



Наукові перспективи
Видавнича група

№ 3 (57)

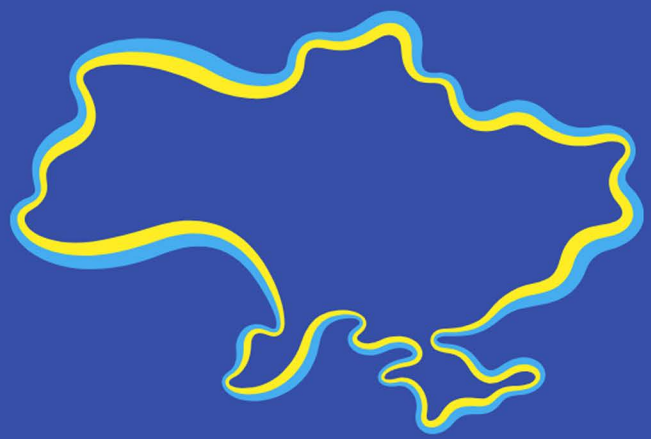
2026

ІТ НАУКА ТЕХНІКА

СЬОГОДНІ



З Україною
в серці!



Видавнича група «Наукові перспективи»

Всеукраїнська Асамблея докторів наук із державного управління

«Наука і техніка сьогодні»

Випуск № 3(57) 2026

Київ – 2026

Publishing Group «Scientific Perspectives»

Ukrainian Assembly of Doctors of Sciences in Public Administration

"Science and technology today"

Issue № 3(57) 2026

Kyiv - 2026

ISSN 2786-6025 Online

УДК 001.32:1 /3](477)(02)

R40-05553

DOI:  Crossref
we use DOIs

[https://doi.org/10.52058/2786-6025-2026-3\(57\)](https://doi.org/10.52058/2786-6025-2026-3(57))

**«Наука і техніка сьогодні» (Серія «Педагогіка», Серія «Право», Серія «Економіка»,
Серія «Фізико-математичні науки», Серія «Техніка»):
журнал. 2026. № 3(57) 2026. С. 3273**



*Згідно наказу Міністерства освіти і науки України від 07.04.2022 № 320
журналу присвоєно категорію "Б" із економіки та педагогіки
(спеціальності – 015 - Педагогічні науки; 076 - Економічні науки)*

*Згідно наказу Міністерства освіти і науки України від 06.06.2022 № 530 журналу
присвоєно категорію "Б" із права (спеціальність – 081 Юридичні науки)*

*Згідно наказу Міністерства освіти і науки України від 10.10.2022 № 894 журналу
присвоєно категорію "Б" із техніки (спеціальність - 122 Комп'ютерні науки)*

Журнал видається за підтримки Міждержавної гільдії інженерів консультантів, Інституту філософії та соціології Національної Академії Наук Азербайджану (Баку, Азербайджан), громадської організації «Християнська академія педагогічних наук України» та громадської організації «Всеукраїнська асоціація педагогів і психологів з духовно-морального виховання»

*Рекомендовано до видавництва Президією Всеукраїнської Асамблеї докторів наук з державного управління
(Рішення від 24.03.2026, № 9/3-26)*



Журнал включено до міжнародної наукометричної бази Index Copernicus (IC), міжнародної пошукової системи Google Scholar та до міжнародної наукометричної бази даних Research Bible

Згідно Порядку формування Переліку наукових фахових видань України, затвердженого наказом МОН України від 15.01.2018 № 32, повнотекстовий доступ до наукових статей журналу представлений на платформі «Наукова періодика України» в Національній бібліотеці України імені В.І. Вернадського НАН України та в Національному репозитарії академічних текстів

Головний редактор:



Коренева Інна Миколаївна - доктор педагогічних наук, професор, декан факультету природничої і фізико-математичної освіти Глухівського національного педагогічного університету імені Олександра Довженка; професор кафедри теорії і методики викладання природничих дисциплін Глухівського національного педагогічного університету імені Олександра Довженка (Україна)

Редакційна колегія:

1. **Біляковська Ольга Орестівна** доктор педагогічних наук, професор, завідувачка кафедри загальної педагогіки та педагогіки вищої школи Львівського національного університету імені Івана Франка (Україна)

2. **Воровка Маргарита Іванівна** – докторка педагогічних наук, професорка, професорка кафедри освітології та педагогіки мистецтва Мелітопольського державного педагогічного університету імені Богдана Хмельницького (Україна)

3. **Гончарук Валентина Анатоліївна** кандидат педагогічних наук, доцент, доцент кафедри української літератури, українознавства та методик їх навчання Уманського державного педагогічного університету імені Павла Тичини (Україна)

Кирильчук М.Л., Белзецький Р.С. **2460**
*МОБІЛЬНА ІНФОРМАЦІЙНА СИСТЕМА «ВЕЕPLANNER» ДЛЯ
АВТОМАТИЗОВАНОГО КЕРУВАННЯ ТА ПІДТРИМКИ ПРИЙ-
НЯТТЯ РІШЕНЬ У БДЖІЛЬНИЦТВІ*

Кобус О.С. **2475**
*ІНФОРМАЦІЙНО-ТЕОРЕТИЧНІ ОСНОВИ ФОРМУВАННЯ
СТІЙКИХ МОДЕЛЕЙ ПЕРЕДАВАННЯ ДАНИХ У СКЛАДНИХ
БАГАТОРІВНЕВИХ КОМУНІКАЦІЙНИХ СЕРЕДОВИЩАХ*

Ковалевич Р.В., Колпіков А.С., Устенко С.А. **2488**
*МОДЕЛІ ТА МЕТОДИ ПОВЕДІНКОВО-ІДЕНТИФІКАЦІЙНОГО
АНАЛІЗУ ЛЮДИНИ В ІНТЕЛЕКТУАЛЬНИХ СИСТЕМАХ
ВІДЕОСПОСТЕРЕЖЕННЯ*

