, розроблених безпосередньо в межах українських університетів. Такий підхід дозволяє не лише виокремити переваги та обмеження кожної з існуючих систем, а й виявити спільні

структурні вади, що значною мірою впливають на можливість повноцінного формування офіційних навчальних відомостей. У цьому контексті особливу увагу заслуговують платформи, що охоплюють як відкриті середовища навчального управління (LMS), так і спеціалізовані інституційні інформаційні комплекси, призначені для обліку академічної успішності.

Однією з найбільш поширених у світі систем є Moodle, зокрема її модуль журналу оцінок. Цей інструмент дозволяє викладачам виставляти оцінки, категоризувати завдання, а також формувати підсумкові бали. Moodle підтримує експорт оцінок у широкому спектрі форматів – зокрема CSV, Excel, OpenDocument та XML, а також імпорт даних у зворотному напрямку. Оскільки журнал оцінок є частиною базової архітектури LMS, користувачам не потрібно проходити складне навчання, що забезпечує порівняно швидке освоєння системи. Водночас, Moodle у своєму стандартному варіанті не містить модулів для формування офіційних відомостей, зокрема з електронним підписом чи відповідно до вимог закладів вищої освіти. Для таких функцій зазвичай необхідно встановлювати додаткові плагіни або створювати спеціальні скрипти. Інтеграція з зовнішніми хмарними сервісами – зокрема Google Drive або Microsoft 365 – можлива, але вимагає окремого налаштування та підтримки на рівні адміністрування.

Іншим прикладом, який активно використовується у закордонній практиці, є система Classroom.cloud (NetSupport), орієнтована переважно на управління класними приміщеннями та забезпечення ІТ-безпеки. Вона підтримує такі функції, як голосове управління, обмін файлами та моніторинг екранів студентів, однак не має вбудованого модуля для фіксації оцінок або ведення навчальної звітності. Через це система більше підходить для організаційного адміністрування, а не для реалізації освітніх функцій, пов'язаних із формуванням офіційних звітів про успішність.

В Україні прикладом успішної локальної розробки є електронний кампус Національного технічного університету України "КПІ ім. Ігоря Сікорського". Інформаційно-аналітичний комплекс "Campus" забезпечує можливість формування електронних відомостей за результатами семестрового контролю, де

викладачі мають змогу вводити оцінки онлайн, а дані автоматично передаються до підсистеми “Деканат”. Починаючи з 2020 року, в системі реалізовано функцію автоматичного формування звітів деканатом, які відображаються в модулі “Сесія” особистого кабінету викладача. Також реалізована часткова інтеграція з централізованими державними системами, зокрема ЄДЕБО (Єдина державна електронна база з питань освіти). При цьому первісні версії електронного кампусу мали низку обмежень, зокрема неінтуїтивний інтерфейс та значне навантаження на ІТ-персонал, пов’язане з обслуговуванням. Хоча нові модулі поступово відповідають вимогам МОН, проблеми мобільного доступу, а також застарілий візуальний інтерфейс усе ще залишаються актуальними.

Ще одним прикладом української локальної розробки є програмний засіб “Журнал успішності”, створений у Львівському національному університеті імені Івана Франка. Система реалізована як власне рішення для реєстрації успішності студентів, що функціонує в рамках кредитно-трансферної системи та забезпечує можливість формування різного типу звітів – від семестрових та річних відомостей до додатків до диплому й рейтингових списків. Згідно з методичними матеріалами університету, студенти мають змогу переглядати свої оцінки через веб-інтерфейс, а викладачі – формувати електронні журнали з автоматичним підрахунком результатів. Інтеграція з ЄДЕБО, можливість експорту звітів у шаблони МОН, а також підтримка друку індивідуальних навчальних планів свідчать про високий рівень адаптації до національних освітніх вимог. Утім, застарілий та громіздкий інтерфейс, а також складність у налаштуванні, що вимагає високої кваліфікації ІТ-фахівців, істотно знижують ефективність повсякденного використання системи.

На тлі описаних прикладів заслуговують на увагу й інші популярні платформи, зокрема ті, що використовуються у міжнародній практиці. Наприклад, система Canvas є повнофункціональною LMS, яка підтримує потужний механізм роботи з оцінками та може бути інтегрована з Google Drive чи Microsoft 365. Також комерційні рішення на кшталт Gradelink або Skyward поєднують у собі функції обліку відвідуваності, управління навчальними планами та бюджетами. Деякі університети у світі та в Україні використовують Google Classroom (особливо у

складі Google Workspace), Microsoft Teams for Education або інструменти швидкої розробки, такі як Google AppSheet чи Microsoft Power Apps, для побудови власних електронних журналів. Хоча такі платформи пропонують простий у використанні інтерфейс та інтеграцію зі звичними офісними сервісами, вони зазвичай не мають модулів, адаптованих до формальних вимог державної освітньої звітності. Їх основні переваги полягають у знижених витратах на ІТ-інфраструктуру та високій зручності адміністрування. Проте водночас залишаються виклики, пов'язані з необхідністю адаптації до національних стандартів освіти, залежністю від стабільного інтернет-з'єднання та відсутністю гарантій довготривалої технічної підтримки.

Комплексне порівняння міжнародних і локальних електронних журналів демонструє, що жодна з систем не є універсальним рішенням. Вибір платформи має базуватися на балансі між функціональністю, зручністю, відповідністю нормативним вимогам і можливостями інтеграції в існуючу інформаційну екосистему закладу освіти.

Таблиця 2.1

Порівняльна таблиця функціональності систем [27, 25]

Функціонал	Moodle (Gradebook)	Classroom. cloud	КРІ «Електронний кампус»	ЛНУ «Журнал успішності»	Інші (Google, MS тощо)
Генерація відомостей	Через плагіни/скрипти	—	Автоматичне формування залікових відомостей	Семестрові, річні, додатки до диплому	В основному ні (небагато LMS прямо формують PDF-відомості)
Експорт (CSV/PDF, Excel)	CSV, Excel, ODS	—	Частково (внутрішній формат, перехід в Excel через АС «Деканат»)	Формати MS Excel, OpenDocument	Обмежено: напр. Google Classroom – в Google Таблиці, Teams – через Excel

Продовження таблиці 2.1

Функціонал	Moodle (Gradebook)	Classroom. cloud	КРІ «Електронний кампус»	ЛНУ «Журнал успішності»	Інші (Google, MS тощо)
Статистика та звіти	Мінімально (середній бал, історія)	—	Формуються звіти за потребою адміністрації	Розширені звіти: успішність по предметам, рейтинги	Зведені таблиці в Google Sheets, Power BI інтеграція тощо
Електронний підпис (КЕП)	Ні (зовнішня інтеграція потрібна)	—	Передбачена (за допомогою КЕП при затвердженні відомостей)	Ні (може готувати документи для підпису)	Якщо вбудовані засоби КЕП (наприклад, через Міністерство юстиції)
Інтеграція з АС «Деканат»/ЄДЕБО	Часткова (плагіни для експорту даних)	—	Активно (автоматичний обмін з АС «Деканат»)	Синхронізація з ЄДЕБО, XML-замовлення на студентські	Залежить від реалізації (Google створює CSV для ЄДЕБО, MS збір через Excel)
Управління користувачами (Ролі)	Групи, класи, налаштування доступу	Викладачі, студенти	Через корпоративні акаунти КПП	Студенти/викладачі інтегровані в ЄДЕБО	LDAP/AD для MS, Google Workspace акаунти
Мобільна підтримка	Адаптивний дизайн	Додаток	Частково (моб. версія вузу)	Обмежено (адаптивна версія сайту деканату)	Мобільні додатки Google/Teams/Canvas

Технічні проблеми: Деякі системи демонструють низьку стабільність і продуктивність. Повідомляється, що популярні хмарні платформи можуть зазнавати «масштабних технічних збоїв», які унеможливлюють роботу з тестами або журналами [19]. Наприклад, у коментарях користувачів зазначається, що іноді вони працюють повільно або «зникають» під час перевірок та формування звітності. Існують також проблеми з безпекою даних: якщо система не має надійного резервного копіювання, записи оцінювання можуть бути втрачені. Частою проблемою є недостатня автономність роботи в автономному режимі, що ускладнює ведення журналу за відсутності зв'язку.

Організаційні недоліки: Впровадження електронного журналу вимагає інтенсивного навчання для викладачів та адміністрації. Багато співробітників звикли до паперових журналів, тому перехід зустрічає опір. Відсутність єдиного стандарту серед університетів означає, що кожен розробляє власну процедуру, що ускладнює обмін досвідом. Проблемою також залишається дублювання інформації: наприклад, паралельне ведення електронного журналу та внесення тих самих даних до централізованої системи (деканату) може призвести до дублювання роботи рисунок 2.2.

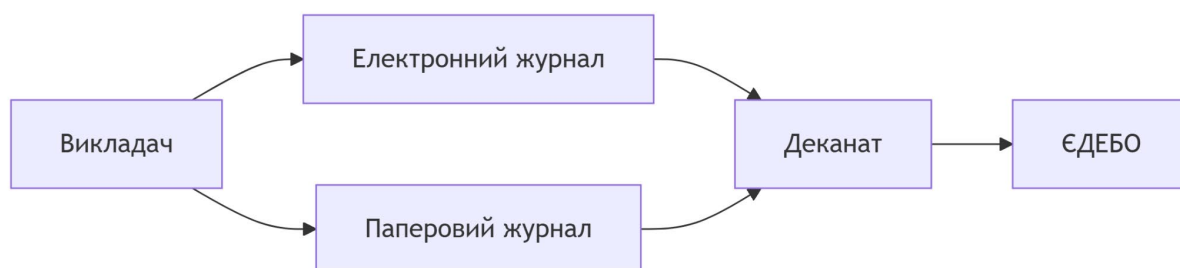


Рис. 2.2 Процес дублювання даних

Юридичні недоліки: Відповідно до Закону № 851-IV, електронні документи мають повну юридичну силу, а кваліфікований електронний підпис прирівнюється до власноручного [23]. Однак не всі платформи мають налагоджений механізм електронного підпису. Невідповідність формату інформації державним стандартам (через відсутність PDF-файлів, сформованих за необхідним шаблоном) призводить до труднощів з їх офіційним затвердженням. Наприклад, згідно з чинною інструкцією, електронний журнал повинен мати можливість генерувати дані у форматі, затвердженому Міністерством освіти і науки, але не всі системи це підтримують. Відсутність протоколів безпечного зберігання даних у хмарних сервісах може суперечити стандартам зберігання архівів документів.

Проблеми індивідуальних планів, модульно-рейтингової системи, практичних дисциплін. Індивідуальні навчальні плани: учні можуть мати різні набори курсів (факультативи, скорочена програма тощо). Більшість базових платформ не передбачають зручного налаштування індивідуальних маршрутів, тому кожному викладачеві доводиться вручну налаштовувати журнал для

конкретних студентів. Просунуті системи, такі як «PS-Журнал успішності» ЛНУ імені Тараса Шевченка, реалізують генерацію індивідуальних навчальних планів (ІНП) та їх автоматичний друк. У загальних LMS ця функція виконується через сторонні надбудови або скрипти, що збільшує складність адміністрування [25].

Модульно-рейтинговий підхід (MRS): Українська освітня політика передбачає MRS, де академічна успішність враховується у вигляді модульних балів і за шкалою, відмінною від традиційної (наприклад, 0-100 балів). Не всі системи підтримують автоматичну конвертацію в Європейську кредитно-трансферну систему (ECTS) або переведення оцінок в MRS. Наприклад, у старому «Електронному кампусі» КПІ формули підсумовування задавалися вручну а в Moodle доводиться встановлювати спеціальні плагіни для ECTS. Недостатня підтримка MRS призводить до помилок у підсумкових оцінках рисунок 2.3 [27].

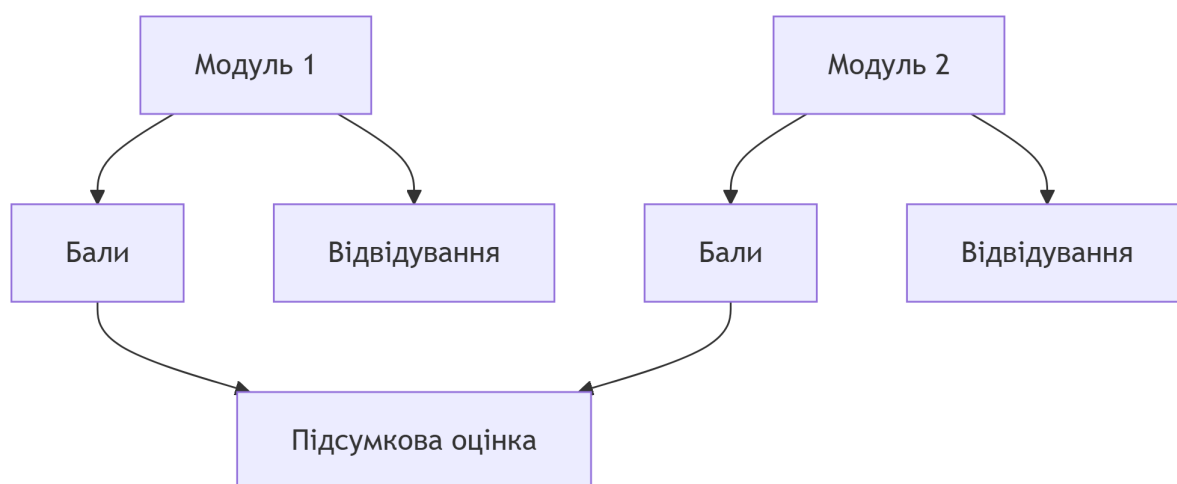


Рис. 2.3 Модульно-рейтингова система в е-журналі

Практичні та лабораторні курси: Такі курси часто вимагають окремих оцінок за практику та звіт, контролю відвідування тощо. Не всі журнали передбачають окремі поля для практичних оцінок або автоматичного обліку відвідуваності. Через це викладачам доводиться вести окремі таблиці або записи. У той же час, PS-Skill Log ЛНУ дозволяє розбивати журнал по заняттях окремо за предметами, що полегшує відстеження практичних робіт [25]. Для більшості систем це нестандартна функція, яка реалізується додатковими модулями або інтеграцією з навчальним планом.

