

Міністерство освіти і науки України
Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

Кваліфікаційна робота
за освітнім ступенем “бакалавр”
на тему:
СИСТЕМА ЕЛЕКТРОННОЇ ВЗАЄМОДІЇ
КУРАТОРА ТА СТУДЕНТІВ

Виконала:

здобувач IV курсу
групи КН-41
спеціальності 122 “Комп'ютерні
науки”

Анастасія Русланівна Карпюк

Науковий керівник:

кандидат технічних наук,
доцент Степанія Михайлівна Бабич

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	5
1.1. Аналіз існуючих платформ для електронної взаємодії.....	5
1.2. Потреба у власному вебінтерфейсі	6
РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ.....	9
2.1. Вимоги до системи.....	9
2.2. Модель даних.....	14
2.3. Архітектура системи.....	19
РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ.....	24
3.1. Технології для розробки.....	24
3.2. Реалізація компонентів	24
3.3. Реалізація групового чату взаємодії куратора та студентів	28
3.4. Розгортання та запуск системи.....	31
РОЗДІЛ 4. ТЕСТУВАННЯ ТА ОЦІНКА	34
4.1. Методи тестування.....	34
4.2. Результати тестування	34
4.3. Тестування безпеки системи.....	44
4.4. Оцінка продуктивності системи	44
ВИСНОВКИ.....	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	47
ДОДАТКИ.....	Помилка! Закладку не визначено.
Додаток А. Файл конфігурації.....	Помилка! Закладку не визначено.
Додаток Б. Система авторизації	Помилка! Закладку не визначено.
Додаток В. Кабінет куратора	60
Додаток Г. Кабінет студента.....	Помилка! Закладку не визначено.
Додаток Д. Індивідуальні повідомлення.....	Помилка! Закладку не визначено.
Додаток Е. Груповий чат.....	Помилка! Закладку не визначено.

Додаток Ж. Структура бази даних інформаційної системи **Помилка!**
Закладку не визначено.

ВСТУП

Сучасний освітній процес вимагає ефективних засобів комунікації між викладачами та студентами. Особливу актуальність це питання набуває в контексті розвитку дистанційної та змішаної форм навчання, де якісний обмін інформацією стає одним із ключових чинників успішності навчального процесу. Традиційні методи комунікації, такі як електронна пошта чи месенджери, часто виявляються недостатньо ефективними для організації структурованого діалогу між кураторами навчальних груп та студентами.

Актуальність роботи зумовлена необхідністю створення спеціалізованого програмного рішення, яке б забезпечувало цілеспрямовану та організовану комунікацію в освітньому середовищі. Існуючі системи управління навчанням (LMS) не рідко мають надлишковий функціонал для простих завдань обміну повідомленнями, тоді як уніфіковані системи листування не враховують специфіку освітнього процесу.

Мета дослідження полягає в розробці та впровадженні веборієнтованої системи управління повідомленнями між кураторами та студентами, яка б забезпечувала ефективну, безпечну та зручну комунікацію в рамках закладу освіти.

Основні завдання дослідження:

1. проаналізувати існуючі рішення та обґрунтувати доцільність розроблення системи;
2. сформулювати вимоги до системи та визначити її функціональні модулі;
3. спроектувати модель даних і архітектуру вебзастосунку;
4. реалізувати серверну та клієнтську частини системи, включаючи модуль групового чату;
5. забезпечити механізми захисту та логування подій;

6. виконати тестування основних сценаріїв роботи системи.

Об'єкт дослідження – процес організації комунікації між учасниками освітнього процесу в умовах сучасного закладу вищої освіти.

Предмет дослідження – методи та засоби розробки спеціалізованих вебсистем для управління індивідуальними та груповими повідомленнями в освітньому середовищі.

У кваліфікаційній роботі використано комплекс загальнонаукових та спеціальних **методів дослідження**, зокрема: аналіз і узагальнення наукових джерел та існуючих платформ електронної взаємодії; порівняльний аналіз; методи системного аналізу та проектування; методи моделювання; методи програмної інженерії; методи тестування програмного забезпечення (функціональне, тестування безпеки).

Практичне значення дослідження. Розроблена система може бути впроваджена в навчальних закладах для підвищення ефективності комунікації між учасниками освітнього процесу, оперативного інформування академічних груп, організації групових обговорень та збереження історії листування.

Апробація і впровадження результатів дослідження. Основні теоретичні та практичні результати дослідження доповідалися та обговорювалися на звітній науковій конференції викладачів, співробітників і здобувачів вищої освіти Рівненського державного гуманітарного університету за 2025 рік (м. Рівне, 14 травня 2026 року), на Всеукраїнській науково-практичній Інтернет-конференції «Автоматизація та комп'ютерно-інтегровані технології у виробництві й освіті: стан, досягнення, перспективи розвитку» (м. Черкаси, 11-12 травня 2026 року).

Структура роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та семи додатків. Основний обсяг роботи становить 46 сторінок. Список використаних джерел включає 20 найменувань.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз існуючих платформ для електронної взаємодії

Електронна взаємодія між куратором і студентами в умовах дистанційного або гібридного навчання зазвичай реалізується через LMS-платформи. Такі системи надають вебінтерфейс, базу даних користувачів і навчальних матеріалів, а також інструменти комунікації (повідомлення, коментарі, оголошення). Для кураторства групи особливо важливими є: швидкий обмін повідомленнями, фіксація історії спілкування та контроль доступу користувачів.

У більшості LMS-платформ інструменти комунікації реалізовані у вигляді форумів, коментарів або оголошень, що не завжди забезпечує оперативність обміну повідомленнями в межах академічної групи. Механізми групових чатів або відсутні, або інтегровані у вигляді сторонніх сервісів, що ускладнює централізоване управління комунікацією.

Moodle відкрита LMS, орієнтована на керування курсами й навчальним контентом. Переваги: гнучкі ролі доступу, розширюваність через модулі, наявність форумів, повідомлень і звітів. Недоліки для задачі «куратор студенти»: надлишковий функціонал і складніше адміністрування в межах невеликого проєкту [1].