Ковальський С.С., Коваль В.С., Панчак Д.В. **2502**
*МЕТОДИ FEW-SHOT LEARNING ТА TRANSFER LEARNING
ДЕФЕКТУВАННЯ ДЕРЕВ'ЯНИХ ВИРОБІВ В УМОВАХ
ОБМЕЖЕНОЇ КІЛЬКОСТІ ДАНИХ*

Колесник О.Б., Тітов С.В., Тітова О.В., Чорна О.С. **2518**
*МАТЕМАТИЧНА МОДЕЛЬ ПРОЦЕСУ ФУНКЦІОНАЛЬНОГО
УРАЖЕННЯ ЕЛЕМЕНТІВ РТЗ ПРИ ВИКОРИСТАННІ БАГАТО-
ЧАСТОТНИХ ПРОСТОРОВО-ЧАСОВИХ СИГНАЛІВ ПЛОСКОЇ
ФАР*

Коповський С.М. **2533**
*ЗАСТОСУВАННЯ ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ
У ФОРМУВАННІ ТЕСТОВОЇ ДОКУМЕНТАЦІЇ*

Котляревський О.В. **2541**
*ІНТЕГРАЦІЯ ВІЙСЬКОВОЇ ТА БУДІВЕЛЬНОЇ ЛОГІСТИКИ В
УКРАЇНІ ЯК ЧИННИК ЗАБЕЗПЕЧЕННЯ ОБОРОННОЇ ІНФРА-
СТРУКТУРИ*

Крилик Л.В. **2550**
*ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ЧИСЕЛЬНИХ МЕТОДІВ В
НЕЛІНІЙНИХ ЗАДАЧАХ*

ISSN 2786-6025 Online

Крилик Л.В., Павлуш Є.П.

*РОЗРОБКА WEBРЕСУРСУ ОБЛІКУ ВІДВІДУВАННЯ ЗАНЯТЬ
СТУДЕНТАМИ*

2560

Кудрявський Дмитро А., Колесник Л.В., Колесник .Б.

*АДАПТИВНА ГІБРИДНА МОДЕЛЬ РЕКОМЕНДАЦІЙ КРУЇЗНИХ
ТУРІВ З ДИНАМІЧНИМ БАЛАНСУВАННЯМ ВАГОВИХ КОЕФІ-
ЦІЄНТІВ*

2573

**Ліп'яніна-Гончаренко Х.В., Комар М.П., Биковий П.Є.,
Юрків Х.В., Гончаренко О.О.**

*МЕТРИКИ ВІДПОВІДАЛЬНОГО ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ
ПРИКЛАДНИХ СИСТЕМ*

2586

Лобачев М.В., Шутов С.І.

*МОДЕЛІ ТА МЕТОДИ ПІДТРИМКИ УПРАВЛІННЯ РИЗИКАМИ
ІТ-ПРОЄКТІВ ІЗ ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ ШТУЧНОГО
ІНТЕЛЕКТУ*

2605

Любарський С.В.

*ЕВОЛЮЦІЯ ПІДХОДІВ ДО ЖИТТЄВОГО ЦИКЛУ ІНФОРМА-
ЦІЙНИХ СИСТЕМ В УМОВАХ ЦИФРОВОЇ ТРАНСФОРМАЦІЇ*

2619

Люшенко А.М.

*СИСТЕМНО-РІВНЕВА МЕТОДОЛОГІЯ ІДЕНТИФІКАЦІЇ АРХІ-
ТЕКТУРНИХ ВУЗЬКИХ МІСЦЬ У МОБІЛЬНИХ ПЛАТФОРМАХ*

2637

Малашонок Г.І., Сухарський С.С.

*ТРИДАГОНАЛІЗАЦІЯ СТРІЧКОВИХ СИМЕТРИЧНИХ МАТ-
РИЦЬ*

2647

Мних А.С., Гончаренко І.С., Тищенко Д.О., Франчук Т.М.

*ВИКОРИСТАННЯ МІКРОІНТЕРАКЦІЙ У СУЧАСНОМУ ВЕБ-
ДИЗАЙНІ ДЛЯ ПОКРАЩЕННЯ КОРИСТУВАЦЬКОГО ДОСВІДУ*

2664

Мудрий І.В.

*МОНІТОРИНГ МЕРЕЖЕВОГО ТРАФІКУ ІЗ ЗАСТОСУВАННЯМ
МЕТОДІВ МАШИННОГО НАВЧАННЯ*

2675

Крилик Людмила Вікторівна кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, <https://orcid.org/0000-0001-6642-754X>;

Павлуш Єлизавета Петрівна бакалавриня факультету інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця

РОЗРОБКА WEBРЕСУРСУ ОБЛІКУ ВІДВІДУВАННЯ ЗАНЯТЬ СТУДЕНТАМИ

Анотація. У статті представлено особливості розробки WEBресурсу обліку відвідування занять студентами. Метою розробки є розширення функціональних можливостей WEBресурсу обліку відвідування занять студентами шляхом впровадження клієнт-серверної архітектури, автоматизації процесу фіксації присутності, генерації звітів у різних форматах та надання студентам доступу до власної статистики. Розробка є сучасним WEBзастосунком, що складається з двох основних частин, а саме, клієнтської та серверної. Серверна частина реалізована на платформі Node.js з використанням фреймворку Express та реляційної бази даних SQLite. Клієнтська частина побудована з використанням HTML5, CSS3 та JavaScript із застосуванням шаблонізатора EJS для серверного рендерингу сторінок. Для підвищення ефективності розробки, масштабованості та зручності підтримки WEBресурс поділено на окремі модулі: автентифікації, управління студентами, обліку відвідуваності, генерації звітів та статистики. Такий підхід дозволяє ізолювати бізнес-логіку кожного компонента, спрощує тестування та модернізацію коду. Розроблено UML-діаграм класів для клієнтської та серверної частин WEBресурсу обліку відвідування занять студентами, це забезпечує цілісне проектування системи.