Аналіз показав, що ключовими проблемними точками систем електронних журналів є фрагментарність рішень та відсутність єдиного стандарту даних. Недостатня інтеграція різних платформ призводить до втрати даних при переході з однієї системи на іншу [19]. Технічна нестабільність («падіння серверів», DDoS-атаки на шкільні сервіси тощо) ускладнює надійне використання таких систем. З точки зору користувача, часто бракує зручного мобільного доступу та інтуїтивно зрозумілих інтерфейсів – системи на базі LMS (Moodle, Canvas) орієнтовані на навчання в цілому, а не на щоденне відвідування та облік успішності. Для викладачів та адміністраторів це означає додатковий час на адаптацію та постійну підтримку. Юридично часто не виконується пункт про електронний підпис: система повинна генерувати документ у форматі, в якому можна поставити кваліфікований КЕП (або генерувати підписаний PDF самостійно). Ключові напрямки для покращення: розробка єдиного державного стандарту обміну даними журналів (щоб «журнали різних університетів розуміли один одного»), вдосконалення інтерфейсу відповідно до реальних потреб викладачів (наприклад, спрощення введення оцінок та відвідування), а також забезпечення повного юридичного циклу (формування інформації з КЕП та офіційних звітів) [19].

Варто рухатися в бік хмарної архітектури з відкритими API для полегшення інтеграції з системою деканату, ЄДЕБО та інструментами аналітики. Також рекомендується розширити підтримку індивідуальних навчальних планів та ІНПС на рівні платформи, щоб автоматизувати формування маршрутів студентів та зробити модуль Електронний журнал максимально адаптивним [19].

2.2 Дослідження потреб викладачів, адміністрації та студентів щодо створення та використання навчальної відомості

Навчальна відомість – офіційний документ для обліку успішності студентів – виконує багатофункціональну роль у системі вищої освіти. Вона поєднує відомості про відвідування занять, поточні й підсумкові оцінки, а також інші дані, що відображають навчальну діяльність студентів і викладачів. Згідно з

нормативними документами, електронний журнал (відповідник навчальної відомості) запроваджується з метою відкритого моніторингу та оперативного управління навчальним процесом, підвищення мотивації студентів, виявлення проблемних питань і покращення якості освіти. Він має забезпечувати централізований збір інформації про відвідуваність та успішність, аналітичну обробку даних і зворотний зв'язок між студентами, викладачами та адміністрацією. Проте реальні очікування й проблеми кожної групи користувачів можуть суттєво відрізнятися [22].

Викладачам потрібні зручні інструменти для швидкого та коректного виставлення оцінок та обліку відвідуваності студентів. Зокрема, важливими є автоматизовані розрахунки середнього балу та підсумкової оцінки, можливість гнучкого налаштування формул оцінювання, інтеграція з навчальними планами та базами даних студентів. У форматі документа (паперова форма або веб-інтерфейс) викладачі повинні мати інтуїтивно зрозумілий інтерфейс і мінімальне дублювання даних (щоб уникнути непотрібного введення оцінок у зошитах і подальшого перенесення в систему). Технічні очікування включають гарантію доступності системи (в тому числі мобільний або веб-доступ, можливість офлайн-роботи з подальшим синхронним оновленням) та коректне зберігання всіх даних із захистом від несанкціонованих змін. Викладачів також турбує відповідність форми транскрипту встановленим стандартам: наприклад, у багатьох університетах його структура визначається внутрішніми положеннями і має включати академічні кредити, форми контролю, рекомендовані шрифти тощо.

Однією з основних вимог є зменшення обсягу ручної роботи. Дослідження показують, що електронні журнали та системи оцінювання забезпечують зручність для викладачів та покращують комунікацію з батьками та учнями (можливість вчасно виставляти нові оцінки та коментарі) [3]. Однак при цьому викладачі стикаються з проблемою зайвих рутинних операцій: неконсолідовані системи або застаріле програмне забезпечення вимагають багато часу на перенесення даних, виправлення помилок, підрахунок балів вручну. Як зазначає Ф. Гаскі, хоча комп'ютеризовані журнали і полегшують облік, їхня математична точність не

гарантує чесності та справедливості в оцінюванні [17]. Для вирішення цих проблем необхідні інструменти, які б перевіряли правильність введених оцінок, інформували про невірні дані та надавали чіткі звіти для вчителя. Крім того, однією зі значних труднощів є недостатня підготовка викладачів до роботи з новим програмним забезпеченням: відсутність підготовки призводить до помилок і недовіри до системи [7]. Таким чином, викладачі очікують не лише функціонального наповнення системи, а й простоти використання та підтримки з боку технічної служби.

Таблиця 2.2

Вимоги викладачів та рішення питань

Проблема	Технічне рішення
Ручні розрахунки балів	Інтеграція формул оцінювання у систему
Дублювання даних	Синхронізація з електронними журналами
Недостатня підготовка	Інтерактивні tutorіали та чат-підтримка

Адміністрація факультету чи університету має потребу в узагальненні та аналізі академічних даних за різними критеріями. Основними функціями для цих користувачів є формування звітів про успішність груп, факультетів та навчального закладу в цілому, моніторинг відвідуваності та статистична оцінка навчальної роботи. Очікується, що формати даних будуть стандартизовані та формалізовані: інформаційна система університету має забезпечувати експорт у поширені формати (Excel, XML тощо) для передачі в державні реєстри (наприклад, Єдину державну електронну базу з питань освіти), а також автоматичне формування додатків до диплома з переліком дисциплін та балів. З технічної точки зору, адміністрація потребує централізованого електронного ресурсу з доступом до історичних даних, можливостями резервного копіювання та захисту даних. Від адміністрації вимагається дотримання державних стандартів бухгалтерського

обліку, електронний документообіг з накладними підписами та зберігання даних протягом багатьох років [9].

На практиці адміністрація часто стикається з фрагментарністю інформації (дані з різних підсистем не завжди пов'язані між собою) та неузгодженістю. На думку експертів, відсутність єдиного процесу управління даними спричиняє «труднощі у відстеженні та обробці даних про студентів через технічні проблеми, людські помилки та недостатню підготовку персоналу» [7]. Тому типові вимоги включають чітке призначення відповідальних за введення та оновлення записів, а також надання викладачам інструментів для введення оцінок і відвідування в режимі реального часу [7]. Це забезпечує своєчасність даних та мінімізує помилки. Потреба в оптимізації процесу створення інформації для адміністрації зумовлена можливістю автоматичного формування звітів будь-якої складності та зменшенням витрат на паперовий документообіг (як зазначається в одній замітці, раніше тисячі файлів зберігалися в підвалах університетів) [9].

Таблиця 2.3

Проблеми при роботі з навчальною відомістю

Проблема	Кого стосується	Наслідки
Відсутність інтеграції між модулями	Адміністрація, викладачі	Подвійне введення, помилки в даних
Інтерфейс	Викладачі, студенти	Складність роботи, зниження довіри до системи
Відсутність мобільного доступу	Студенти, викладачі	Затримки в оновленні, обмежений доступ
Відсутність юридичної валідації (ЕЦП)	Адміністрація	Неможливість використовувати відомість як офіційний документ
Високе навантаження на ІТ-персонал	Адміністрація	Затримки в обслуговуванні, збої систем

З точки зору студентів, академічна документація має бути прозорою та доступною. Основні очікування – можливість переглядати оцінки та коментарі без затримок, а також чіткі механізми оскарження у разі виявлення помилок. Студентам потрібен простий інтерфейс для перегляду своїх результатів: згідно з опитуваннями, студенти найбільше цінують чітку структуру системи, де легко знайти розділ зі своїми оцінками та завданнями [1]. Важливим моментом є забезпечення «зворотного зв'язку»: студенти повинні чітко бачити критерії оцінювання, свою середню статистику та отримувати сповіщення про оновлення в електронній системі [22, 1]. Формат навчальної інформації їм бажано отримувати в електронному вигляді (наприклад, у вигляді PDF-звіту або через особистий кабінет на навчальному порталі), і якщо можливо, з мобільним доступом. Студенти стаціонару покладаються на точність і достовірність інформації (коди дисциплін ЄКТС, кредити тощо), оскільки ці дані вносяться до їхніх офіційних документів про освіту та анкет. Недостатній та заплутаний доступ до інформації призводить до незадоволення: наприклад, фрагментація LMS та різні інтерфейси викладачів можуть вимагати від студентів запам'ятовування багатьох паролів та платформ [1]. Щоб вирішити ці проблеми, університети прагнуть запровадити єдині «електронні журнали» для всього університету, що гарантуватимуть однаковий досвід для студентів незалежно від факультету.

Варто зазначити, що в сучасних умовах студенти очікують можливості швидкого отримання академічних довідок та виписок. Наприклад, у США та Канаді впроваджено онлайн-сервіси (Parchment, MyCreds тощо), які дозволяють замовити офіційну виписку з оцінками в електронному вигляді [9, 11]. У Канаді платформа MyCreds забезпечує «швидкий і легкий доступ» до запиту та передачі транскриптів та інших документів, працює цілодобово і є офіційним каналом для їх перевірки [11]. Так само в європейському просторі програма Erasmus+ стимулює оцифрування академічних документів: транскрипт обов'язково має містити перелік прослуханих курсів, кредитів ECTS та балів для автоматизації визнання результатів у приймаючому закладі [13].

Сучасні студенти очікують, що системи управління навчанням (LMS, особисті кабінети) мають бути простими та інтуїтивно зрозумілими. Це стосується і «залікової книжки», адже її інтерфейс має бути зрозумілим: наявність зручних дашбордів, великих підказок та структурованих розділів полегшує пошук оцінок та завдань [1]. Більшість студентів – цифрові аборигени, і вони очікують швидкого доступу до своїх даних у будь-який час [1]. Негативними факторами є надмірна складність системи або дублювання записів у різних модулях порталу, що спричиняє плутанину та втрату часу. Таким чином, типові вимоги студентів до академічної інформації включають інтерактивні онлайн-платформи з простим інтерфейсом, єдиною системою для всіх дисциплін (щоб вони не мали декількох акаунтів) та мобільну доступність.

Таблиця 2.4

Ключові потреби різних груп користувачів

Категорія користувачів	Основні потреби	Формат/Функціональність
Викладачі	Швидке введення оцінок, автоматичні розрахунки, інтеграція з планами	Автоматичні підсумки, шаблони МОН, зручний UI
Адміністрація	Формування звітів, експорт у держреєстри, статистика	Централізоване зберігання, експорт у Excel/XML
Студенти	Прозорість, мобільний доступ, швидкий перегляд результатів	Онлайн-інтерфейс, нотифікації, доступ до PDF

В українських університетах, згідно з внутрішніми нормативними документами, електронні журнали (еквівалент залікових книжок) вже вважаються стандартом. Наприклад, в НТУ «ХАІ» затверджено «Положення про електронний журнал», в якому зазначено, що цей журнал є частиною системи автоматизації університету і містить дані про відвідування, поточні та підсумкові результати, а також про успішність викладачів. Система повинна генерувати як електронні, так і паперові форми звітності відповідно до вимог законодавства [22]. Водночас

практика показує, що деякі університети досі використовують змішаний підхід: викладачі заповнюють паперові форми, а потім вносять ці дані в електронні бази, що створює зайвий етап і помилки в транскрипції. Крім того, трапляються випадки, коли через відсутність єдиного стандарту в різних університетах викладачам доводиться звикати до різних форм і систем, що суперечить меті оптимізації. З точки зору нормативного регулювання, Україна сприяє діджиталізації процесів: нові накази Міністерства освіти і науки передбачають запровадження єдиного державного реєстру студентів та електронного документообігу. Однак технічні проблеми (брак грошей на модернізацію, кібератаки під час воєнного стану, недостатня ІТ-компетентність персоналу) гальмують цей процес.