Canvas LMS хмарна LMS із сучасним інтерфейсом та інструментами для комунікації і спільної роботи. Переваги: швидкий старт, інтеграції та мобільна підтримка; недоліки: залежність від інтернет-з'єднання й комерційна модель для розширених можливостей [2].

Blackboard Learn комерційна платформа для великих закладів освіти з розвиненою аналітикою та інтеграціями. Переваги: масштабованість і звітність; недоліки: висока вартість, складність інтерфейсу та надмірність для невеликої системи кураторської взаємодії.

Google Classroom – сервіс у складі Google Workspace for Education, зручний для швидкої організації навчального процесу. Переваги: простота використання, інтеграція з Google Drive та Meet; недоліки: обмежена кастомізація й прив’язка до Google-акаунтів, що ускладнює впровадження в окремих корпоративних сценаріях.

Отже, наявні платформи здебільшого орієнтовані на керування навчальним контентом і організацію курсів, а не на цільову комунікацію формату «куратор ↔ академічна група» із мінімально необхідним функціоналом. Це обґрунтовує доцільність розроблення власного вебзастосунку, у якому індивідуальні повідомлення доповнюються механізмом групового чату, що забезпечує швидке інформування, колективне обговорення та збереження історії взаємодії.

Порівняльний аналіз функціональних можливостей розглянутих платформ електронної взаємодії наведено в табл. 1.1.

Таблиця 1.1

Порівняння платформ електронної взаємодії

Критерій	Moodle	Canvas LMS	Blackboard Learn	Google Classroom
Індивідуальні повідомлення	✓	✓	✓	✓
Групові чати	Частково	Частково	✓	Обмежено
Контроль доступу користувачів	✓	✓	✓	✓
Логуювання дій користувачів	✓	✓	✓	Частково
Кастомізація функціоналу	✓	✓	✓	Обмежено
Автономне розгортання	✓	✗	✓	✗
Інтеграція з іншими системами	✓	✓	✓	✓
Орієнтація на кураторську взаємодію	✗	✗	✗	✗

Як видно з таблиці, жодна з розглянутих платформ не орієнтована виключно на кураторську взаємодію з акцентом на оперативний обмін повідомленнями та централізоване управління груповими комунікаціями. Це підтверджує доцільність розроблення спеціалізованого вебзастосунку.

1.2. Потреба у власному вебінтерфейсі

У сучасних умовах навчального процесу важливим є швидкий, зручний та безпечний обмін індивідуальними та груповими повідомленнями між кураторами та студентами. Особливо це актуально для кураторів груп, які постійно потребують передавати важливі повідомлення: про зміну розкладу, проведення заходів, необхідність подання документів чи особисті рекомендації. Традиційні способи комунікації такі як електронна пошта, месенджери або усне оповіщення мають ряд недоліків: низька формалізація, відсутність архіву, можливість втрати повідомлень, відсутність контролю за прочитанням тощо.

На цьому тлі виникає потреба у створенні спеціалізованого вебінтерфейсу, що забезпечує структуровану, централізовану та автономну систему обміну повідомленнями всередині навчального закладу. Такий інтерфейс дозволяє уникнути залежності від сторонніх сервісів, забезпечує конфіденційність даних та дає можливість точково керувати взаємодією між користувачами різних рівнів доступу.

Створення власного вебінтерфейсу обґрунтоване наступними факторами.

1. Контроль над даними. У власній системі адміністратор має повний контроль над базою даних, правами доступу, логуванням активності та безпекою. Це важливо для дотримання політики конфіденційності особистої інформації студентів.

2. Адаптивність до потреб установи. Готові рішення (наприклад, Telegram чи Google Classroom) часто не враховують специфіки внутрішніх процесів конкретного навчального закладу. Власний інтерфейс можна легко модифікувати: додавати нові ролі, типи повідомлень, функції відповідей, сповіщень тощо.

3. Інтеграція з іншими системами. У майбутньому така система може бути інтегрована з електронним журналом, системою реєстрації на заняття або порталом студентів, що забезпечить єдиний цифровий простір навчального закладу.

4. Зручність та інтуїтивність. Простий і зрозумілий вебінтерфейс дозволяє користувачам будь-якого рівня підготовки швидко освоїти систему. Відсутність зайвих функцій сприяє швидкій навігації та ефективній комунікації.

5. Автономність та доступність. Вебінтерфейс працює через браузер, не потребує встановлення додатків і доступний з будь-якого пристрою з підключенням до Інтернету. Це забезпечує сталу взаємодію між кураторами та студентами незалежно від місця перебування.

6. Формалізація комунікації. Кожне повідомлення зберігається в базі даних із вказівкою дати, часу та відправника. Це створює історію взаємодії, яку можна використовувати для аналізу, звітності або роз'яснювальних цілей.

7. Підтримка групових комунікацій. Наявність групових чатів дозволяє одночасно інформувати всіх учасників академічної групи, організовувати обговорення та забезпечувати збереження історії колективної взаємодії.

Отже, розробка власного вебінтерфейсу для обміну повідомленнями між куратором та студентами є актуальним рішенням, яке підвищує ефективність управління навчальним процесом, забезпечує надійність комунікації та відповідає сучасним вимогам до цифровізації освіти.

РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ

2.1. Вимоги до системи

Функціональні вимоги

FR1. Система авторизації та управління користувачами

- FR1.1 Реєстрація нових користувачів з розподілом за ролями
- FR1.2 Вхід до системи за логіном та паролем
- FR1.3 Розподіл користувачів на ролі: "Куратор" та "Студент"
- FR1.4 Захист паролів з використанням хешування
- FR1.5 Вихід з системи

FR2. Функціонал кабінету куратора

- FR2.1 Перегляд списку закріплених студентів
- FR2.2 Надсилання повідомлень конкретним студентам
- FR2.3 Перегляд історії надісланих повідомлень
- FR2.4 Перегляд відповідей від студентів
- FR2.5 Фільтрація повідомлень за студентами
- FR2.6 Позначення повідомлень як прочитаних

FR3. Функціонал кабінету студента

- FR3.1 Перегляд отриманих повідомлень від куратора
- FR3.2 Відповідь на повідомлення куратора
- FR3.3 Перегляд історії листування
- FR3.4 Відображення дати та часу повідомлень

FR4. Управління повідомленнями

- FR4.1 Зберігання повідомлень у базі даних
- FR4.2 Відстеження статусу повідомлень (прочитано/не прочитано)
- FR4.3 Відображення часу відправки та отримання
- FR4.4 Обмеження довжини

FR5. Функціонал групового чату

- FR5.1 Створення групового чату куратором

- FR5.2 Автоматичне формування складу чату на основі академічної групи
- FR5.3 Відображення учасників групового чату
- FR5.4 Перевірка належності користувача до чату перед доступом
- FR5.5 Надсилання повідомлень у груповий чат
- FR5.6 Перегляд історії повідомлень чату
- FR5.7 Відображення автора, дати та часу повідомлення
- FR5.8 Автоматичне збереження повідомлень у базі даних

Вимоги до інтерфейсу користувача

UR1. Загальні вимоги до інтерфейсу

- UR1.1 Інтуїтивно зрозумілий та простий у використанні інтерфейс
- UR1.2 Адаптивний дизайн для різних розмірів екранів
- UR1.3 Чітка навігація між розділами
- UR1.4 Візуальне розрізнення ролей користувачів

UR2. Сторінка авторизації

- UR2.1 Форма входу з полями логіну та пароля
- UR2.2 Вибір типу облікового запису (куратор/студент)
- UR2.3 Посилання на сторінку реєстрації
- UR2.4 Валідація введених даних

UR3. Сторінка реєстрації

- UR3.1 Форма реєстрації з обов'язковими полями
- UR3.2 Динамічне відображення полів залежно від ролі
- UR3.3 Вибір куратора для студентів
- UR3.4 Перевірка унікальності логіну та email

UR4. Кабінет куратора

- UR4.1 Панель статистики (кількість повідомлень, відповідей)
- UR4.2 Форма відправки нового повідомлення
- UR4.3 Список студентів з можливістю фільтрації
- UR4.4 Історія повідомлень у форматі чату
- UR4.5 Індикатори нових відповідей

UR5. Кабінет студента

- UR5.1 Інформація про профіль студента
- UR5.2 Список отриманих повідомлень
- UR5.3 Форма відповіді на повідомлення
- UR5.4 Візуальні індикатори статусу повідомлень

UR6. Інтерфейс групового чату

- UR6.1 Відображення списку доступних групових чатів
- UR6.2 Інтерфейс переписки у форматі діалогу
- UR6.3 Відображення повідомлень у хронологічному порядку
- UR6.4 Індикатори нових повідомлень
- UR6.5 Поле введення повідомлення
- UR6.6 Кнопка надсилання повідомлення

Вимоги до безпеки

SR1. Аутентифікація та авторизація

- SR1.1 Захист паролів з використанням алгоритму хешування bcrypt
- SR1.2 Перевірка прав доступу для кожного запиту
- SR1.3 Захист від несанкціонованого доступу до кабінетів
- SR1.4 Автоматичний вихід при закритті браузера

SR2. Захист даних

- SR2.1 Захист від SQL–ін'єкцій через prepared statements
- SR2.2 Валідація та санація вхідних даних
- SR2.3 Захист від XSS атак
- SR2.4 Перевірка сесійної автентифікації користувача перед виконанням дій
- SR2.5 Перевірка належності користувача до групового чату перед відображенням повідомлень

SR3. Захист від атак

- SR3.1 Обмеження кількості спроб входу
- SR3.2 Блокування доступу після перевищення допустимої кількості невдалих спроб входу

- SR3.3 Логування подій безпеки

Вимоги до бази даних

DR1. Структура бази даних

- DR1.1 Таблиця кураторів (curators)
- DR1.2 Таблиця студентів (students)
- DR1.3 Таблиця повідомлень (messages)
- DR1.4 Таблиця групових чатів (group_chats)
- DR1.5 Таблиця учасників групового чату (group_chat_members)
- DR1.6 Таблиця повідомлень групового чату (group_chat_messages)
- DR1.7 Таблиця спроб входу (login_attempts)
- DR1.8 Таблиця системних логів (system_logs)
- DR1.9 Таблиця системних налаштувань (system_settings)

DR2. Вимоги до цілісності даних

- DR2.1 Зовнішні ключі для підтримки реляційних зв'язків
- DR2.2 Обмеження унікальності для логінів та email
- DR2.3 Автоматичне оновлення timestamp полів
- DR2.4 Каскадне видалення/оновлення залежних записів

DR3. Продуктивність

- DR3.1 Індокси для полів, що використовуються в пошуку
- DR3.2 Оптимізація запитів для швидкого відгуку
- DR3.3 Регулярне очищення тимчасових даних

Технічні вимоги

TR1. Серверне середовище

- TR1.1 PHP версії 8.0 або вище
- TR1.2 MySQL/MariaDB версії 10.4 або вище
- TR1.3 Підтримка HTTPS
- TR1.4 Можливість розгортання на стандартному хостингу

TR2. Клієнтська частина

- TR2.1 Підтримка сучасних браузерів (Chrome, Firefox, Safari, Edge)
- TR2.2 Адаптивний дизайн для мобільних пристроїв

- TR2.3 Підтримка JavaScript для покращення UX
- TR2.4 Крос-браузерна сумісність

TR3. Продуктивність

- TR3.1 Час завантаження сторінок не більше 3 секунд
- TR3.2 Обробка запитів до БД не більше 1 секунди
- TR3.3 Оптимізація зображень та стилів
- TR3.4 Кешування статичних ресурсів