Діаграми наочно демонструють архітектуру WEBресурсу, розподіл відповідальностей між компонентами та їх взаємозв'язки. Розроблений WEBресурс має розширений функціонал порівняно з існуючими аналогами щонайменше на 4 можливості: динамічний підрахунок статистики в реальному часі під час відмітки присутності; генерація звітів у трьох форматах з PDF, Word та Excel з попереднім переглядом; надання студентам доступу до персоналізованої статистики з візуальними індикаторами; реалізація відкритого API для подальшої інтеграції з іншими системами. Завдяки цьому

ISSN 2786-6025 Online

розроблений WEBресурс забезпечує зручний україномовний інтерфейс, швидке внесення даних, прозорість навчального процесу та розширені можливості масштабування у порівнянні з існуючими системами обліку відвідування.

Ключові слова: облік відвідування, WEBресурс, автоматизація, студенти, Node.js, Express, SQLite, генерація звітів.

Krylik Lyudmila Viktorivna Candidate of Technical Sciences, Associate Professor, Associate Professor at the Department for Computer Science, Vinnytsia National Technical University, Vinnytsia, <https://orcid.org/0000-0001-6642-754X>

Pavlush Yelyzaveta Petrivna Bachelor of the Department for Computer Science, Vinnytsia National Technical University, Vinnytsia

DEVELOPMENT OF A WEB RESOURCE FOR ACCOUNTING STUDENT ATTENDANCE

Abstract. The article presents the features of the development of a WEB resource for recording student attendance. The purpose of the development is to expand the functionality of the WEB resource for recording student attendance by implementing a client-server architecture, automating the process of recording attendance, generating reports in various formats, and providing students with access to their own statistics. The development is a modern WEB application, consisting of two main parts, namely, client and server. The server part is implemented on the Node.js platform using the Express framework and the SQLite relational database. The client part is built using HTML5, CSS3 and JavaScript using the EJS template engine for server-side page rendering. To increase the efficiency of development, scalability and ease of support, the WEB resource is divided into separate modules: authentication, student management, attendance accounting, report generation and statistics. This approach allows isolating the business logic of each component, simplifies testing and code modernization. UML class diagrams have been developed for the client and server parts of the WEB resource for accounting for student attendance, which ensures a holistic system design. The diagrams clearly demonstrate the architecture of the WEB resource, the distribution of responsibilities between components, and their relationships. The developed WEB resource has expanded functionality compared to existing analogues by at least 4 possibilities: dynamic calculation of statistics in real time during attendance marking; generation of reports in three formats with PDF, Word, and Excel with preview; providing students with access to personalized statistics with visual indicators; implementation of an open API for further integration with other systems. Due to this, the developed WEB resource provides a convenient Ukrainian-language interface, quick data entry,

ISSN 2786-6025 Online

transparency of the educational process, and expanded scalability compared to existing attendance accounting systems.

Keywords: attendance accounting, WEB resource, automation, students, Node.js, Express, SQLite, report generation.

Постановка проблеми. Цифровізація освітнього процесу є одним із ключових напрямів розвитку сучасних закладів освіти. Важливим аспектом організації навчального процесу є облік відвідуваності студентів, що безпосередньо впливає на якість навчання, дисципліну та контроль академічної успішності. Традиційні методи реєстрації відвідуваності, такі як паперові списки або електронні таблиці, часто виявляються недосконалими через низьку ефективність, значні витрати часу викладачів та ризик втрати чи спотворення даних.

Крім того, інформація зберігається у розрізненому вигляді, що ускладнює централізований доступ та оперативне формування статистичних даних для груп, факультетів чи університету в цілому [1, 2].

Незважаючи на наявність окремих програмних рішень для обліку відвідуваності, більшість існуючих систем мають обмежений функціонал, відсутність україномовного інтерфейсу, платний доступ або недоступність для використання студентами та співробітниками інших ВНЗ конкретного навчального закладу. Це створює потребу у створенні сучасного WEBресурсу, який забезпечував би простий, надійний та оперативний механізм контролю відвідуваності студентів з можливістю автоматизованого формування звітів та надання статистики як викладачам, так і студентам [3–5].

Аналіз останніх досліджень і публікацій. Нині існує кілька програмних рішень для обліку відвідуваності занять студентами, зокрема MyAttendanceTracker [3], Attendance Radar [4] та JetIQ [5]. MyAttendanceTracker є простим сервісом з базовою аналітикою, однак він не підтримує україномовний інтерфейс та не автоматизує процес відмітки присутності. Attendance Radar використовує технологію Bluetooth для автоматичної фіксації присутності, але потребує додаткового обладнання та є платним. JetIQ – україномовна система, інтегрована у внутрішнє середовище Вінницького національного технічного університету, проте вона недоступна для використання студентами та співробітниками інших ВНЗ.

На основі проведеного аналізу встановлено, що існуючі рішення не забезпечують повною мірою поєднання таких важливих характеристик, як безкоштовність, україномовний інтерфейс, доступність для будь-якого закладу, генерація звітів у різних форматах та надання статистики студентам. Це обґрунтовує актуальність розробки нового WEBресурсу, який би поєднував ці можливості.

Метою статті є розширення функціональних можливостей WEBресурсу обліку відвідування занять студентами шляхом впровадження клієнт-серверної архітектури, автоматизації процесу фіксації присутності, генерації звітів у різних форматах та надання студентам доступу до власної статистики.

Виклад основного матеріалу. WEBресурс обліку відвідування занять студентами це комплексне рішення, яке поєднує зручний інтерфейс для різних категорій користувачів із потужним функціоналом управління навчальним процесом. Для забезпечення високої продуктивності, масштабованості та простоти супроводу було обрано гібридний архітектурний підхід, який поєднує принципи багаторівневої архітектури (N-Layer) та компонентно-орієнтованого проектування. Це дозволяє забезпечити як цілісність системи в цілому, так і високу модульність окремих функціональних частин.