У світі спостерігається тенденція до переведення документів про освіту в електронний формат та створення єдиних студентських порталів. В Європі, зокрема в країнах Болонської системи, запроваджено уніфіковані формати транскриптів (у тому числі додатку до диплома), що містять назви дисциплін, кількість кредитів ЄКТС та балів. Програми мобільності (наприклад, Erasmus+) підвищують вимоги до швидкого обміну цифровими транскриптами між університетами [13]. У США багато університетів використовують LMS (Blackboard, Canvas тощо) та інтегровані SIS, де «оцифрування» оцінок є нормою: це полегшує студентам і батькам доступ до даних про успішність (досвід і дослідження показали, що викладачі позитивно оцінюють можливості таких систем для комунікації з батьками) [3]. Онлайн-портали широко використовуються в Канаді: студенти можуть замовити офіційні транскрипти через такі системи, як MyCreds або Parchment, які автоматизують видачу та перевірку документів [11].

Міжнародний досвід демонструє, що успішні системи враховують потреби всіх груп: зручність введення даних для викладача (швидкий імпорт/експорт, автозаповнення), аналітика для адміністративного персоналу (формування динамічних звітів), прозорість для студента (онлайн-статус, сповіщення). Часто закордонні модулі містять готові шаблони форм відповідно до акредитаційних вимог та підтримують функції «what-if» для аналізу можливих оцінок. Загальні проблеми – відсутність уніфікації між різними платформами та недостатня увага

до UX-дизайну – вирішуються шляхом уніфікованих стандартів та додаткового навчання користувачів [3].

Таблиця 2.5

Порівняння міжнародних практик

Країна / регіон	Рішення / система	Ключові особливості
США	Canvas, Blackboard, Parchment	Інтеграція з LMS, онлайн-транскрипти, доступ батькам
Канада	MyCreds	Цифрові транскрипти, офіційна перевірка, доступ 24/7
ЄС / Erasmus+	Transcript of Records	Формат ECTS, електронна передача між університетами
Україна	Електронний журнал + ЄДЕБО	Часткова автоматизація, залежність від локальних впроваджень

Сформульовані вимоги до академічної документації базуються на трьох ключових категоріях: автоматизації та достовірності даних, прозорості інформаційних потоків та ергономічності інтерфейсів. Викладацький склад акцентує на необхідності інтуїтивно зрозумілого інструментарію для внесення оцінювальних даних, автоматизованого формування аналітичних витягів та дотримання нормативних вимог щодо звітності. Адміністративні структури вимагають централізованої системи архівації навчальної статистики з уніфікованими стандартами генерації звітів, що дозволить елімінувати ручну обробку масштабних датасетів. Студентський контингент очікує оперативного доступу до персональних академічних результатів та цифрового підтвердження навчальних досягнень.

Вирішення цих завдань передбачає імплементацію інтегрованого інформаційного середовища, що забезпечує консолідацію учасників освітнього процесу, автоматизацію обчислювальних процедур та валідацію вхідних даних. Подібна систематизація сприятиме оптимізації часових витрат, зниженню кількості

арифметичних помилок та підвищенню обґрунтованості управлінських рішень. Упровадження стандартизованого формату документації додатково забезпечить підвищення прозорості навчальних процедур, покращить індекс задоволеності користувачів від взаємодії з електронними ресурсами.

Отже, модернізація системи обліку академічних відомостей становить критичний етап трансформації вищої освіти, спрямований на досягнення операційної ефективності, підвищення рівня довіри до освітніх інституцій та забезпечення відповідності сучасним викликам цифрової епохи.

2.3 Технічні вимоги до розроблюваного модуля автоматичного генерування відомостей

Основною функціональною метою розроблюваного модуля є автоматична генерація навчальних відомостей для кожного студента з використанням даних існуючого електронного журналу (рис. 2.4). Модуль має надавати можливість формування відомості за весь період навчання або за вибраний навчальний рік. Генерація відомості здійснюється за запитом користувача через персональний кабінет студента у веб-інтерфейсі системи. Звіти повинні створюватись у форматах PDF або Excel і формуватись для конкретного студента у вигляді окремого файлу.

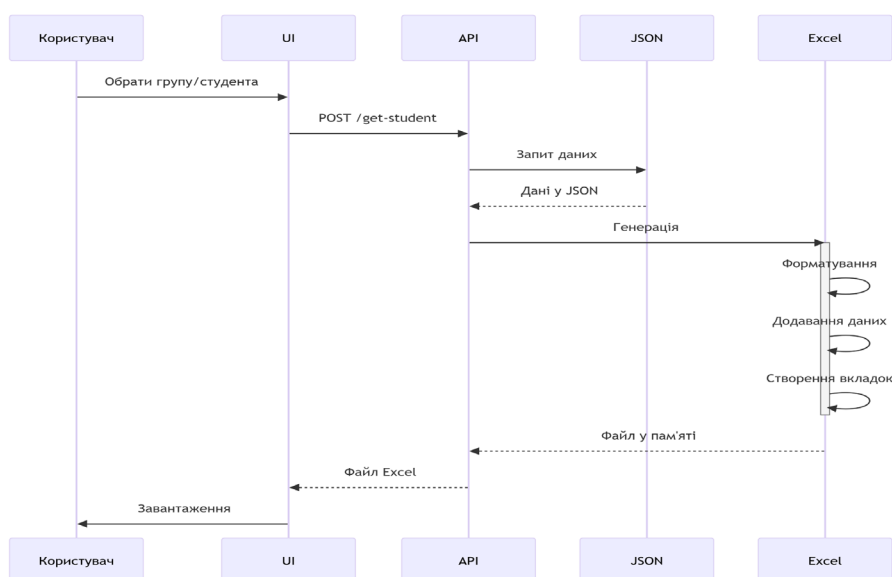


Рис. 2.4 Процес генерації відомості

Розроблений модуль автоматичного формування навчальної відомості відповідає низці ключових функціональних вимог, які забезпечують його ефективну роботу в контексті освітньої інформаційної системи. Насамперед, доступ до функцій модуля надається виключно автентифікованим студентам, які вже здійснили вхід до системи через особистий кабінет. Модуль не реалізує окремої системи реєстрації або авторизації, а використовує наявну сесію чи токен автентифікації, що гарантує єдиний захищений контур доступу до освітніх даних та унеможливорює несанкціоноване втручання у навчальну інформацію.

Користувачеві надається можливість через веб-інтерфейс самостійно обирати параметри формування звіту, зокрема навчальний рік, академічну групу, конкретного студента або бажаний формат вихідного документа. Завдяки цьому звіт формується з урахуванням індивідуальних запитів, що дає змогу зменшити обсяг оброблюваної інформації та уникнути дублювання результатів. Додатково реалізовано можливість динамічного оновлення даних шляхом використання елемента інтерфейсу «Перезавантажити БД», що дозволяє виводити актуальні списки студентів та груп без необхідності перезапуску модуля.

Формування навчальної відомості відбувається на основі вхідного JSON-файлу, який містить структуровані дані про результати навчання студентів – зокрема, інформацію про дисципліни, оцінки, сесії та семестри. На відміну від класичних підходів, система не здійснює прямого зчитування з бази даних, що суттєво спрощує впровадження рішення у різні освітні середовища без необхідності втручання у внутрішню структуру електронного журналу. Завдяки цьому модуль є незалежним і легко адаптується до зовнішніх джерел даних.

Після обробки вхідної інформації система здійснює генерацію відомості у форматі Excel за допомогою бібліотеки github.com/xuri/excelize/v2. Звіт має повноцінну структуру та включає інформацію про вивчені дисципліни, оцінки, підсумковий бал, а також загальний середній показник успішності. Студенту автоматично надається доступ до сформованого файлу, який він може зберегти на свій пристрій. У майбутньому передбачено розширення функціоналу модуля

шляхом додавання можливості експорту у формат PDF для забезпечення офіційного друку.

Особливістю модуля є те, що він генерує лише індивідуальні звіти для конкретного студента. Формування групових або агрегованих звітів не передбачено, що відповідає вимогам законодавства щодо захисту персональних даних та забезпечує високий рівень конфіденційності. Модуль безпосередньо вбудовується у вже наявний інтерфейс особистого кабінету студента, завдяки чому користувач після проходження авторизації має можливість негайно скористатися функціоналом генерації відомостей без потреби переходити до сторонніх інтерфейсів або систем. Такий підхід сприяє підвищенню зручності використання, інтеграції інформаційного середовища та оптимізації взаємодії з освітніми сервісами.

Модуль також характеризується можливістю багаторазового використання без необхідності перезапуску серверного середовища або оновлення вихідного коду. Це дає змогу використовувати його в повсякденній практиці як студентів, так і викладачів без додаткових витрат часу та ресурсів на технічну підтримку. Варто зазначити, що в межах чинної реалізації модуля не передбачено використання цифрового підпису або QR-коду у сформованих відомостях, хоча ці функції можуть бути реалізовані як частина майбутніх оновлень програмного забезпечення.

Нефункціональні вимоги до розробленого модуля визначають його якісні характеристики, що безпосередньо впливають на надійність, продуктивність, безпеку та зручність експлуатації у контексті електронної освітньої інфраструктури. Особливу увагу при формуванні цих вимог було зосереджено на забезпеченні стабільної роботи модуля за умов інтенсивного використання, зокрема при обробці великих обсягів даних або одночасному надходженні запитів від кількох користувачів. Система має бути стійкою до потенційних збоїв – наприклад, до помилок у структурі JSON-файлу, спроб обробки порожніх значень або звернень до неіснуючого об'єкта. Усі критично важливі процеси супроводжуються системним логуванням, що дає змогу своєчасно виявляти та

усувати причини збоїв у роботі. Відповідну візуалізацію архітектури помилок наведено на рисунку 2.5.

Завдяки використанню мови програмування Go у поєднанні з високопродуктивним вебфреймворком Fiber, модуль забезпечує високу швидкість обробки запитів, що дозволяє генерувати індивідуальну навчальну відомість у межах однієї секунди при звичайному рівні навантаження. Важливо, що архітектура системи розроблена з урахуванням майбутнього масштабування: за необхідності вона підтримує підключення додаткових обчислювальних ресурсів, розширення обсягу оброблюваних даних і збільшення кількості одночасних запитів без втрати продуктивності.

Питання інформаційної безпеки реалізоване через низку технічних рішень, серед яких – використання протоколу HTTPS для всіх HTTP-запитів, суворе обмеження можливості ручної модифікації запитів та повна авторизація користувача до початку формування навчальної відомості. Модуль працює виключно в межах вже автентифікованої сесії, що гарантує збереження конфіденційності освітніх даних. Захист логіки API від несанкціонованих дій забезпечує загальну цілісність системи. На рисунку 2.6 наведено загальну схему реалізації безпечної взаємодії з модулем.

Для забезпечення подальшої підтримки, тестування та розширення, система побудована за принципами модульної архітектури, де кожен функціональний блок реалізований як окремий компонент. Внутрішня структура проекту передбачає чітке розділення на логічні шари: контролери, сервіси, генератори, логери. Такий підхід спрощує процес локалізації помилок, забезпечує високу підтримуваність коду та дозволяє розширювати функціонал без ризику дестабілізації всієї системи.

Журналювання подій реалізовано як обов'язковий компонент внутрішньої логіки. Кожна дія користувача, починаючи з моменту входу до системи й закінчуючи формуванням фінального документа, фіксується у логах за допомогою відповідних бібліотек (наприклад, log, logrus або zap), що забезпечує аудит, моніторинг активності та можливість відтворення хронології взаємодії у разі потреби.

Сумісність модуля з наявним цифровим середовищем закладу вищої освіти була закладена ще на етапі проєктування. Модуль не потребує зміни структури існуючих баз даних або технологічних рішень, що використовуються в електронному журналі, й може бути інтегрований у вже функціональну систему без ризику порушення її стабільної роботи. Така адаптивність до різних архітектур забезпечує високу практичну цінність і універсальність запропонованого технічного рішення.