Вимоги до обслуговування

MR1. Адміністрування

- MR1.1 Можливість перегляду системних логів
- MR1.2 Керування обліковими записами користувачів
- MR1.3 Моніторинг активності системи
- MR1.4 Резервне копіювання бази даних

MR2. Масштабованість

- MR2.1 Можливість додавання нових функцій
- MR2.2 Підтримка збільшення кількості користувачів
- MR2.3 Модульна архітектура для легкого розширення

Вимоги до документації

DocR1. Технічна документація

- DocR1.1 Опис архітектури системи
- DocR1.2 Інструкція з встановлення та налаштування
- DocR1.3 Опис структури бази даних

DocR2. Користувацька документація

- DocR2.1 Інструкція для кураторів
- DocR2.2 Інструкція для студентів

Нефункціональні вимоги

Система повинна забезпечувати стабільність роботи, швидкий час відгуку інтерфейсу, безпечне зберігання даних користувачів, а також можливість подальшого масштабування функціоналу без порушення існуючої архітектури.

2.2. Модель даних

Модель даних наведено на рис. 2.1.

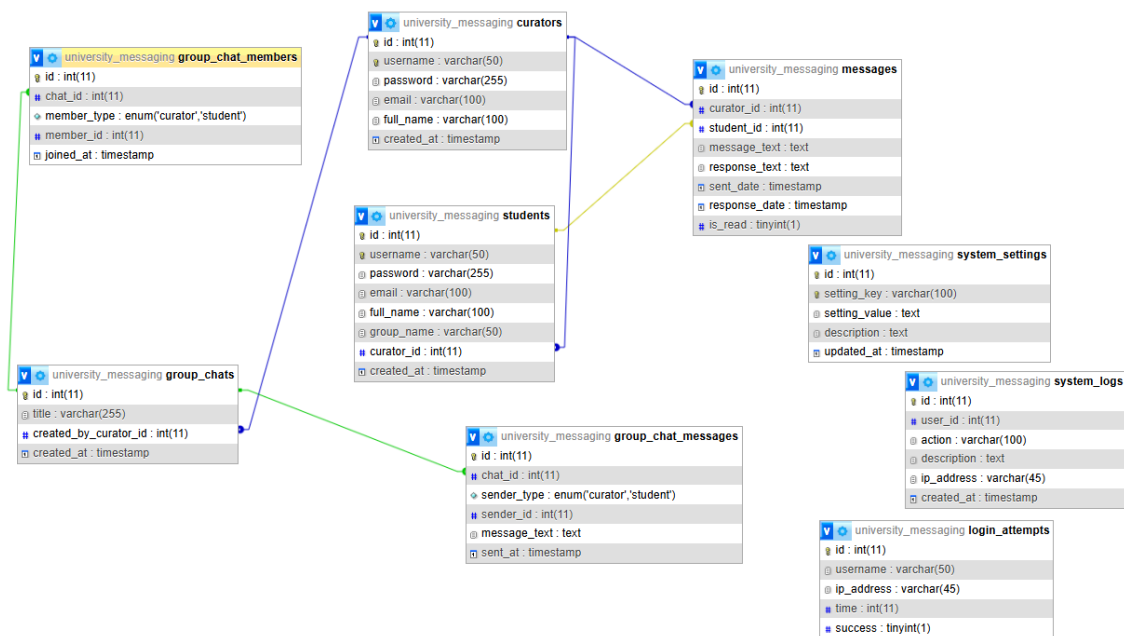


Рисунок 2.1. Модель даних

Сутність: curators (Куратори)

Призначення: зберігання інформації про кураторів.

Атрибути:

- id унікальний ідентифікатор (PK);
- username унікальний логін;
- password хешований пароль;
- email електронна пошта;
- full_name ПІБ;
- created_at дата створення.

Зв'язки:

- один-до-багатьох зі students;
- один-до-багатьох із messages;
- один-до-багатьох із group_chats;

- один-до-багатьох із group_chat_members;
- один-до-багатьох із group_chat_messages.

Сутність: students (Студенти)

Призначення: зберігання інформації про студентів.

Атрибути:

- id унікальний ідентифікатор (PK);
- username логін;
- password пароль;
- email пошта;
- full_name ПІБ;
- group_name навчальна група;
- curator_id FK → curators.id;
- created_at дата створення.

Зв'язки:

- багато-до-одного з curators;
- один-до-багатьох із messages;
- один-до-багатьох із group_chat_messages;
- один-до-багатьох із group_chat_members;

Сутність: messages (Повідомлення)

Призначення: особисте листування куратор студент.

Атрибути:

- id унікальний ідентифікатор (PK);
- curator_id FK;
- student_id FK;
- message_text текст повідомлення;
- response_text відповідь;
- sent_date дата відправки;
- response_date дата відповіді;
- is_read статус прочитання.

Зв'язки:

- багато-до-одного з curators;
- багато-до-одного зі students.

Сутність: group_chats (Групові чати)

Призначення: зберігання інформації про групові діалоги.

Атрибути:

- id унікальний ідентифікатор (PK);
- title назва чату;
- created_by_curator_id FK → curators.id;
- created_at дата створення.

Зв'язки:

- багато-до-одного з curators;
- один-до-багатьох із group_chat_messages;
- один-до-багатьох із group_chat_members.

Сутність: group_chat_messages (Повідомлення групового чату)

Призначення: зберігання повідомлень у групових чатах.

Атрибути:

- id унікальний ідентифікатор (PK);
- chat_id FK → group_chats.id;
- sender_type тип відправника (student / curator);
- sender_id ID відправника;
- message_text текст повідомлення;
- sent_at дата та час відправлення.

Зв'язки:

- багато-до-одного з group_chats;
- логічний зв'язок із сутностями students та curators через поля sender_type і sender_id.

Сутність: group_chat_members (Учасники чатів)

Призначення: зберігання складу учасників групових чатів.