Розроблюваний WEBресурс складається з клієнтської та серверної частин. Серверна частина реалізована на платформі Node.js з використанням фреймворку Express, що забезпечує стабільність роботи, продуктивність та можливість масштабування системи [6, 7]. Клієнтська частина побудована з використанням HTML5, CSS3 та JavaScript із застосуванням шаблонізатора EJS для серверного рендерингу сторінок, що гарантує швидке завантаження та зручний інтерфейс. Для зберігання та обробки даних використано реляційну базу даних SQLite, яка забезпечує надійність, цілісність даних та простоту розгортання [8].

Клієнтська частина відповідає за взаємодію користувача із системою через інтуїтивний графічний інтерфейс, а серверна частина забезпечує логіку роботи застосунка: обробку запитів, автентифікацію, збереження даних студентів, облік відвідуваності та формування звітів.

WEBресурс поділено на окремі модулі, які виконують конкретну функцію системи, це підвищує ефективність розробки, масштабованість та зручність підтримки. Такий підхід дозволяє ізолювати бізнес-логіку кожного компонента, спрощує тестування і модернізацію коду.

У системі передбачено кілька основних модулів: модуль автентифікації (забезпечує реєстрацію, вхід, вихід та розмежування прав доступу для викладачів і студентів); модуль управління студентами (відповідає за додавання, перегляд, редагування, видалення та пошук студентів); модуль обліку відвідуваності (реалізує процес відмітки присутності студентів з динамічним підрахунком статистики); модуль генерації звітів (забезпечує формування звітів у форматах PDF, Word та Excel); модуль статистики (надає студентам деталізовану інформацію про їхню відвідуваність за дисциплінами); модуль профілю (дозволяє користувачам переглядати та редагувати персональні дані, змінювати пароль). На макроархітектурному рівні система організована за принципом багаторівневої архітектури з чітким розділенням

ISSN 2786-6025 Online

відповідальності між рівнем представлення, рівнем бізнес-логіки, рівнем доступу до даних та рівнем зберігання даних [9]. Загальна структурна схема функціонування WEBресурсу обліку відвідування занять студентами наведена на рис. 1.

Головна сторінка системи є основним інтерфейсом взаємодії користувача з програмним комплексом. Вона містить компонент відображення журналу відвідувань, який ініціює API-запит до модуля управління навчальними даними. Запит обробляється на серверній стороні, отримує відповідні дані з бази даних та повертає структуровану відповідь до клієнтської частини. Отримана інформація, яка включає дані про групу, студентів, дисципліни та попередні відмітки про відвідування, візуалізується у формі електронного журналу. Ключовим функціоналом системи є механізм фіксації відвідування. При виборі відповідного заняття користувач отримує доступ до інтерфейсу відмітки присутності. Подання форми з даними про відвідування ініціює HTTP POST запит до модуля обліку відвідування. Запит містить структурований набір даних, що включає ідентифікатори студентів, статус присутності, мітку часу та ідентифікатор викладача.

Модуль обліку відвідування здійснює валідацію отриманих даних щодо відповідності бізнес-правилам системи. Після успішної валідації дані передаються до модуля керування даними, який відповідає за операції збереження інформації. Цей модуль забезпечує транзакційність операцій та цілісність даних при виконанні операцій запису в базу даних.

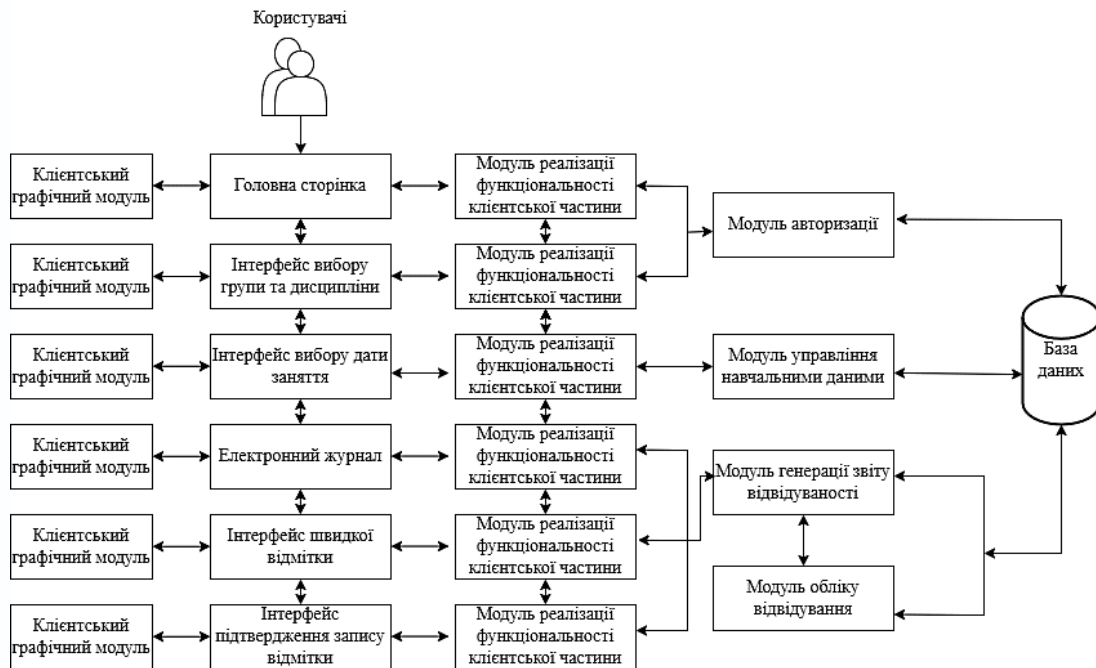


Рис. 1. Загальна структурна схема функціонування WEBресурсу обліку відвідування занять студентами

Загальна структурна схема компонентів WEBресурсу обліку відвідування занять студентами подана на рис. 2.