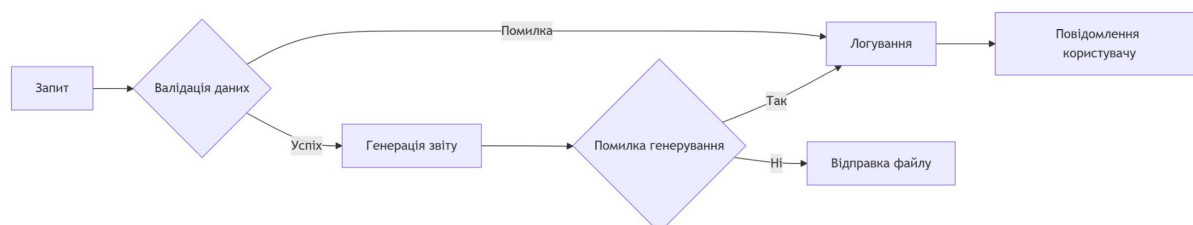


Рис. 2.5 Захист від помилок

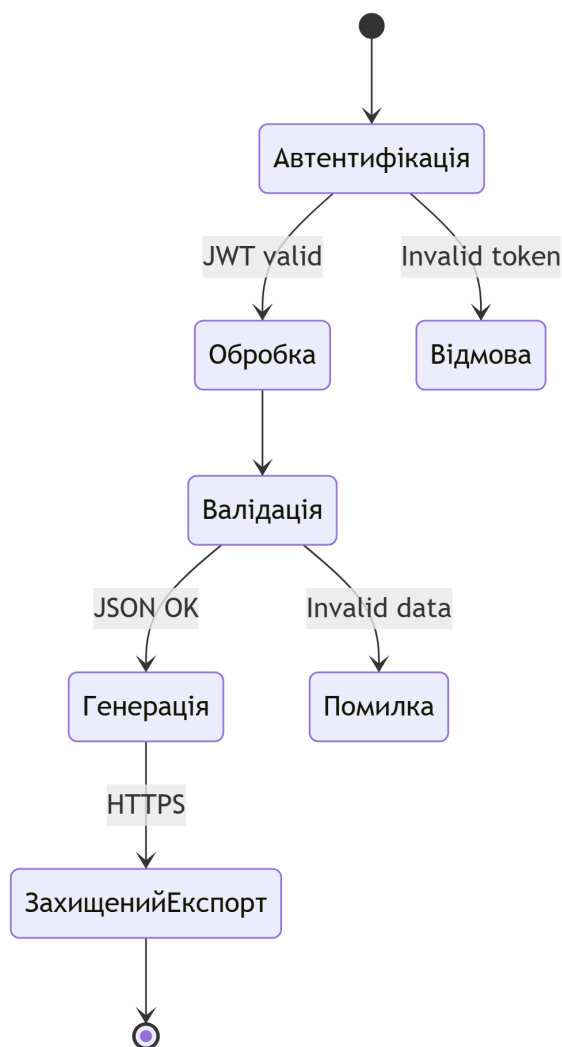


Рис. 2.6 Рівні безпеки даних

Вимоги до інтерфейсу розробленого модуля визначають принципи взаємодії користувача з програмним забезпеченням і передбачають створення простого, інтуїтивно зрозумілого та функціонально достатнього середовища для формування навчальної відомості. Передбачено, що модуль має бути забезпечений веб-інтерфейсом, доступним безпосередньо через особистий кабінет студента, без необхідності додаткової авторизації або переходу в інші зовнішні сервіси. Такий підхід дозволяє зберегти цілісність освітнього цифрового середовища та підвищує рівень зручності користування.

Особлива увага приділяється простоті використання: інтерфейс повинен містити лише необхідний набір елементів – таких як вибір навчального року, групи, студента та бажаного формату документа (Excel або PDF), супроводжуваних короткими текстовими підказками для користувача. Крім того, після ініціації запиту система має надавати зворотний зв'язок щодо його виконання у вигляді повідомлень про успіх або помилку. У разі виникнення помилок відображаються пояснення з рекомендаціями щодо подальших дій. З технічного боку інтерфейс повинен бути сумісним із сучасними браузерами, функціонувати через захищений протокол HTTPS, адаптуватися до різних розмірів екранів та не вимагати встановлення сторонніх компонентів чи плагінів.

Технологічна реалізація модуля базується на сучасних вебтехнологіях та засобах, що забезпечують ефективну та безпечну роботу системи. Модуль реалізовано мовою програмування Go (Golang), яка вирізняється високою продуктивністю, підтримкою паралелізму (goroutines), статичною типізацією та стабільністю у роботі. Завдяки цим особливостям Go є оптимальним вибором для побудови серверної частини REST API, що забезпечує комунікацію між інтерфейсом та логікою генерації звітів.

Серверна логіка реалізована за допомогою високопродуктивного фреймворку Fiber, що побудований на основі бібліотеки fasthttp. Він забезпечує ефективну маршрутизацію, обробку HTTP-запитів, підтримку middleware, зручну роботу з JSON і має високу швидкодію, необхідну для обробки множинних запитів у реальному часі. Формування навчальної відомості здійснюється за допомогою

бібліотеки Excelize, яка дозволяє створювати повноцінні Excel-файли формату .xlsx із підтримкою форматування, вставки формул, стилів, кількох аркушів та інших необхідних функцій. Кожен сформований документ містить детальну інформацію про навчальні дисципліни, оцінки, семестри, а також загальні показники успішності студента.

Уся вхідна інформація про успішність студентів імпортується у вигляді JSON-файлу. Структура файлу є чітко визначеною, що забезпечує уніфіковану обробку на сервері. Файл може завантажуватися вручну через веб-інтерфейс або заздалегідь розміщуватися в кореневій директорії проєкту. Користувацький інтерфейс реалізовано засобами HTML і CSS без використання додаткових JavaScript-фреймворків, що робить його легким, простим для підтримки і зручним в інтеграції. Інтерфейс має мінімалістичний дизайн з чіткою логічною структурою, яка включає вибір групи, студента, навчального року, формату документа та кнопки управління.

З метою забезпечення масштабованості та зручного розгортання модуль передбачено до контейнеризації за допомогою Docker. Це дозволяє легко розгортати рішення в інфраструктурі ЗВО, ізолювати середовище виконання, керувати оновленнями та масштабувати систему за потреби. Для збереження історії змін і спільної роботи над кодом використовується система контролю версій Git, яка також дозволяє інтегрувати модуль у CI/CD-процеси для автоматичного тестування, збирання та доставки оновлень на сервер. У цьому контексті можуть використовуватись такі інструменти, як GitHub Actions або GitLab CI.

Тестування логіки модуля здійснюється за допомогою вбудованих можливостей мови Go, зокрема пакету testing, який дозволяє створювати модульні тести, перевіряти цілісність логіки формування звітів, валідувати структуру JSON-файлів і коректність обробки HTTP-запитів. Конфігурація модуля здійснюється через зовнішні конфігураційні файли або змінні середовища (наприклад, .env), де зберігаються шляхи до файлів, порти, ключі доступу та інші параметри. Це забезпечує гнучкість налаштувань та адаптацію системи до різних середовищ розгортання (розробка, тестування, продуктивне середовище).

Усі події, пов'язані з функціонуванням системи – від авторизації користувача до формування документа – супроводжуються журналюванням. Для цього можуть використовуватись як стандартна бібліотека Go (log), так і сторонні засоби (logrus, zap). Ведення журналу подій дозволяє здійснювати аудит, виявляти збої, фіксувати ключові етапи взаємодії користувача з модулем і підтримувати загальну стабільність системи. Таким чином, комплексність технологічних рішень, що лежать в основі реалізації модуля, забезпечує його надійність, продуктивність і готовність до подальшого масштабування та впровадження в цифрову інфраструктуру закладів вищої освіти.

Таблиця 2.6

Технологічний стек

Компонент	Технологія	Версія
Мова програмування	Go	1.24
Веб-фреймворк	Fiber	v2.52.8
Робота з Excel	Excelize	v2.9.1
Логуювання	Вбудований log	-
Зберігання даних	JSON-файли	-

Сформовані технічні вимоги до модуля автоматизованого формування навчальної документації забезпечують його відповідність сучасним стандартам електронного документообігу в освітній сфері. Впровадження даного рішення на основі мови програмування Go дозволить досягти оптимальних показників продуктивності, масштабованості системи та стабільності її функціонування. Інтеграційний потенціал модуля, що передбачає його взаємодію з платформою електронного журналу та особистими кабінетами користувачів, забезпечує ергономічність та зручність експлуатації. Детально опрацьовані функціональні, нефункціональні, інтерфейсні та технологічні вимоги становлять методологічну основу для успішної імплементації розробленого модуля в діючу інформаційно-освітню інфраструктуру.

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ МОДУЛЮ ГЕНЕРУВАННЯ НАВЧАЛЬНОЇ ВІДОМОСТІ

3.1 Проєктування архітектури системи

Проєктована система реалізована за принципами багатошарової архітектури. Відповідно до поширеного підходу, у Go додатках виокремлюють рівні контролерів (інтерфейсів), сервісів (бізнес-логіки) та сховищ даних [14]. Як показано на рисунку 3.1, API Gateway виконує роль єдиної точки входу для зовнішніх запитів. Саме через нього користувачі звертаються до системи, а далі запит маршрутизується до внутрішніх компонентів. Згідно з патерном «API Gateway», цей модуль може також реалізовувати базову безпеку – наприклад, перевіряти авторизаційні токени [10]. Таким чином відбувається розділення відповідальностей: контролер оброблює HTTP-запит, сервісна логіка виконує перетворення даних і бізнес-алгоритми, а файлове сховище забезпечує довготривале збереження результатів. Такий підхід полегшує підтримку та розширення системи, адже кожний модуль ізольований за функціональністю [14].

Архітектура програмного модуля для автоматичного формування навчальної відомості реалізована у вигляді структурованої системи з низкою взаємодіючих компонентів, кожен із яких виконує специфічну функцію в межах загального технологічного процесу. Центральним елементом взаємодії між користувачем і системою виступає особистий кабінет, що реалізований як веб-інтерфейс для адміністраторів, викладачів або студентів. Саме через цей інтерфейс здійснюється завантаження вхідного JSON-файлу або введення даних вручну, а також ініціація запиту на формування звіту. Зв'язок особистого кабінету із серверною частиною відбувається через REST API, що забезпечує чітку маршрутизацію запитів до внутрішніх служб системи.

Вхідною точкою для всіх HTTP-запитів є API Gateway – універсальний шлюз, що перевіряє правильність запиту, за необхідності виконує аутентифікацію користувача та передає запит до відповідного модуля обробки. Дотримуючись

патерну API Gateway, цей компонент також може реалізовувати базові механізми безпеки, зокрема перевірку JWT-токена для контролю автентичності запитів [10]. Усі дії з перевірки облікових даних реалізуються в межах системи автентифікації, яка базується на механізмі JSON Web Token (JWT). Серверна частина на базі фреймворку Fiber використовує middleware для валідації токена, перевіряє його на чинність, і, у разі успішної перевірки, передає інформацію про користувача в контекст запиту. Якщо ж токен виявляється недійсним, система повертає повідомлення про помилку типу 401 Unauthorized [8]. Таким чином, доступ до сервісів отримують виключно авторизовані користувачі, що гарантує збереження цілісності та конфіденційності системи.

Обробка запиту реалізована через контролери – функціональні одиниці, які відповідають за обробку окремих REST-маршрутів. Вони отримують вхідні дані у форматі JSON, здійснюють первинну валідацію, перевіряючи наявність усіх необхідних полів, та ініціюють виклик відповідної бізнес-логіки. У цьому контексті використовується фреймворк Fiber, що відомий своєю високою продуктивністю та гнучкістю в налаштуванні middleware. Завдяки цьому контролери можуть легко поєднувати логіку перевірки, обробки запитів, логування та взаємодії з іншими модулями системи [8].

Основна обробка запиту відбувається в межах сервісу логіки – окремого програмного шару, що реалізує бізнес-логіку системи відповідно до принципів багаторівневої архітектури. Сервіс логіки приймає структуровані дані від контролера, виконує всі необхідні перевірки, обробку й трансформацію, координує взаємодію з генератором звітів і формує підсумкову відповідь, яка у разі помилки містить відповідне повідомлення для користувача. У рамках цієї логіки реалізовано також підрахунок підсумкових показників успішності, структурування таблиці звіту та узгодження форматів даних для виводу [14].

Генерація фінального документа відбувається у спеціалізованому модулі – генераторі звітів, який працює на основі бібліотеки Excelize. Цей компонент формує Excel-файл у форматі XLSX, додає відповідні аркуші, заповнює комірки, застосовує стилі та форматування, відповідно до параметрів, визначених сервісом

логіки. За потреби, можлива реалізація експорту в PDF через конвертацію створеного Excel-файлу або використання додаткових бібліотек. Однак основним форматом виводу наразі залишається Excel, що забезпечує найвищу гнучкість для роботи з даними [4].

Після успішного створення звіту, файл зберігається у файловому сховищі, яке може функціонувати як локальна директорія на сервері, так і як зовнішнє об'єктне сховище (наприклад, Amazon S3). Такий підхід дозволяє відокремити процес генерації документа від його зберігання, що сприяє гнучкості, довготривалому доступу до результатів генерації та зручному управлінню звітами.

Одним з важливих аспектів архітектури є реалізація системи логування, яка дозволяє відстежувати події в системі, фіксувати помилки та забезпечувати дієвий аудит взаємодій. Використання middleware Logger у Fiber дозволяє автоматично фіксувати всі HTTP-запити та відповіді, а також уможливорює розширене логування за допомогою додаткових бібліотек, таких як log, zap або logrus, для фіксації внутрішніх подій на рівні сервісної логіки.

Доповненням до функціоналу модуля є система автоматичного формування та відправлення електронних листів користувачам після завершення генерації звіту. Завдяки використанню стандартної бібліотеки Go (net/smtp) або сторонніх поштових сервісів, реалізується можливість надіслати студенту або викладачу повідомлення з прикріпленим документом або посиланням на його завантаження зі сховища. Така функція не лише покращує зручність користування системою, а й дозволяє реалізувати асинхронне інформування про завершення операцій, що сприяє підвищенню загального рівня інтерактивності цифрового освітнього середовища [8].

Кожен компонент архітектури виконує визначену роль у загальній логіці функціонування системи, забезпечуючи її цілісність, гнучкість, масштабованість та безпеку. Узгоджена робота всіх частин дозволяє досягти високої продуктивності та зручності як для кінцевого користувача, так і для технічного адміністратора системи.