Атрибути:

- id унікальний ідентифікатор (PK);

- chat_id FK → group_chats.id;
- member_type student / curator;
- member_id ID учасника;
- joined_at дата приєднання.

Зв'язки:

- багато-до-одного з group_chats;
- багато-до-одного зі students;
- багато-до-одного з curators.

Первинна унікальність: Унікальність участі користувача в чаті забезпечується на логічному рівні застосунку шляхом перевірки відсутності дублювання записів із однаковими значеннями chat_id, member_id та member_type.

Сутність: login_attempts (Спроби входу)

Призначення: Відстеження спроб входу для захисту від brute-force атак

Атрибути:

- id – унікальний ідентифікатор (PK);
- username – Ім'я користувача, для якого виконувалась спроба;
- ip_address – IP-адреса, з якої виконувалась спроба;
- time – Час спроби входу (Unix timestamp);
- success – Результат спроби (успішна/невдала).

Індекси:

- idx_username_time – Для перевірки кількості спроб за період;
- idx_ip_time – Для блокування IP-адрес.

Сутність: system_logs (Системні логи)

Призначення: Логування дій користувачів та системних подій

Атрибути:

- id – унікальний ідентифікатор (PK);
- user_id – Ідентифікатор користувача (якщо дія виконувалась авторизованим);
- action – Тип виконаної дії;

- description – Детальний опис події;
- ip_address – IP-адреса, з якої виконувалась дія;
- created_at – Час реєстрації події.

Сутність: system_settings (Системні налаштування)

Призначення: зберігання конфігураційних параметрів системи.

Атрибути:

- id унікальний ідентифікатор (PK);
- setting_key унікальний ключ налаштування;
- setting_value значення налаштування;
- description опис призначення налаштування;
- updated_at час останнього оновлення.

Таблиця використовується для централізованого зберігання параметрів конфігурації системи, зокрема налаштувань безпеки, параметрів сесій, системних обмежень, службових констант та інших змінних, що впливають на функціонування програмного застосунку. Застосування окремої сутності для системних налаштувань забезпечує гнучкість адміністрування та можливість зміни конфігурації без модифікації програмного коду.

Бізнес-правила та обмеження

Правила цілісності даних:

1. Кожен студент може мати призначеного куратора відповідно до організаційної структури навчального закладу.
2. Повідомлення може бути відправлене лише авторизованим куратором зареєстрованому студенту.
3. Логіни та email повинні бути унікальними в межах всієї системи.
4. Паролі зберігаються у хешованому вигляді.
5. Груповий чат може створити лише куратор.
6. Повідомлення в груповий чат можуть надсилати лише його учасники.
7. Користувач може бути учасником кількох групових чатів.
8. Склад учасників групового чату формується на основі академічної групи студентів, закріплених за куратором.

Обмеження:

- Максимальна довжина повідомлення: 1000 символів
- Мінімальна довжина пароля: 6 символів
- Формат email повинен відповідати стандарту RFC 5322

Автоматизація на рівні бази даних:

1. Автоматичне встановлення `sent_date` при створенні повідомлення
2. Автоматичне встановлення `created_at` у сутностях користувачів
3. Автоматичне оновлення `updated_at` у `system_settings`

2.3. Архітектура системи

Система управління повідомленнями між кураторами та студентами реалізована за багаторівневою архітектурою (multi-tier architecture), що забезпечує чітке розділення функцій між компонентами та спрощує масштабування і супровід програмного забезпечення. Архітектура побудована на принципах MVC (Model–View–Controller) і включає рівень представлення, рівень бізнес-логіки та рівень доступу до даних.

Система функціонує за клієнт-серверною моделлю взаємодії (рис. 2.2). Клієнтський рівень реалізований у вигляді вебінтерфейсу, що працює у браузері користувача. Відображення даних та взаємодія з інтерфейсом забезпечуються технологіями HTML, CSS та JavaScript, які відповідають за формування сторінок, маніпуляцію DOM-структурою та обробку подій користувача.

На серверному рівні розміщено рівень представлення, який реалізований у вигляді PHP-контролерів та HTML-шаблонів. Контролери обробляють HTTP/HTTPS-запити, виконують виклик відповідних бізнес-модулів і генерують динамічні вебсторінки для подальшого відображення у браузері.

Рівень бізнес-логіки включає функціональні модулі системи, зокрема модуль авторизації, модуль користувачів, модуль повідомлень та модуль групового чату. Модуль авторизації забезпечує реєстрацію, вхід до системи та

управління сесіями користувачів. Модуль користувачів відповідає за роботу з профілями кураторів і студентів. Модуль повідомлень реалізує індивідуальне листування між учасниками освітнього процесу. Модуль групового чату забезпечує створення груп, управління списком учасників і обмін повідомленнями в межах спільних чатів.

Окремо виділено модуль безпеки, який використовується всіма бізнес-компонентами та відповідає за перевірку прав доступу, валідацію даних і захист від типових вебзагроз.