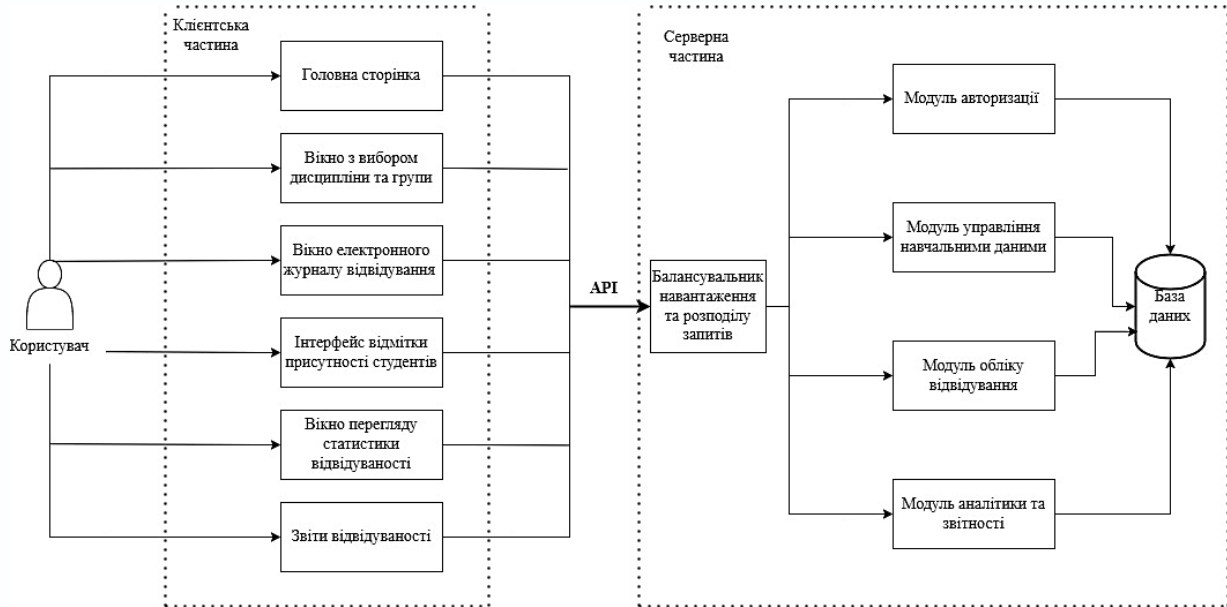


Рис. 2. Загальна структурна схема компонентів WEBресурсу обліку відвідування занять студентами

Усі API-запити від клієнтської частини до серверних компонентів проходять через балансувальник навантаження. Основне завдання балансувальника полягає в оптимальному розподілі запитів між доступними екземплярами мікросервісів, що забезпечує високу доступність системи та ефективне використання ресурсів.

Система використовує реляційну базу даних як основне сховище інформації. База даних зберігає структуровані дані про користувачів, навчальні групи, дисципліни та записи про відвідування. Архітектура бази даних забезпечує швидкий доступ до даних, підтримує оперативне оновлення інформації та гарантує цілісність даних при паралельній роботі з багатокористувацьким доступом. Проектування внутрішньої будови програмного забезпечення здійснювалося із застосуванням уніфікованої мови моделювання UML [10, 11]. З її допомогою було створено діаграми класів, які відображають статичну структуру як серверної, так і клієнтської частин застосунка. Серверна частина містить класи, згруповані за функціональними модулями: керування даними, облік відвідуваності, автентифікація, робота з групами та студентами, аналітика та звітність. Клієнтська частина складається з компонентів, що забезпечують відображення журналу відвідувань, панелі статистики, генератора звітів, інтерфейсу для відмітки присутності та сервісу для комунікації з сервером. Взаємодія між компонентами організована через

ISSN 2786-6025 Online

головний модуль, який координує їхню роботу та підтримує єдиний стан застосунка. Діаграми класів для серверної та клієнтської частин представлено на рис. 3 та рис. 4 відповідно.