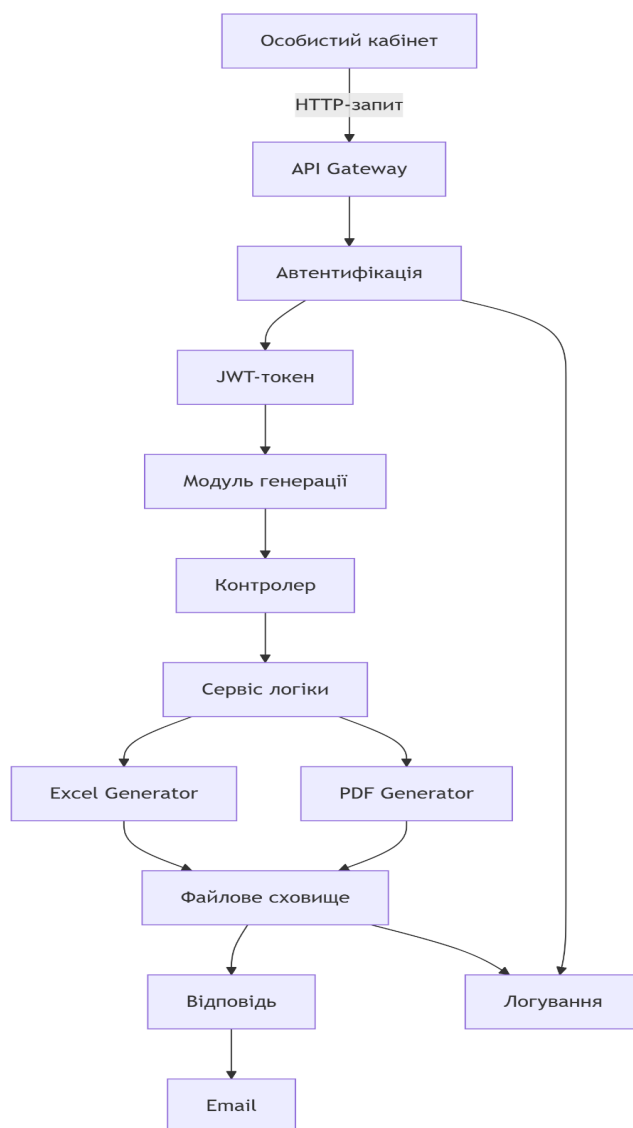


Рис. 3.1 Архітектура модуля

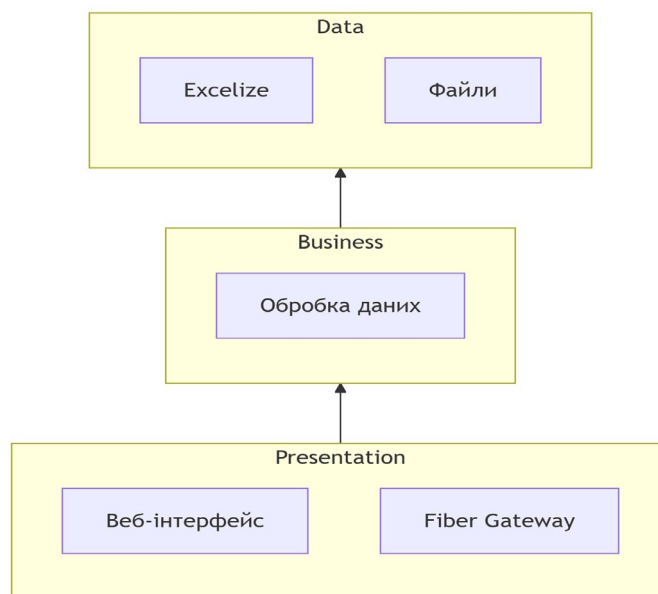


Рис. 3.2 Взаємодія між рівнями

Процес обробки запиту на формування навчальної відомості в межах розробленого модуля реалізується як послідовність логічно зв'язаних етапів, кожен із яких має власне функціональне призначення в архітектурі системи. На початковому етапі користувач, увійшовши до особистого кабінету, ініціює формування запиту шляхом завантаження вхідного JSON-файлу через відповідний елемент інтерфейсу, або ж шляхом ручного введення даних у форму. У відповідь браузер клієнта формує стандартний HTTP-запит, який надсилається на REST API системи.

Цей запит спочатку надходить до компонента API Gateway, який виконує роль центрального шлюзу системи та відповідає за первинну обробку всіх вхідних звернень. У межах цього етапу відбувається перевірка автентичності користувача – зокрема, здійснюється вилучення JWT-токена із заголовків HTTP-запиту та його подальша перевірка через middleware, що реалізований за допомогою засобів фреймворку Fiber. У випадку, якщо токен не проходить перевірку або є відсутнім, формується відповідь з кодом помилки 401 Unauthorized, і подальша обробка припиняється. У разі успішної автентифікації Gateway переспрямовує запит до відповідного контролера, що відповідає за обробку конкретного маршруту в рамках REST API [8].

На рівні контролера, який реалізований як HTTP handler у структурі вебзастосунку, відбувається зчитування JSON-даних із тіла запиту. Дані проходять первинну перевірку на синтаксичну коректність, зокрема парсинг у відповідні структури мови Go та перевірку наявності обов'язкових полів. У разі виявлення некоректних або неповних даних система повертає повідомлення про помилку валідації, що унеможлиблює подальшу обробку запиту.

Після успішної валідації контролер передає отриману структуру даних до відповідного сервісу бізнес-логіки. Цей сервіс виконує ключові обчислення: формує внутрішню модель звіту, обчислює середні оцінки, додає допоміжні елементи (наприклад, коментарі чи підсумкові результати) та адаптує структуру даних до вимог генератора звітів. Оброблені дані передаються далі до модуля генерації, де активується виклик функцій бібліотеки Excelize. У межах цього етапу

створюється новий файл Excel, додаються аркуші, заповнюються комірки значеннями, стилями та формулами відповідно до логіки звіту.

Після завершення формування, згенерований документ зберігається в тимчасовому сховищі або оперативній пам'яті, а далі записується у постійне файлове сховище. Це сховище може бути реалізоване як локальна директорія на сервері, або як зовнішній об'єктний репозиторій – наприклад, Amazon S3 чи інше хмарне середовище. Система фіксує шлях до сформованого файлу або створює URL-посилання, яке згодом може бути використане для завантаження документа.

Після збереження документа сервіс бізнес-логіки фіксує успішне виконання операції у логах системи, що дозволяє здійснювати аудит подій та діагностику потенційних проблем у разі їх виникнення. У межах цього ж етапу, за потреби, активується модуль надсилання електронних повідомлень: система формує email із повідомленням про готовність звіту, у якому міститься прикріплений документ або гіперпосилання на його завантаження зі сховища. Надсилання повідомлення відбувається автоматично, за допомогою SMTP-серверів або сторонніх API, у фоновому асинхронному режимі, не затримуючи основну логіку формування документа.

Завершальним етапом є повернення результату користувачу у вигляді HTTP-відповіді, що може містити безпосередній файл (наприклад, Excel у форматі XLSX) або перенаправлення на сторінку із посиланням для його завантаження. Паралельно із цим, у разі активації функції сповіщення, користувач отримує email із повідомленням про готовність звіту. Уся послідовність дій, від початкового запиту до отримання фінального документа, реалізується в рамках узгодженої багаторівневої архітектури, що забезпечує стабільність, масштабованість і захищеність системи.

У результаті проєктування архітектури було визначено ключові компоненти системи, їхні функції та взаємозв'язки, що дозволило сформувати надійну та масштабовану серверну інфраструктуру для генерації навчальних відомостей. Використання модульного підходу, REST API, механізмів автентифікації та сучасних бібліотек (Fiber, Excelize) забезпечує ефективну взаємодію між клієнтом

і сервером, а також відповідає вимогам продуктивності, безпеки та гнучкості подальшого розвитку системи.

3.2 Проєктування користувацького інтерфейсу системи

Інтерфейс системи надає веб-додаток, який авторизований студент сприймає як персоналізований робочий простір після входу. Основна мета інтерфейсу – забезпечити спрощене та інтуїтивно зрозуміле середовище для отримання власної навчальної інформації. Згідно з визначенням, інтерфейс повинен бути «зрозумілим, ефективним і приємним (зручним для оператора)» і мінімізувати зусилля користувача [2]. Іншими словами, кожен елемент сторінки виконує певну функцію, пов'язану з роботою студента обліку та інформації.

До основних елементів користувацького інтерфейсу, що реалізуються в межах функціоналу розробленого модуля, належить низка інтерактивних компонентів, які забезпечують зручну взаємодію з системою та повноцінне виконання усіх етапів формування навчальної відомості. Центральне місце в інтерфейсі займає поле для завантаження JSON-файлу, реалізоване у вигляді HTML-елемента типу «file» із супровідною кнопкою, яка ініціює процес імпорту. Цей елемент дозволяє студенту або викладачу імпортувати початкові дані – наприклад, інформацію про академічний прогрес – безпосередньо в систему. Після вибору відповідного файлу активується REST API-запит методом POST, який передає файл до серверного модуля для подальшої обробки, валідації та збереження в базі даних або тимчасовому сховищі.

Після успішного завантаження даних в інтерфейсі з'являється можливість обрати навчальну групу за допомогою випадającego списку, який динамічно відображає актуальні дані, отримані з REST API у відповідь на запит про всі наявні групи. Такий селектор груп виконує функцію фільтрації, обмежуючи перелік доступних студентів до тих, що належать до конкретної академічної групи. Подальша взаємодія з інтерфейсом передбачає використання селектора студента, який також реалізовано у вигляді динамічного випадającego списку. Після обрання

конкретної групи система надсилає асинхронний запит (AJAX, REST API методом GET), у відповідь на який сервер повертає перелік студентів, що належать до цієї групи. Таким чином, користувач має змогу вибрати конкретного студента для формування індивідуальної навчальної відомості.

З метою забезпечення актуальності даних у селекторах інтерфейс містить кнопку «Оновити базу даних», яка виконує функцію ініціації повторного запиту до серверного модуля синхронізації. Після її натискання відбувається перерахунок збережених даних або повторний імпорт JSON-файлу, внаслідок чого оновлюється структура бази даних і відповідно – вміст випадajuчих списків. Такий механізм дозволяє своєчасно відображати зміни, що можуть мати місце в реальних навчальних процесах.

Завершальним етапом у взаємодії користувача із системою є натискання кнопки «Завантажити відомість», яка активує формування персоналізованого документа для вибраного студента. У результаті цієї дії відправляється відповідний REST API-запит (наприклад, GET-запит із параметром `student_id`), і сервер повертає згенерований файл у визначеному форматі – зазвичай Excel або PDF. Отриманий документ може бути негайно завантажений користувачем та використаний для подальших освітніх чи адміністративних потреб. Така архітектура інтерфейсу поєднує в собі простоту, адаптивність до потреб користувача та тісну інтеграцію з логікою серверної частини модуля.

Принципи проектування інтерфейсу. При проектуванні інтерфейсу важливо забезпечити чіткість і зручність – підписи до полів, кнопок і повідомлень викладені прозоро, щоб студент міг легко зрозуміти, які дії необхідно виконати [15]. Використовується мінімалістичний дизайн: на екрані розміщуються тільки необхідні елементи (завантаження файлів, селектори, кнопки), що знижує когнітивне навантаження на користувача. Зворотній зв'язок також надається: після завантаження файлу, оновлення бази даних або завантаження інформації користувачеві відображається повідомлення про успіх або помилку, при завантаженні може з'явитися індикатор процесу. Ці методи відповідають загальним

принципам дизайну інтерфейсу, де чіткість, послідовність та зворотний зв'язок підвищують ефективність сприйняття та виконання завдань (рис. 3.3) [15].