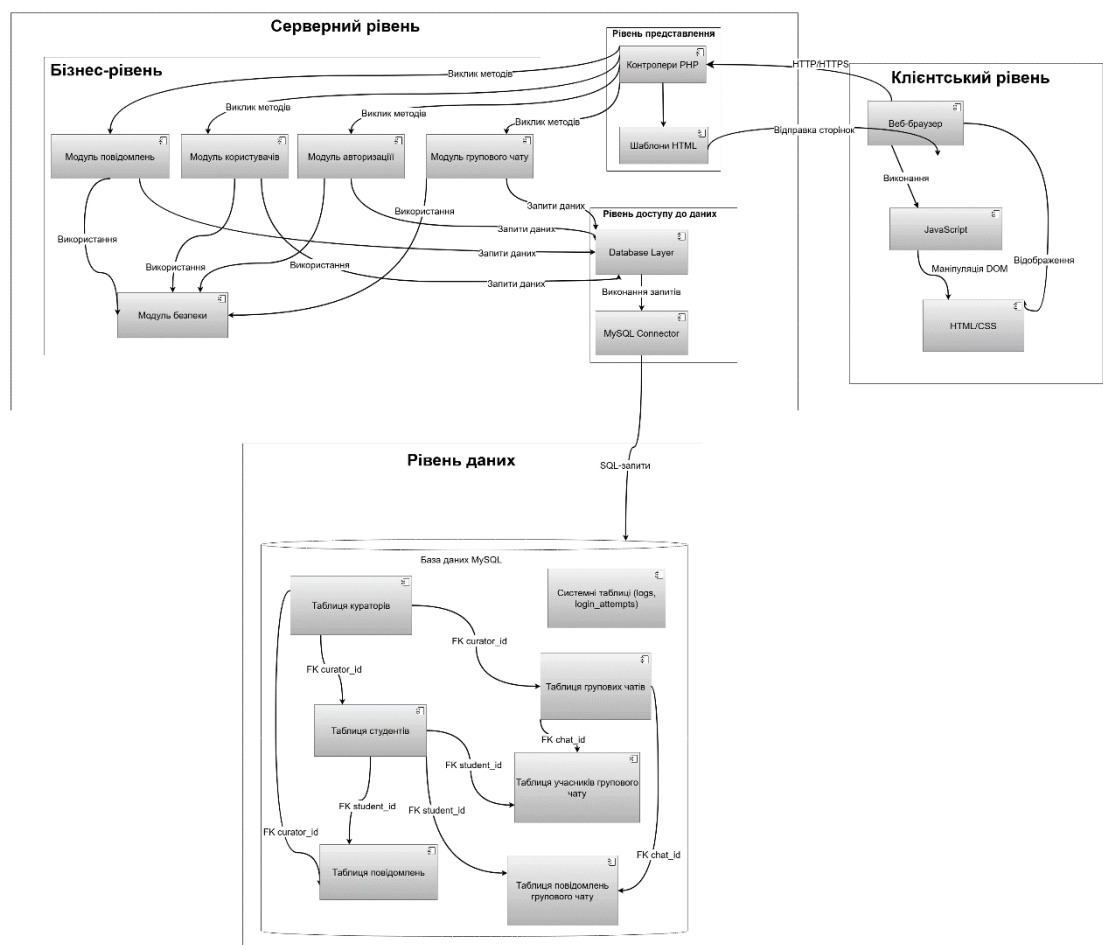


Рисунок 2.2. Діаграма архітектури системи

Рівень доступу до даних представлений спеціалізованим шаром роботи з базою даних (Database Layer), який виконує обробку запитів і взаємодіє з СУБД через розширення MySQLi. Такий підхід забезпечує ізоляцію бізнес-логіки від фізичної структури зберігання даних.

На рівні даних розташована база даних MySQL, у якій зберігається інформація про користувачів, повідомлення, групові чати, учасників чатів та службові журнали системи. Взаємодія між серверною частиною та базою даних здійснюється за допомогою SQL-запитів.

Запропонована архітектура забезпечує модульність, розширюваність і надійність системи, а також створює основу для подальшого розвитку функціональних можливостей платформи.

Система складається з взаємопов'язаних функціональних компонентів, взаємодія яких відображена на діаграмі діяльності процесу авторизації та подальшої роботи користувача із груповим чатом.

Процес розпочинається з ініціації входу користувача до системи. На рівні вебінтерфейсу відображається форма авторизації, після заповнення якої облікові дані передаються до контролера авторизації. На серверному рівні виконується валідація вхідних даних і перевірка CSRF-токена, що запобігає несанкціонованим запитам.

Модуль безпеки додатково здійснює перевірку brute-force захисту та валідацію пароля. Після цього відбувається звернення до бази даних, де виконується пошук користувача та оновлення параметра `last_login` у разі успішної автентифікації.

Якщо авторизація пройшла успішно, користувач перенаправляється до особистого кабінету, де отримує доступ до функціоналу системи. Зокрема передбачено відкриття групового чату, перегляд історії повідомлень і роботу з ними. Дані повідомлень отримуються із бази даних і відображаються у вебінтерфейсі.

У разі помилки авторизації система формує повідомлення про помилку та надає можливість повторної спроби входу.

Таким чином, діаграма демонструє повний цикл взаємодії компонентів від дій користувача у вебінтерфейсі до обробки запитів контролером, перевірок модулем безпеки та роботи з базою даних. Діаграму потоків даних наведено на рис. 2.3.

Модульна структура системи управління повідомленнями побудована за сервісно-орієнтованим підходом і включає взаємопов'язані компоненти, взаємодія яких відображена на діаграмі компонентів.