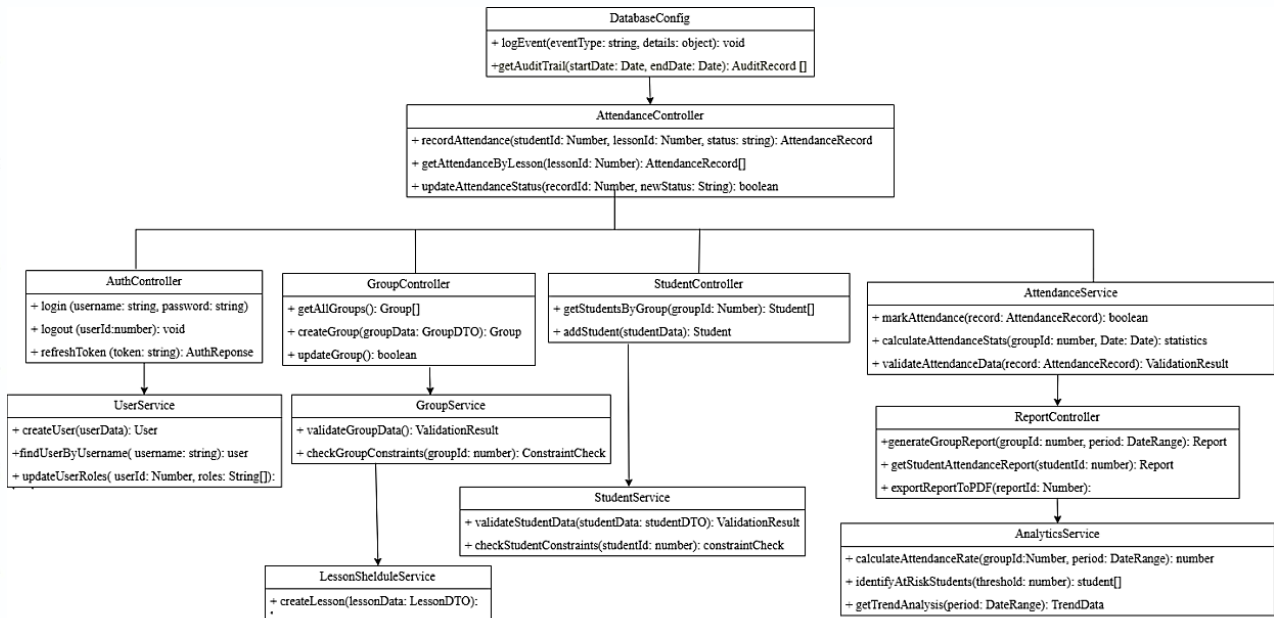


Рис. 3. UML-діаграма класів серверної частини WEBресурсу обліку відвідування занять студентами

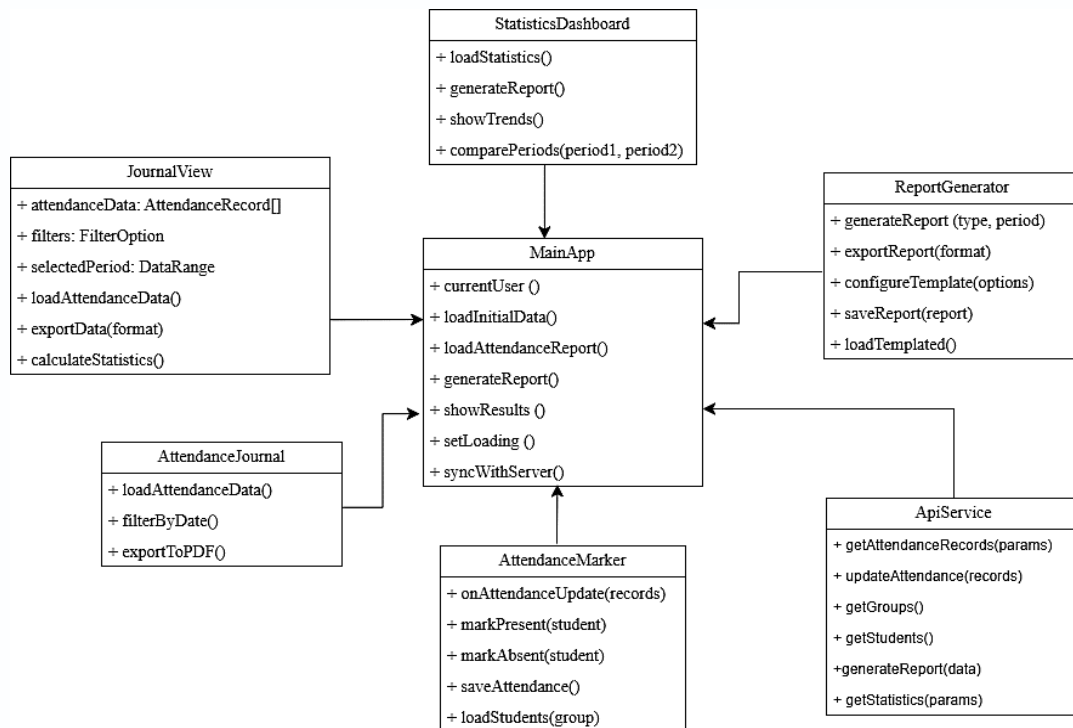


Рис. 4. UML-діаграма класів клієнтської частини WEBресурсу обліку відвідування занять студентами

Розглянемо одне з ключових програмних рішень, яке забезпечує розширення функціональних можливостей WEBресурсу обліку відвідуваності, а саме сторінку перегляду відвідуваності з можливістю фільтрації та експорту даних.

Фрагмент програмної реалізації маршруту /view_attendance, що відповідає за відображення історії відвідуваності:

```
app.get('/view_attendance', requireAuth, requireRole('teacher'), (req, res) => {
  db.all(`
    SELECT a.*, s.surname, s.name, s.lastname, s."group"
    FROM attendance a
    LEFT JOIN students s ON a.student_id = s.id
    ORDER BY a.date DESC, s."group", s.surname
    LIMIT 100
  ` , (err, attendanceRecords) => {
    if (err) {
      console.error('Error:', err.message);
      return res.status(500).render('error', {
        title: 'Помилка сервера',
        message: 'Помилка завантаження відвідуваності',
        status: 500
      });
    }

    res.render('view_attendance', {
      title: 'Перегляд відвідуваності | Attendance Student',
      attendanceRecords,
      username: req.session.username,
      formatStatus: (status) => status === 'Present' ? 'Присутній' : 'Відсутній'
    });
  });
});
```

У наведеному фрагменті коду реалізовано:

- виконання SQL-запиту з об'єднанням таблиць attendance та students для отримання детальної інформації про кожен запис;
- сортування результатів за датою (від новіших до старіших), групою та прізвищем студента;
- обмеження кількості записів до 100 для оптимізації продуктивності;
- передачу даних у шаблон view_attendance.ejs разом із допоміжною функцією formatStatus для локалізації статусів.

ISSN 2786-6025 Online

На рис. 5 наведено загальний вигляд сторінки перегляду відвідуваності, яка містить фільтри, статистичні картки та таблицю з даними.

Сторінка перегляду відвідуваності (view_attendance.ejs) реалізує такий функціонал:

1. Фільтрація даних: на стороні клієнта реалізовано JavaScript-функцію `applyFilters`, яка дозволяє фільтрувати записи за ПІБ студента, групою, дисципліною та датою без перезавантаження сторінки.

2. Експорт даних: реалізовано функцію `exportData`, яка дозволяє вивантажувати відфільтровані дані у форматах PDF, Word та Excel [12].