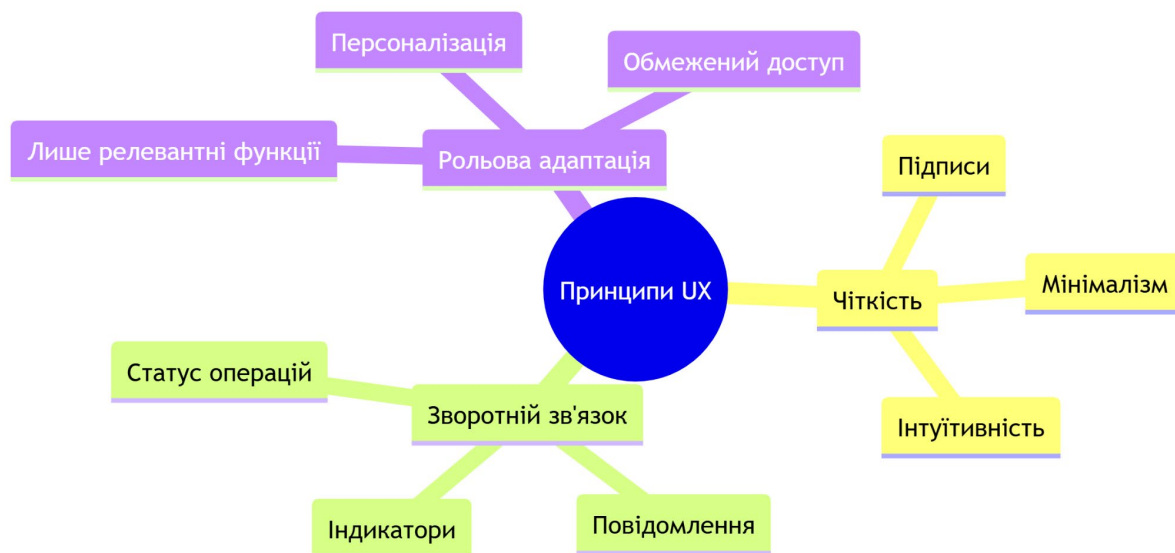


Рис. 3.3 принцип побудови UX

Особлива увага приділяється рольовій адаптації. Інтерфейс призначений для студента, тому після авторизації він бачить тільки елементи, пов'язані з особистим навчальним планом. Тут відсутні функції вчителя або адміністратора (наприклад, редагування оцінок інших учнів) [6]. Такий підхід забезпечує орієнтованість на користувача: студент не відволікається на непотрібні варіанти, а вся інформація надається «з одного вікна». Надалі для інших ролей (наприклад, вчителя або адміністратора) можуть бути реалізовані окремі інтерфейси з додатковими селекторами і кнопками.

Інтеграція з системою електронного журналу. Всі дії в інтерфейсі засновані на REST запитах до серверних модулів (рис. 3.4), що забезпечує високу модульність і масштабованість рішення. Як зазначають у практиці сучасних систем, API дозволяють використовувати сторонні додатки спільно зі студентською інформаційною системою та обмінюватися даними між ними. Таким чином, розроблений інтерфейс буде легко включений як окремий сервіс або компонент в загальну систему електронного журналу: наприклад, через внутрішній API дані імпортуються/експортуються з єдиної бази даних студентських записів. Стиль дизайну та принципи навігації можуть бути синхронізовані з інтерфейсом

електронної системи журналів (загальні кольори, шрифти, меню), а аутентифікація може бути реалізована через єдину систему (SSO) університету.

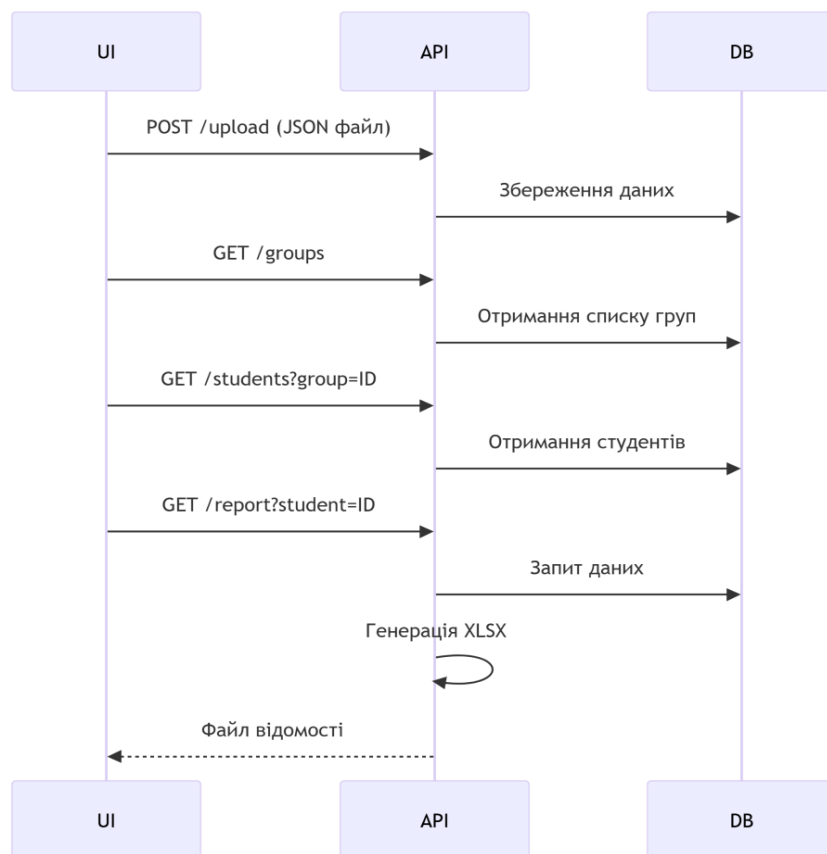


Рис. 3.4 Схема REST-запитів

Таблиця 3.1

Послідовність дій користувача та взаємозв'язок з функціональними модулями

Елемент інтерфейсу	Дія та призначення	Серверний модуль / API
Поле завантаження JSON-файлу	Імпорт вихідних даних в систему (REST POST /api/import)	Модуль імпорту/обробки даних
Селектор групи	Вибір групи для фільтрації студентів (REST GET /api/groups)	Модуль управління переліками груп
Селектор студента	Вибір конкретного студента (REST GET /api/students?group={groupId})	Модуль управління переліком студентів
Кнопка «Оновити БД»	Оновлення даних у сховищі (REST GET /api/refresh)	Модуль синхронізації/бази даних
Кнопка «Завантажити відомість»	Генерація і завантаження файлу персональної відомості (REST GET /api/report?student={id})	Модуль генерації звітів / звіти

3.3 Реалізація програмних модулів генерування відомостей

У межах розробки програмного модуля для автоматичного формування навчальних відомостей студентів була реалізована серверна частина, що працює з локальними структурованими даними у форматі JSON. Основна мета полягає у забезпеченні можливості обробки та візуалізації академічної інформації на основі оцінок студентів з різних форм контролю (екзамени, заліки, додаткові сесії тощо) та формуванні табличного документа з результатами у форматі Excel.

Для подання і обробки даних застосовано спеціалізовані типи даних. Зокрема, структура Record (рис. 3.5) використовується для збереження інформації про результат окремого студента за певною дисципліною, включаючи ідентифікатор предмета, номер залікової книжки, оцінку у форматі ECTS, загальний бал і наявність підпису викладача. Це дозволяє гнучко оперувати даними для подальшого формування таблиці.

```
type Record struct {
    ItemID      int    `json:"item_id"`
    RecordBookID int   `json:"record_book_id"`
    ResultECTS  string `json:"result_ects"`
    ResultGeneral int  `json:"result_general"`
    Signed      bool  `json:"signed"`
}
```

Рис. 3.5 Інформація про студента з БД

Опис дисципліни та її оцінювання реалізовано через тип Item, що містить поля з інформацією про накопичувальні, екзаменаційні та загальні бали, а також назву дисципліни (Рисунок 3.6).

```
type Item struct {
    StartPoints int    `json:"start_points"`
    ExamPoints  int    `json:"exam_points"`
    TotalPoints int    `json:"total_points"`
    Subject     string `json:"subject"`
}
```

Рис. 3.6 Запис предмета та його оцінювання

Для об'єднання інформації в межах групи студентів застосовується тип GroupData, який групує записи оцінювання за кількома формами контролю (основна сесія, додаткова, екзамен, додатковий екзамен) у вигляді мап типу map[string][]Record. Це забезпечує ефективний доступ до інформації по кожному студенту.

```

type GroupData struct {
    Exam          map[string][]Record `json:"exam"`
    AdditionalExam map[string][]Record `json:"additional_exam"`
    Term          map[string][]Record `json:"term"`
    AdditionalTerm map[string][]Record `json:"additional_term"`
    Items         map[string]Item      `json:"items"`
}

```

Рис. 3.7 GroupData тримає в собі дані про групу та студентів

Сукупність усіх груп об'єднується у структуру Data, що є загальним контейнером для оперативної пам'яті рисунок 3.8.

```

type Data struct {
    Groups map[string]GroupData
}

```

Рис. 3.8 Сукупність усіх груп

Для завантаження інформації з JSON-файлів у пам'ять реалізована функція LoadDataFromFolder(folder string) (рисунок 3.9). Вона здійснює пошук усіх файлів з розширенням .json у вказаному каталозі, зчитує їх вміст і за допомогою функції json.Unmarshal() перетворює в структуровані об'єкти GroupData. Назва файлу використовується як ідентифікатор групи. Це дозволяє не використовувати зовнішні бази даних, а працювати виключно з локальними файлами, що підвищує автономність модуля.

```

func LoadDataFromFolder(folder string) (*Data, error) {
    files, err := filepath.Glob(filepath.Join(folder, "*.json"))
    if err != nil {
        return nil, err
    }

    data := &Data{Groups: make(map[string]GroupData)}

    for _, file := range files {
        bytes, err := os.ReadFile(file)
        if err != nil {
            log.Printf("failed reading %s: %v", file, err)
            continue
        }

        var gd GroupData
        if err := json.Unmarshal(bytes, &gd); err != nil {
            log.Printf("failed parsing %s: %v", file, err)
            continue
        }

        // extract group name from filename
        base := filepath.Base(file)
        group := strings.TrimSuffix(base, filepath.Ext(base))
        data.Groups[group] = gd
    }

    return data, nil
}

```

Рис. 3.9 Завантаження інформації

Для внутрішнього використання при побудові таблиці Excel застосовується утиліта ToAlpha(col int), яка виконує перетворення індексації стовпців у буквенний формат, прийнятний для Excel-документів (рис. 3.10).

```
func ToAlpha(col int) string {
    result := ""
    for col > 0 {
        col-- // Excel columns are 1-based
        result = string(rune('A'+(col%26))) + result
        col /= 26
    }
    return result
}
```

Рис 3.10 Використання таблиці Excel

Серверна частина застосунку розгортається за допомогою фреймворку Fiber. Під час ініціалізації (fiber.New()) задається обробка статичних ресурсів і визначаються маршрути для обробки HTTP-запитів. (рисунк 3.11)

```
app := fiber.New()
app.Static("/", "./frontend")
app.Post("/data", handleUpload)
app.Get("/reload", handleReload)
app.Get("/get-groups", handleGetGroups)
app.Get("/get-students", handleGetStudents)
app.Get("/get-student", handleGetStudent)
log.Fatal(app.Listen(":3000"))
}
```

Рис. 3.11 Запуск бекенду

Основна функція генерації навчальної відомості реалізована у вигляді методу createStudentExcel(group, name string) (рисунк 3.12). У процесі виконання функції відбувається формування аркушів Excel, кожен з яких відповідає певному виду контролю: екзаменаційна відомість, додаткова екзаменаційна, сесія, додаткова сесія. Всі оцінки для вибраного студента фільтруються із загальної структури GroupData, після чого послідовно записуються у таблицю Excel з відповідними заголовками. Для форматування клітинок застосовуються стилі (рамки, центрування), що забезпечує офіційний вигляд звітнього документа.

```
func (d *Data) createStudentExcel(group, name string) (*excelize.File, error) {
    gd, ex, ax, tr, atr, err := d.GetStudent(group, name)
    if err != nil {
        return nil, err
    }
}
```

Рис. 3.12 Функція генерації навчальних відомостей

Фрагмент коду що створює масив об'єктів SheetRecord (рисунок 3.13), кожен із яких містить назву аркуша Excel та відповідні записи оцінок, які будуть виводитись на ньому.

```
records := []SheetRecord{
    {"Екзаменаційна", ex},
    {"Додаткова екзаменаційна", ax},
    {"Сесія", tr},
    {"Додаткова сесія", atr},
}
```

Рис. 3.13 Масив об'єктів SheetRecord

Фрагмент на рисунку 3.14 реалізує запис назви предмета, кількості балів та оцінок до таблиці. Кожен рядок Excel відповідає одному предмету, а кожна колонка – певному параметру (наприклад, колонка A – назва предмета, E – загальний бал, F – оцінка у форматі ECTS).

```
for _, rec := range sr.Records {
    item, exists := items[fmt.Sprintf(rec.ItemID)]
    if !exists {
        continue
    }

    cols := []interface{}{
        rec.ItemID,
        name,
        rec.RecordBookID,
        item.StartPoints,
        item.ExamPoints,
        item.TotalPoints,
        rec.ResultECTS,
        rec.ResultGeneral,
        rec.Signed,
    }

    for c, v := range cols {
        cell := fmt.Sprintf("%s%d", ToAlpha(c+1), row)
        f.SetCellValue(sheet, cell, v)
        f.SetCellStyle(sheet, cell, cell, style)
    }
    row++
}

specRow := row + 2
f.SetCellValue(sheet, "A"+strconv.Itoa(specRow), "ID предмету")
f.SetCellValue(sheet, "B"+strconv.Itoa(specRow), "Назва предмету")
f.SetCellStyle(sheet, "A"+strconv.Itoa(specRow), "B"+strconv.Itoa(specRow), style)
```

Рис. 3.14 Загальна таблиця

Після завершення генерації документ передається користувачу у вигляді байтового потоку через HTTP-відповідь. Для цього у заголовках відповіді встановлюється Content-Disposition: attachment, що сигналізує браузеру про необхідність завантаження файлу (рис. 3.15).

```
c.Set("Content-Type", "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet")
c.Set("Content-Disposition", fmt.Sprintf("attachment; filename=%s-%s.xlsx", group, strings.ReplaceAll(student, " ", "_")))
return c.Send(buf.Bytes())
```

Рис. 3.15 Задає формат даних та тип як їх відправити

Архітектурно рішення відзначається чітким розподілом відповідальностей: модуль даних оперує лише структурами і не містить бізнес-логіки, утилітарні функції реалізують повторно використововану логіку (читання файлів, обробка форматів), а основний сервер координує взаємодію між модулями. Такий підхід відповідає принципам модульності та дозволяє легко розширювати систему. Наприклад, у майбутньому можна додати функцію створення PDF-документів, підключення до централізованої БД чи API ЄДЕБО, або ж реалізацію цифрового підпису електронної відомості.