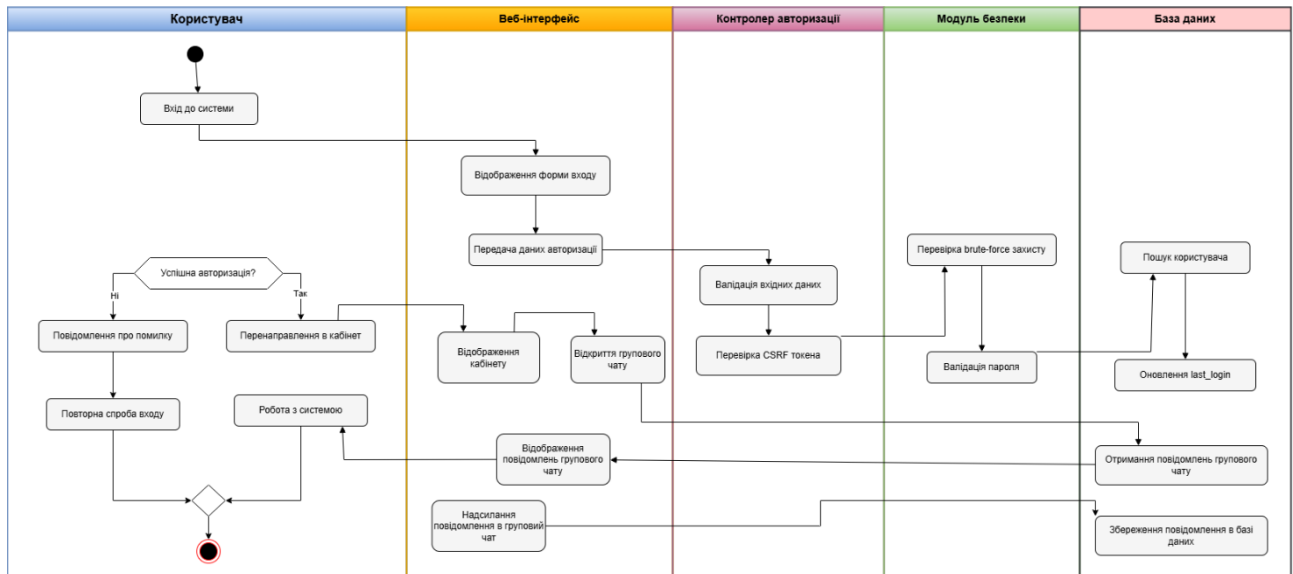


Рисунок 2.3. Діаграма потоків даних

Вхідною точкою системи виступає вебінтерфейс, який передає HTTP-запити до API Gateway. Даний компонент виконує маршрутизацію запитів і виклик відповідних методів бізнес-логіки залежно від функціонального призначення операції.

До рівня бізнес-логіки належать основні функціональні модулі системи:

- модуль авторизації, що забезпечує реєстрацію користувачів, вхід до системи та управління сесіями;
- модуль користувачів, який реалізує роботу з профілями кураторів і студентів;
- модуль повідомлень, що відповідає за індивідуальне листування, надсилання повідомлень, відстеження статусів і збереження історії;
- модуль групового чату, який забезпечує створення чатів, управління списком учасників і обмін повідомленнями в межах груп;

- модуль безпеки, інтегрований у всі бізнес-компоненти та призначений для перевірки доступу, валідації даних і контролю безпекових подій.

На рівні сервісів даних функціонують окремі сервіси:

- сервіс повідомлень;
- сервіс групових чатів;
- сервіс користувачів;
- сервіс логування.

Сервіс логування забезпечує збір і збереження інформації про дії користувачів і системні події, що дозволяє здійснювати моніторинг, аудит і аналіз роботи системи.

Модуль безпеки реалізує комплексний захист застосунку від типових вебзагроз, зокрема SQL-ін'єкцій, XSS-атак, CSRF-атак і brute-force спроб несанкціонованого доступу.

Взаємодія сервісів із базою даних здійснюється через SQL-запити. Така організація забезпечує розподіл навантаження, модульність і можливість незалежного масштабування окремих компонентів системи.

Діаграму компонентів наведено на рис. 2.4.

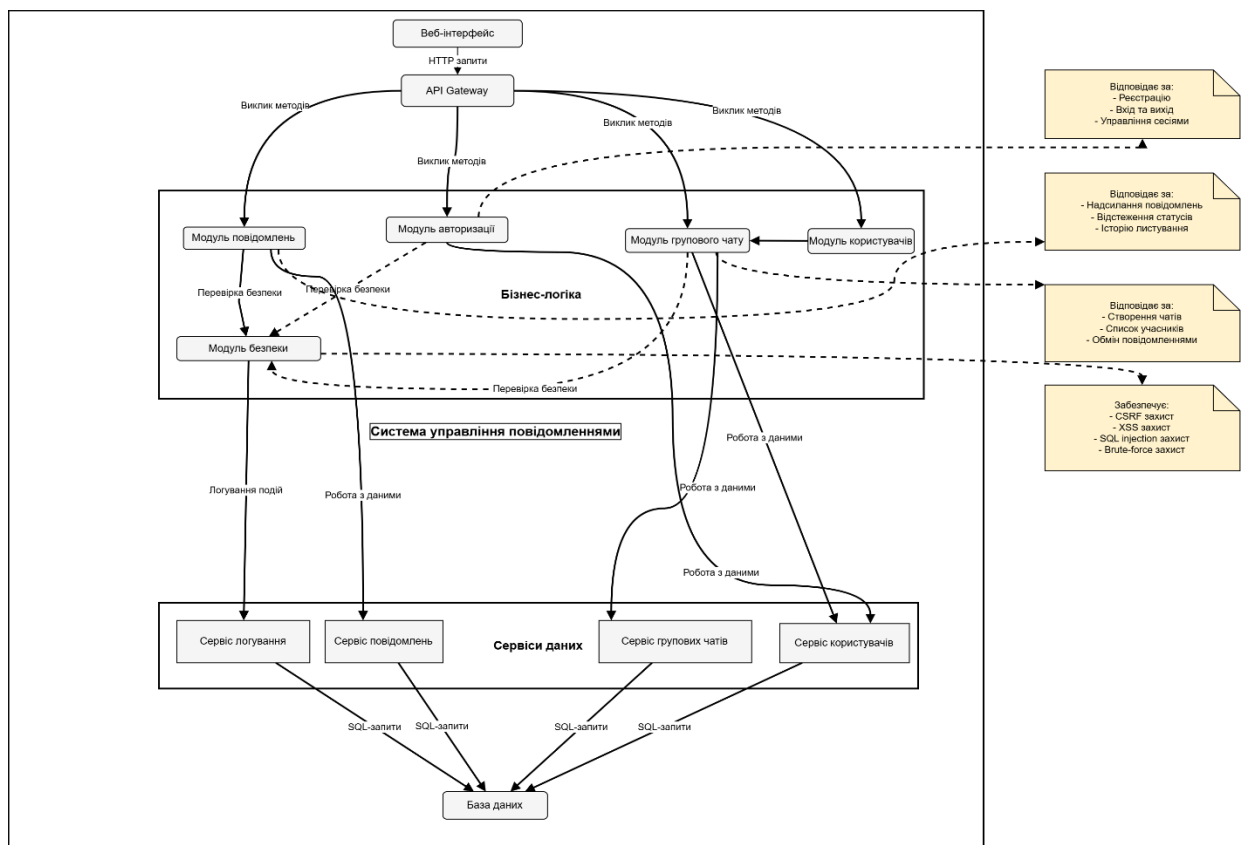


Рисунок 2.4. Діаграма компонентів

РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ

3.1. Технології для розробки

У розробці прототипу використано поширений набір вебтехнологій, що застосовується для створення навчальних і невеликих корпоративних вебпроектів та добре підтримується хостинг-середовищами.