3. Статистичні картки: у верхній частині сторінки розташовано три картки із загальною статистикою, які розраховуються на сервері.

4. Таблиця даних: записи відображаються у вигляді таблиці з кольоровим кодуванням статусів: зелений фон для присутніх, червоний – для відсутніх.

Дата	Студент	Група	Дисципліна	Статус	Час запису
2026-02-19	ID: 0	-	ММДО	Присутній	Записано
2026-02-19	ID: 0	-	Прикладна робототехніка	Відсутній	Записано
2026-02-19	Тестовий Тестування Тест ID: 1	Тест-1	ММДО	Відсутній	Записано

Рис. 5. Відображення сторінки перегляду відвідуваності з фільтрацією та експортом

Розроблений WEBресурсу обліку відвідування занять студентами успішно пройшов тестування.

В табл. 1 подано результати порівняльного тестування розробки та аналогів: MyAttendanceTracker, Attendance Radar, JetIQ.

Таблиця 1

Результати тестування розробленого програмного продукту та аналогів

Критерій	Розробка	MyAttendanceTracker	Attendance Radar	JetIQ
Україномовний інтерфейс	+	–	–	+
Безкоштовність	+	±	–	±
Автоматизація відмітки	–	–	+	+
Доступність поза межами закладу	+	+	+	–
Генерація звітів (PDF, Word, Excel)	+	–	–	+
Статистика для студентів	+	–	–	+
Розмежування прав доступу	+	–	–	+
Адаптивний дизайн	+	+	+	±
Простота впровадження	Висока	Висока	Середня	Низька
Вимоги до обладнання	Відсутні	Відсутні	Bluetooth-пристрої	Серверне ПЗ
Наявність API для інтеграції	+	–	–	±

На основі аналізу даних табл. 1 видно, що розроблений WEBресурс має суттєві переваги порівняно з існуючими аналогами. На відміну від MyAttendanceTracker та Attendance Radar, система має повноцінний україномовний інтерфейс, що робить її зручною для використання в українських закладах освіти. JetIQ – це система підтримки навчального процесу Вінницького національного технічного університету, яка також підтримує українську мову. Однак вона недоступна для використання студентами та співробітниками інших ВНЗ, тоді як розроблений ресурс може бути впроваджений у будь-якому навчальному закладі без обмежень.

Важливою перевагою є повна безкоштовність системи, тоді як Attendance Radar є платним, а MyAttendanceTracker має обмежену безкоштовну версію.

ISSN 2786-6025 Online

JetIQ, хоча і безкоштовний для ВНТУ, але вимагає значних витрат на впровадження та підтримку в інших навчальних закладах.

Розроблений WEBресурс забезпечує генерацію звітів у трьох форматах (PDF, Word, Excel), що відсутнє в MyAttendanceTracker та Attendance Radar. Це дозволяє викладачам легко формувати необхідну звітність для адміністрації, деканатів або для власного аналізу. Крім того, система надає студентам доступ до власної статистики відвідуваності з деталізацією за дисциплінами та візуальними індикаторами, що підвищує прозорість навчального процесу та мотивацію студентів. Ця функція відсутня в MyAttendanceTracker та Attendance Radar.

З технічної точки зору, розроблений ресурс має просте впровадження та не потребує додаткового обладнання на відміну від Attendance Radar, який вимагає Bluetooth-пристроїв, або JetIQ, що потребує складного серверного програмного забезпечення. Для роботи WEBресурсу достатньо звичайного комп'ютера з Node.js. Система також реалізує фільтрацію даних на стороні клієнта без перезавантаження сторінки, це забезпечує миттєвий зворотний зв'язок та зручність роботи з великими обсягами інформації – жоден з розглянутих аналогів не має подібного функціоналу.

Окремо потрібно відзначити наявність відкритого API для інтеграції з іншими системами, це відкриває можливості для подальшого розширення функціоналу, наприклад, для створення мобільного додатка або синхронізації з розкладом занять.

Таким чином, розроблений WEBресурс обліку відвідування занять студентами поєднує в собі переваги безкоштовності, доступності, багатофункціональності та зручності використання, що робить його оптимальним вибором для автоматизації обліку відвідуваності в закладах освіти. Розробка не лише усуває недоліки існуючих аналогів, але й пропонує додаткові можливості, які відсутні в інших системах, зокрема динамічний підрахунок статистики, фільтрацію на стороні клієнта та розширену аналітику для студентів.

Висновки. На основі аналізу існуючих програмних рішень спрямованих на облік відвідування занять студентів встановлено, що значна частина з них має обмежений функціонал, відсутність україномовного інтерфейсу, платний доступ або прив'язку до конкретного навчального закладу. Саме це слугувало поштовхом для розробки власного WEBресурсу обліку відвідування занять студентами. Для реалізації WEBресурсу були використані сучасні технології та мови програмування. Серверна частина реалізована на платформі Node.js з використанням фреймворку Express, це забезпечує стабільність роботи, продуктивність та можливість масштабування системи. Клієнтська частина побудована з використанням HTML5, CSS3 та JavaScript із застосуванням шаблонізатора EJS для серверного рендерингу сторінок, що гарантує швидке

завантаження та зручний інтерфейс. Для зберігання та обробки даних використано реляційну базу даних SQLite, яка забезпечує надійність, цілісність даних та простоту розгортання. Як середовище виконання обрано WEB-браузери, це дозволяє використовувати розроблений ресурс без необхідності встановлення додаткового програмного забезпечення.