3.4 Тестування розробленого модуля та перспективи впровадження

На етапі тестування модуля перевіряється його відповідність функціональним, технічним та експлуатаційним вимогам, визначеним під час проєктування системи. Тестування дозволяє виявити можливі недоліки у роботі програмного забезпечення, оцінити стабільність, ефективність та зручність взаємодії користувача з інтерфейсом. Особливої важливості цей етап набуває у випадку освітніх інформаційних систем, де точність та надійність даних мають принципове значення. У межах даного підрозділу розглядаються результати функціонального тестування модуля автоматичного формування навчальної відомості, а також перспективи його впровадження у практику цифрового документообігу закладів вищої освіти.

Перехід за локальним маршрутом `http://localhost:3000/` успішно відкривав головну сторінку модуля (рис. 3.16). Перевірено коректне відображення всіх елементів: селекторів, кнопок, інформаційних блоків.

Відомості студентів

Рис. 3.16 Головна сторінка модуля

Після завантаження інтерфейсу було протестовано функціонал вибору академічної групи. Список груп генерується динамічно – через аналіз структури завантаженого JSON-файлу або при натисканні кнопки оновлення бази даних. Відображення списку відбувається у вигляді випадаючого меню (рис 3.17).

Відомості студентів

Рис. 3.17 Обрання академічної групи студента

Після обрання групи система автоматично оновлює селектор студентів. Було протестовано динамічне завантаження списку студентів та правильність відображення ПІБ. Студент обирає власне ім'я зі списку для формування персоналізованої відомості. Усі студенти, присутні у JSON-файлі, коректно виводилися (рис. 3.18).

Відомості студентів

Рис. 3.18 Обрання ПІБ студента

Головною функцією модуля є генерація персоналізованої навчальної відомості студента. Після вибору групи та студента, користувач натискає кнопку «Завантажити відомість» рисунок 3.19, що ініціює REST-запит до серверної частини. Сервер, у свою чергу, опрацьовує дані, формує звіт у форматі Excel за допомогою бібліотеки `excelize/v2`, і повертає його на клієнтську частину.

Було перевірено:

- коректність структури Excel-документа (наявність заголовків, даних, підписів)
- відображення оцінок, предметів, семестру, середнього балу
- швидкість генерації (у середньому < 1 секунди)
- коректне завантаження файлу на пристрій користувача (рис. 3.20)

Відомості студентів

Завантажити загальну відомість

Виберіть один або декілька .json:

Вибрати файли

Файл не вибрано

Виберіть один або декілька .json

Завантажити

Виберіть групу:

PD42

▼

↺

Перезавантажити БД

Виберіть студента:

Anzhelika V.V.

▼

Завантажити відомість

Рис. 3.19 Кнопка Завантажити відомість

A	B	C	D	E	F	G	H	I
ID предмету	ПІБ	№ залікової	Початкові бали			цінка з есе	цінка загал	Підписано
1	Anzhelika V.V.	1234	50	75	100	A	100	ИСТИНА
1	Anzhelika V.V.	1234	50	75	100	A	100	ИСТИНА
ID предмету	Назва предмету							
1	testing							
2	testing2							

Рис. 3.20 Відомість успішності студента

Також було протестовано можливість формування відомості з іншого семестру.

Екзаменаційна	Додаткова екзаменаційна	Сесія	Додаткова сесія
---------------	-------------------------	-------	-----------------

Рисунок 3.21

Проведене тестування підтвердило працездатність, функціональну повноту та надійність розробленого модуля. Усі ключові сценарії взаємодії користувача

були реалізовані відповідно до вимог, а результати засвідчили стабільну роботу системи, коректну обробку даних та зручність використання інтерфейсу. Отримані результати свідчать про готовність модуля до впровадження в освітнє середовище.

Перспективи впровадження. Розроблений модуль має високий потенціал для впровадження у внутрішні освітні платформи закладів вищої освіти. Його архітектура передбачає інтеграцію в електронний журнал успішності – зокрема, як окремий сервіс, доступний студентам через особистий кабінет. Завдяки використанню сучасних вебтехнологій та принципів мікросервісної архітектури, модуль може функціонувати як автономний компонент або бути включений до складу комплексної інформаційної системи навчального закладу. Основні переваги впровадження:

- Автоматизація академічного документообігу. Впровадження модуля забезпечує суттєве скорочення часу на підготовку звітної документації, зменшення кількості ручних операцій, а також знижує ймовірність помилок під час заповнення документів завдяки перевірці введених даних та шаблонізації.
- Персоналізація даних. Система динамічно формує документи відповідно до індивідуального навчального плану студента, з урахуванням дисциплін, терміну навчання, результатів атестації та інших параметрів. Це створює умови для гнучкого та точного відображення академічної інформації.
- Швидке розгортання. Завдяки застосуванню мови програмування Go, яка забезпечує високу продуктивність і низьке споживання ресурсів, а також використанню контейнеризації (Docker), модуль може бути швидко та безболісно розгорнутий у більшості сучасних серверних середовищ без необхідності значних модифікацій або адаптацій.
- Масштабованість. Рішення може бути адаптоване як для використання в межах одного закладу вищої освіти, так і для масштабування до міжуніверситетського або загальнонаціонального рівня. Це відкриває перспективи для його використання в централізованих освітніх системах, наприклад, для формування державної статистичної звітності або аналізу навчальної успішності на рівні регіону чи країни.

- Відкритість до розширення. Програмна структура модуля дозволяє подальше нарощування функціональності – зокрема, додавання механізмів електронного підпису, генерації QR-кодів для перевірки справжності документа, інтеграції з зовнішніми сервісами (такими як ЄДЕБО, системи аналітики або архівного зберігання). Крім того, може бути реалізовано API для доступу до модуля з інших систем.

Передбачені сценарії впровадження:

- Інтеграція з модулем «Сесія» в електронному журналі – з можливістю автоматичного імпорту результатів контролю знань.
- Автоматичне формування звіту на запит студента – з використанням авторизації та обмеженням за періодом (наприклад, після завершення сесії або навчального року).
- Генерація файлів у форматі PDF/Excel для подання до інших установ – наприклад, при переведенні до іншого ЗВО, участі в програмах академічної мобільності, отриманні стипендії чи оформленні академічної відпустки.
- Створення офіційних виписок, які згодом можуть бути прирівняні до академічних довідок, що підтверджують успішність навчання.
- Можливість адаптації інтерфейсу відповідно до корпоративного стилю навчального закладу або національного дизайну державних цифрових сервісів.

Враховуючи сучасні тенденції цифровізації освіти, активне впровадження цифрового документообігу, а також зростаючі вимоги до оперативності та прозорості в обліку навчальних досягнень, запровадження подібного модуля є актуальним, обґрунтованим і перспективним кроком. З технічної точки зору, модуль відповідає вимогам масштабованості, безпеки та доступності. З правової точки зору, його функціонування узгоджується з положеннями чинного законодавства України, зокрема Закону України «Про електронні документи» та Закону України «Про електронний документообіг», що створює належне нормативне підґрунтя для його впровадження в офіційний документообіг закладів вищої освіти [24].

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено та апробовано програмний модуль автоматичного формування навчальної відомості студента на основі даних електронного журналу успішності, що реалізує сучасний підхід до автоматизації освітнього документообігу у закладах вищої освіти. Робота охопила повний цикл системного проектування – від аналізу існуючих рішень і виявлення їх функціональних недоліків до створення ефективного технічного рішення з урахуванням вимог до стабільності, масштабованості та простоти інтеграції в інформаційну інфраструктуру університету. Серверна частина модуля реалізована мовою Go із використанням високопродуктивного вебфреймворку Fiber, а інтерфейсна частина – за допомогою HTML, що забезпечує доступність і простоту взаємодії для кінцевого користувача.

У результаті виконано всебічний огляд сучасних електронних платформ обліку успішності, що використовуються як в Україні, так і за кордоном. Було виявлено основні труднощі, пов'язані з формуванням навчальних відомостей, серед яких – відсутність уніфікованих модулів, складність генерації персоналізованих документів та недостатня підтримка державних освітніх стандартів. На основі цього сформульовано вимоги до функціоналу і архітектури нового модуля, визначено технічні засоби реалізації, а також логіку його взаємодії з користувачем та серверною частиною.

У межах практичної реалізації модуль було спроектовано як універсальне розширення для електронного журналу, здатне обробляти вхідні дані у форматі JSON, формувати структуровану навчальну відомість у форматі Excel і надавати користувачеві можливість її завантаження. Розробка включає підтримку динамічного вибору параметрів запиту (група, студент, рік навчання), стабільну обробку помилок, механізми логування дій та відповідність сучасним вимогам до безпеки даних. Тестування модуля показало його працездатність у реальному середовищі, стійкість до збоїв, оперативність обробки запитів та зручність користування інтерфейсом.

Практична значущість розробки полягає у спрощенні та прискоренні процесу формування офіційних звітних документів, зменшенні навантаження на викладачів та адміністративний персонал, а також у підвищенні точності та прозорості освітнього процесу. Модуль не потребує модифікації структури баз даних електронного журналу, легко інтегрується з наявним інтерфейсом особистого кабінету студента та може бути масштабований до потреб конкретного ЗВО. Запропоноване рішення є перспективним для подальшого розвитку – зокрема, у напрямку реалізації експорту у формат PDF, впровадження електронного підпису, а також глибшої інтеграції з державними інформаційними системами, такими як ЄДЕБО та АС «Деканат».

Результати дослідження підтверджують доцільність застосування автоматизованих підходів до формування навчальної звітності у вищій освіті, а створений модуль становить вагомий крок до підвищення цифрової зрілості навчальних закладів і вдосконалення академічного документообігу в умовах сучасних викликів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Choosing an LMS: the student perspective. eChalk. URL: <https://www.echalk.com/blog/2017/2/13/choosing-an-lms-the-student-perspective> (дата звернення: 30.05.2025).
2. Contributors to Wikimedia projects. User interface - Wikipedia. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/User_interface (дата звернення: 30.05.2025).
3. ERIC - ED532247 - teachers' perceptions of the impact of online grading systems, proquest LLC, 2011. ERIC - Education Resources Information Center. URL: <https://eric.ed.gov/?id=ED532247> (дата звернення: 30.05.2025).
4. GitHub - qax-os/excelize: Go language library for reading and writing Microsoft Excel™ (XLAM / XLSM / XLSX / XLTM / XLTX) spreadsheets. GitHub. URL: <https://github.com/qax-os/excelize> (дата звернення: 30.05.2025).
5. Google AppSheet | Build apps with no code. AppSheet. URL: <https://about.appsheet.com/home/> (дата звернення: 30.05.2025).
6. How apis work with a student information system. Campus Cafe. URL: <https://campuscafesoftware.com/how-apis-work-with-an-sis-integrations/> (дата звернення: 30.05.2025).
7. How to overcome common student information system challenges. Fierce Network. URL: <https://www.fierce-network.com/more-education-news/how-overcome-common-student-information-system-challenges> (дата звернення: 30.05.2025).
8. JWT | fiber. Welcome | Fiber. URL: <https://docs.gofiber.io/contrib/jwt/> (дата звернення: 30.05.2025).
9. Meshds. Why your institution needs a modern student information system. Document Management Blog | MES. URL: <https://blog.mesltd.ca/how-academic-institutions-benefit-from-digital-student-records> (дата звернення: 30.05.2025).
10. Microservices pattern: pattern: API gateway / backends for frontends. microservices.io. URL: <https://microservices.io/patterns/apigateway.html> (дата звернення: 30.05.2025).

11. MyCreds™. MyCreds™. URL: <https://mycreds.ca/> (дата звернення: 30.05.2025).

12. Official microsoft power apps documentation - power apps. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/power-apps/> (дата звернення: 30.05.2025).

13. Preparing for digital nominations and transcripts of records. Erasmus+. URL: <https://erasmus-plus.ec.europa.eu/news/preparing-for-digital-nominations-and-transcripts-of-records> (дата звернення: 30.05.2025).

14. Shershnev S. Layered architecture implementation in Golang. Medium. URL: <https://medium.com/@shershnev/layered-architecture-implementation-in-golang-6318a72c1e10> (дата звернення: 30.05.2025).

15. The basic principles of user interface design. Studio by UXPin. URL: <https://www.uxpin.com/studio/blog/ui-design-principles/> (дата звернення: 30.05.2025).