У порівнянні з аналогами розроблений WEBресурс забезпечує україномовний інтерфейс, повну безкоштовність, доступність для будь-якого закладу освіти, генерацію звітів у форматах PDF, Word та Excel, надання студентам деталізованої статистики відвідуваності, розмежування прав доступу на основі ролей, простоту впровадження, фільтрацію даних на стороні клієнта та наявність відкритого API для інтеграції. Мета роботи, яка полягала в розширенні функціональних можливостей WEBресурсу обліку відвідування занять студентами, була досягнута за рахунок упровадження таких 4 додаткових можливостей: динамічного підрахунку статистики в реальному часі під час відмітки присутності; генерації звітів у трьох форматах з попереднім переглядом; надання студентам доступу до персоніфікованої статистики з візуальними індикаторами; реалізації відкритого API для подальшої інтеграції з іншими системами. Реалізований WEBресурс створює основу для подальшого розвитку, зокрема для впровадження системи сповіщень, розробки мобільного додатка, інтеграції з розкладом занять та розширення аналітичних можливостей.

Література:

1. Міністерство освіти і науки України. Інструкція з ведення ділової документації у закладах загальної середньої освіти в електронній формі [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/z1029-22#Text>.
2. Розвиток інформаційних систем управління освітою як інструмент реалізації державної освітньої політики: монографія [Електронний ресурс] / С.Л. Лондар, А.О. Литвинчук, Г.М. Терещенко, А.В. Кир'янов, Ю.В. Іриневиц, В.В. Гапон, М.І. Шараєвська, М.В. Леснікова, С.В. Мельник. – Київ: ДНУ «Інститут освітньої аналітики», 2020. – 258 с. – Режим доступу: chrome-extension://efaidnbmninnibpcapjpcgiclfndmkaj/https://iea.gov.ua/wp-content/uploads/2025/04/monograph_2020_fin_.pdf
3. MyAttendanceTracker [Електронний ресурс]. – Режим доступу: <https://www.myattendancetracker.com/>.
4. Attendance Radar [Електронний ресурс]. – Режим доступу: <https://attendanceradar.com/>.
5. JetIQ [Електронний ресурс]. – Режим доступу: https://wiki.vntu.edu.ua/uk/%D0%93%D0%BE%D0%BB%D0%BE%D0%B2%D0%BD%D0%B0_%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0.
6. Choi D. Full-Stack React, TypeScript, and Node: Build cloud-ready web applications using React 17 with Hooks and GraphQL / D. Choi. – Birmingham : Packt Publishing Ltd., 2020. – P. 624.

ISSN 2786-6025 Online

7. Express. Web framework for Node.js [Електронний ресурс]. – Режим доступу: <https://expressjs.com/>.

8. Better-sqlite3 [Електронний ресурс]. – Режим доступу: <https://www.npmjs.com/package/better-sqlite3>.

9. Object Management Group. «Unified Modeling Language (UML) Specification», Version 2.5.1, 2017 [Електронний ресурс]. – Режим доступу: <https://www.omg.org/spec/UML/>.

10. Основи об'єктно-орієнтованого програмування : навч. посіб. / О. В. Щербаков, Ю. С. Парфьонов, В. М. Федорченко. – Харків : ХНЕУ ім. С. Кузнеця, 2019. – 237 с.

11. Architecting Modern Web Applications with ASP.NET Core and Azure [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/>.

12. Generate PDFs from a template using JavaScript [Електронний ресурс]. – Режим доступу: <https://github.com/PSPDFKit-labs/pspdfkit-template-generation-from-pdf>.

References:

1. Ministerstvo osvity i nauky Ukrainy. Instruktsiia z vedennia dilovoi dokumentatsii u zakladakh zahalnoi serednoi osvity v elektronnyi formi [Ministry of Education and Science of Ukraine. Instruction on maintaining business documentation in general secondary education institutions in electronic form]. zakon.rada.gov.ua/laws/show/z1029-22#Text. Retrieved from <https://zakon.rada.gov.ua/laws/show/z1029-22#Text> [in Ukrainian].

2. Londar, S.L., Lytvynchuk, A.O., Tereshchenko, H.M., Kyrianov, A.V., Irynevych, Yu.V., Hapon, V.V., Sharaievska, M.I., Liesnikova, M.V., & Melnyk, S.V. (2020). Rozvytok informatsiinykh system upravlinnia osvitoiu yak instrument realizatsii derzhavnoi osvithoi polityky [Development of education management information systems as a tool for the implementation of state educational policy: monograph]. Kyiv: DNU «Instytut osvithoi analityky». Retrieved from https://iea.gov.ua/wp-content/uploads/2025/04/monograph_2020_fin_.pdf [in Ukrainian].

3. MyAttendanceTracker. Retrieved from <https://www.myattendancetracker.com/>.

4. Attendance Radar. Retrieved from <https://attendanceradar.com/>.

5. JetIQ [JetIQ]. wiki.vntu.edu.ua/uk/Головна_сторінка. Retrieved from https://wiki.vntu.edu.ua/uk/Головна_сторінка [in Ukrainian].

6. Choi, D. (2020). Full-Stack React, TypeScript, and Node: Build cloud-ready web applications using React 17 with Hooks and GraphQL. Birmingham: Packt Publishing Ltd.

7. Express. Web framework for Node.js. Retrieved from <https://expressjs.com/>.

8. Better-sqlite3. Retrieved from <https://www.npmjs.com/package/better-sqlite3>.

9. Object Management Group. «Unified Modeling Language (UML) Specification», Version 2.5.1, 2017. Retrieved from <https://www.omg.org/spec/UML/>.

10. Shcherbakov, O.V., Parfonov, Yu.Ye., & Fedorchenko, V.M. (2019). Osnovy obiektno-orientovanoho prohramuvannia [Fundamentals of object-oriented programming]. Kharkiv: KhNEU im. S. Kuznetsia [in Ukrainian].

11. Architecting Modern Web Applications with ASP.NET Core and Azure. Retrieved from <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/>.

12. Generate PDFs from a template using JavaScript. [Retrieved from <https://github.com/PSPDFKit-labs/pspdfkit-template-generation-from-pdf>].

Дата першого надходження статті до видання: 12.03.2026

Дата прийняття статті до друку після рецензування: 26.03.2026