16. The best electronic gradebooks for teachers. TeachHub. URL: <https://www.teachhub.com/technology-in-the-classroom/2025/05/the-best-electronic-gradebooks-for-teachers/#:~:text=Canvas:%20A%20Learning%20Management%20System> (дата звернення: 30.05.2025).

17. Thomas R. Guskey. Computerized gradebooks. 2002.

18. Автоматизована система “Деканат”. Географічний факультет. URL: <https://geography.lnu.edu.ua/academics/avtomatyzovana-systema-dekanat> (дата звернення: 30.05.2025).

19. Електронні журнали. Переваги та недоліки. Всеосвіта. URL: <https://vseosvita.ua/blogs/elektronni-zhurnaly-perevahy-ta-nedoliky-94093.html> (дата звернення: 30.05.2025).

20. Засторожнікова І. В. Електронний документообіг у сфері освіти. 2020.

21. Кулинич М. Б. К-90 Електронний документообіг в обліку і оподаткуванні : конспект лекцій. Луцьк : Волинський національний університет імені Лесі Українки», 2022. 68 с.

22. Положення про електронний журнал обліку навчальної роботи студентів академічної групи | ХАІ. Головна | ХАІ. URL: <https://khai.edu.ua/university/normativna-baza/polozheniya1/polozhennya-yaki-regulyuyut-poryadok-zdiysnennya-osvitnogo-procesu/polozhennya-pro-elektronnij-zhurnal-obliku> (дата звернення: 30.05.2025).

23. Про електронні документи та електронний документообіг в Україні – Блог. Вчасно. URL: <https://vchasno.ua/yurydychna-syla-elektronnykh-dokumentiv> (дата звернення: 30.05.2025).

24. Про електронні документи та електронний документообіг. Офіційний вебпортал парламенту України. URL: <https://zakon.rada.gov.ua/laws/show/851-15#Text> (дата звернення: 30.05.2025).

25. Самоаналіз виконання національним закладом вищої освіти критеріїв надання та підтвердження статусу національного. 2019/10/037. Львівський національний університет імені Івана Франка. Львів, 2019.

26. Семенець А. В. Про досвід впровадження системи електронного документообороту у медичному ВНЗ / А. В. Семенець, В. Ю. Ковалок // Медична інформатика та інженерія. - 2014. - № 3.

27. Система «електронний кампус»: відтепер відомості для внесення результатів семестрового контролю формуються та заповнюються онлайн | КПІ ім. Ігоря Сікорського. КПІ ім. Ігоря Сікорського. URL: <https://kpi.ua/ecampus-online-control> (дата звернення: 30.05.2025).

28. Технології е-урядування. Портал знань. URL: <http://www.znannya.org/?view=e-government-documents> (дата звернення: 30.05.2025).

29. Шевченко В. О. Розвиток системи електронного документообігу в Україні : Кваліфікаційна робота. Полтава, 2021.

30. Що таке система управління документами (DMS) - терміни та визначення кібербезпеки. VPN Unlimited - Fast & Secure VPN service. URL: <https://www.vpnunlimited.com/ua/help/cybersecurity/document-management-system> (дата звернення: 30.05.2025).

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інформаційних технологій та систем

Кваліфікаційна робота

на тему:

**«ПРОГРАМНИЙ МОДУЛЬ ДЛЯ АВТОМАТИЧНОГО ГЕНЕРУВАННЯ НАВЧАЛЬНОЇ
ВІДОМОСТІ СТУДЕНТІВ НА ОСНОВІ ДАНИХ З ЕЛЕКТРОННОГО ЖУРНАЛУ УСПІШНОСТІ»**

На здобуття освітнього рівня бакалавра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи
та технології

Виконала: Гаврилюк А.Р., ІСД-43

Науковий керівник: к.т.н., Аронов А.О.

Київ 2025

02

АКТУАЛЬНІСТЬ

В умовах цифрової трансформації освіти особливої важливості набуває автоматизація процесів обліку успішності студентів. Ручне формування навчальних відомостей є трудомістким, схильним до помилок і не відповідає сучасним вимогам до швидкості, точності та прозорості освітньої документації. Тому створення модуля, який дозволяє автоматизувати цей процес на основі даних з електронного журналу, є актуальним і практично значущим для закладів вищої освіти.

МЕТА

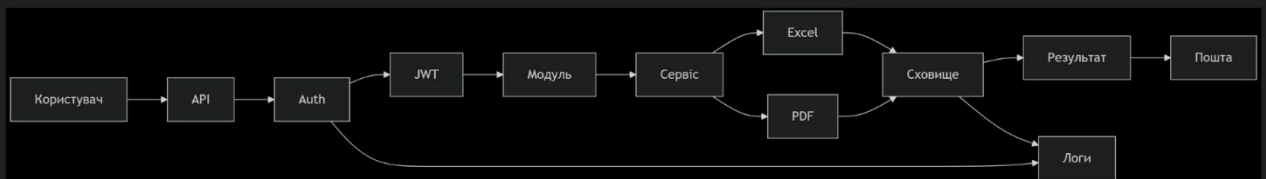
Розробка програмного модуля для автоматичного генерування навчальної відомості студентів на основі даних з електронного журналу успішності. Це дозволить підвищити ефективність документообігу та зменшити навантаження на викладачів і адміністрацію.

ЗАВДАННЯ



- Проаналізувати існуючі системи електронного обліку успішності.
- Визначити вимоги до модуля генерації навчальних відомостей.
- Розробити модуль на та інтегрувати його з електронним журналом.
- Реалізувати експорт відомостей у форматах PDF і Excel.
- Провести тестування та оцінити ефективність рішення.

АРХІТЕКТУРА СИСТЕМИ



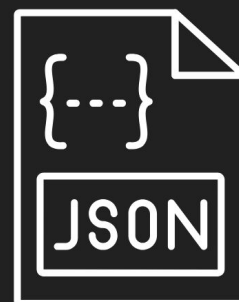
АРХІТЕКТУРА API-ГЕНЕРАЦІЇ ДОКУМЕНТІВ



РЕАЛІЗАЦІЯ



Для реалізації бекенду було використано мову Go та фреймворк Fiber, що забезпечує зручну побудову REST API. Формат вхідних даних представлено у вигляді JSON-файлу, який можна завантажити через інтерфейс або розмістити в корені проекту. У відповідь на запит система генерує Excel-документ за допомогою бібліотеки Excelize на основі даних про конкретного студента.



```
func handleGetStudents(c *fiber.Ctx) error {
    dataMutex.RLock()
    defer dataMutex.RUnlock()

    group := c.Query("group")
    if group == "" {
        return c.Status(fiber.StatusBadRequest).JSON(fiber.Map{"error": fiber.Map{"code": "004001", "attributes": fiber.Map{"message": "Missing group parameter"}}})
    }

    if appData == nil {
        return c.Status(fiber.StatusInternalServerError).JSON(fiber.Map{"error": fiber.Map{"code": "005001", "attributes": fiber.Map{"message": "БД немає даних"}}})
    }

    if _, ok := appData.Groups[group]; !ok {
        return c.Status(fiber.StatusNotFound).JSON(fiber.Map{"error": fiber.Map{"code": "004041", "attributes": fiber.Map{"message": "Групу не знайдено"}}})
    }

    return c.Status(fiber.StatusOK).JSON(fiber.Map{"success": fiber.Map{"code": "002001", "attributes": fiber.Map{"message": appData.GetStudentsByGroup(group)}}})
}

func handleGetStudent(c *fiber.Ctx) error {
    dataMutex.RLock()
    defer dataMutex.RUnlock()

    group := c.Query("group")
    student := c.Query("student")
    if group == "" || student == "" {
        return c.Status(fiber.StatusBadRequest).JSON(fiber.Map{"error": fiber.Map{"code": "004001", "attributes": fiber.Map{"message": "Немає group/student параметрів"}}})
    }

    if appData == nil {
        return c.Status(fiber.StatusInternalServerError).JSON(fiber.Map{"error": fiber.Map{"code": "005001", "attributes": fiber.Map{"message": "БД немає даних"}}})
    }

    // Перевірка на наявність групи БД + студента
    gd, ok := appData.Groups[group]
    if !ok {
        return c.Status(fiber.StatusNotFound).JSON(fiber.Map{"error": fiber.Map{"code": "004041", "attributes": fiber.Map{"message": "Групу не знайдено"}}})
    }
}
```

Відомості студентів

Завантажити загальну відомість

Виберіть один або декілька .json:

Виберіть один або декілька .json

Виберіть групу:

PD42




Перезавантажити
БД

Виберіть студента:

Anzhelika V.V.



 Завантажити відомість

Результат тестування

09

A	B	C	D	E	F	G	H	Строка форму
ID предмету	ПІБ	№ заліково	чаткові ба	и з екза	мально ба	цінка з ес	інка загал	Підписано
1	Anzhelika V.V.	1234	50	75	100	A	100	ИСТИНА
1	Anzhelika V.V.	1234	50	75	100	A	100	ИСТИНА
ID предмету	Назва предмету							
1	testing							
2	testing2							

Екзаменаційна

Додаткова екзаменаційна

Сесія

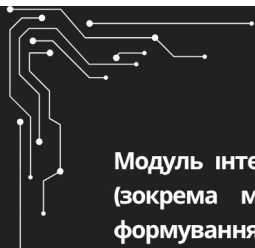
Додаткова сесія



ПЕРСПЕКТИВИ ВПРОВАДЖЕННЯ

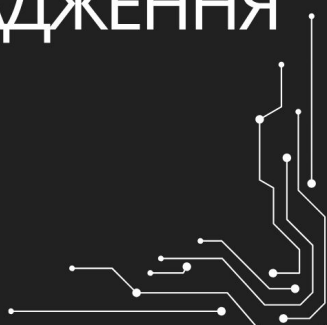
10

Перспективи впровадження розробленого модуля полягають у його інтеграції з електронними журналами закладів вищої освіти для автоматизованого формування навчальних відомостей, що дозволить зменшити адміністративне навантаження, підвищити точність звітності та забезпечити відповідність сучасним вимогам цифровізації освітнього процесу.



Модуль інтегрується з електронним журналом (зокрема модулем «Сесія»), забезпечуватиме формування звітів на запит студента у форматах PDF/Excel, а також дозволить створювати офіційні виписки для подання до інших установ чи участі в академічних програмах. Інтерфейс підтримуватиме адаптацію під корпоративний стиль ЗВО або державних цифрових сервісів. Крім того, система підтримуватиме автоматичне обмеження доступу до формування звітів за періодом (наприклад, після завершення семестру), що підвищує контроль і відповідність адміністративній документації.

ПЕРЕДБАЧЕНІ СЦЕНАРІЇ ВПРОВАДЖЕННЯ



АПРОБАЦІЯ РЕЗУЛЬТАТІВ РОБОТИ

**II ВСЕУКРАЇНЬКА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ ТЕХНОЛОГІЧНІ ГОРИЗОНТИ:
ДОСЛІДЖЕННЯ ТА ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ
ТЕХНОЛОГІЧНОГО ПРОГРЕСУ УКРАЇНИ І СВІТУ**

**ТЕЗИ: «ПЕРСПЕКТИВИ ВИКОРИСТАННЯ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ
АДАПТИВНОГО НАВЧАННЯ У ВИЩІЙ ОСВІТІ »,
«ТЕХНОЛОГІЯ ВІРТУАЛЬНОЇ ТА ДОПОВНЕНОЇ РЕАЛЬНОСТІ В НАВЧАЛЬНИХ
ПРОГРАМАХ: ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ТА МОТИВАЦІЙ СТУДЕНТІВ »,
«КІБЕРБЕЗПЕКА В АРХІТЕКТУРІ ІОТ ДЛЯ РОЗУМНОГО МІСТА: ЗАХИСТ ДАНИХ І
ПРИВАТНОСТІ ГРОМАДЯН »,
«ОСОБЛИВОСТІ РОЗРОБКИ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ УПРАВЛІННЯ ТА
ОПТИМІЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ АВТОМОБІЛЬНОГО САЛОНУ: СУЧАСНІ ПІДХОДИ
ТА ІНСТРУМЕНТИ ЦИФРОВОЇ ТРАНСФОРМАЦІЇ »**



На основі проведеного дослідження встановлено, що сучасні системи електронного обліку успішності студентів мають суттєві обмеження у формуванні офіційних навчальних відомостей, зокрема через фрагментованість даних, складність експорту та відсутність гнучких інструментів автоматизації. Для вирішення цих проблем у дипломній роботі було розроблено програмний модуль, який забезпечує автоматичне генерування навчальних відомостей на основі даних з електронного журналу.

Перевагами розробленого модуля є автоматизація процесу формування навчальних відомостей, зменшення навантаження на викладачів, підтримка зручних форматів (PDF та Excel), інтеграція з особистим кабінетом студента та можливість формування звіту за вибраний період навчання.

Запропонований модуль є практичним кроком до цифровізації освітнього документообігу у вищих навчальних закладах України. Його впровадження сприятиме підвищенню ефективності, точності та прозорості облікових процесів. У перспективі модуль може бути розширено за рахунок впровадження електронного підпису, глибшої інтеграції з ЄДЕБО та підтримки додаткових форматів звітності.

ДЯКУЮ ЗА УВАГУ!