PHP 8.0 застосовано для реалізації серверної логіки та побудови архітектури застосунку за принципами MVC-підходу (розділення на моделі, подання та контролери). Це спрощує супровід і подальше розширення функціоналу, зокрема додавання групового чату.

MySQL 5.7+ використано як реляційну систему керування базами даних для збереження інформації про користувачів, повідомлення та групи. Доступ до бази даних здійснюється через MySQLi з використанням підготовлених запитів (Prepared Statements), що знижує ризик SQL-ін'єкцій.

HTML5 забезпечує семантичну структуру сторінок, а CSS3 (Flexbox, Grid) адаптивну верстку та єдиний стиль інтерфейсу.

Vanilla JavaScript використано для реалізації інтерактивних елементів інтерфейсу, зокрема валідації форм і часткового оновлення вмісту сторінок.

Apache (у складі XAMPP) застосовано як локальний вебсервер для розробки і тестування; Git для контролю версій; Visual Studio Code як редактор коду з підтримкою PHP і зручними інструментами налагодження.

3.2. Реалізація компонентів

Реалізація системи аутентифікації була розроблена з використанням об'єктно-орієнтованого підходу. Створено клас SecureAuth, який інкапсулює всю логіку реєстрації, входу та управління сесіями. Для зберігання паролів реалізовано хешування за допомогою функції password_hash із використанням

константи `PASSWORD_DEFAULT`, що забезпечує автоматичний вибір найбільш стійкого алгоритму доступного в поточній версії PHP.

Ключовим елементом безпеки є захист від brute-force атак. Реалізовано механізм обмеження кількості спроб входу через таблицю `login_attempts`. Система відстежує спроби входу за іменем користувача та IP-адресою, блокує подальші спроби після досягнення ліміту (5 спроб за 15 хвилин). Автоматичне очищення старих записів забезпечує оптимальну продуктивність бази даних.

CSRF-захист реалізований через генерацію унікальних токенів для кожної форми. При відкритті форми генерується випадковий токен, який зберігається в сесії та додається до форми як приховане поле. При відправці форми токен перевіряється на відповідність збереженому в сесії, що запобігає міжсайтовій підробці запитів.

Система обміну повідомленнями реалізована через клас `MessageSystem`, який надає комплексний API для роботи з повідомленнями. Основний функціонал включає методи для відправки повідомлень, отримання історії листування, управління статусами повідомлень та генерації статистики.

Для відправки повідомлень реалізовано перевірку прав доступу – куратор може надсилати повідомлення лише студентам, які закріплені за ним. При створенні нового повідомлення автоматично встановлюється `timestamp` відправки, що забезпечує точне відстеження часу комунікації.

Система відповідей реалізована через оновлення існуючого запису повідомлення. Коли студент відповідає на повідомлення, поле `response_text` заповнюється текстом відповіді, а `response_date` отримує поточний час. Прапорець `is_read` дозволяє відстежувати, чи переглянув куратор відповідь студента.

Кабінет куратора реалізований у файлі `curator_dashboard.php` та включає кілька ключових секцій. Панель статистики відображає загальну кількість відправлених повідомлень, отриманих відповідей та нових повідомлень, що очікують перегляду. Інтерактивна форма відправки повідомлень забезпечує вибір студента зі списку та валідацію тексту повідомлення.

Список студентів реалізований у вигляді адаптивних карток, де відображається основна інформація про кожного студента (ПІБ, група, email) з можливістю швидкого переходу до історії листування. Історія повідомлень представлена у форматі, схожому на чат, з візуальним розділенням відправлених повідомлень та отриманих відповідей.

Кабінет студента (`student_dashboard.php`) організований навколо перегляду отриманих повідомлень та формування відповідей. Інтерфейс відображає повідомлення у хронологічному порядку з чіткими індикаторами статусу (очікує відповіді, відповідь надана). Форма відповіді інтегрована безпосередньо в картку повідомлення, що забезпечує зручний та інтуїтивний інтерфейс для комунікації.

Архітектура бази даних реалізована з дотриманням принципів нормалізації.

У системі створено основні таблиці: `curators`, `students`, `messages`, `group_chats`, `group_chat_members`, `group_chat_messages`, `login_attempts`, `system_logs` та `system_settings`.

Для оптимізації продуктивності додано індекси на поля, що часто використовуються в умовах пошуку та сортування. У таблиці `messages` створено індекси `idx_student_date` (`student_id`, `sent_date`) для швидкого пошуку повідомлень конкретного студента та `idx_curator_date` (`curator_id`, `sent_date`) для ефективного отримання історії повідомлень куратора.

Зв'язки між таблицями реалізовані через зовнішні ключі, що забезпечує цілісність даних. Наприклад, поле `curator_id` у таблиці `students` посилається на `id` у таблиці `curators`, а поля `curator_id` та `student_id` у таблиці `messages` формують зв'язки з відповідними таблицями користувачів.

Адаптивний дизайн реалізований за допомогою CSS Grid та Flexbox. Система автоматично адаптується під різні розміри екранів – від десктопних моніторів до мобільних пристроїв. На малих екранах багатостовпкові макети перетворюються на одностовпкові, що забезпечує зручний перегляд на смартфонах.

Візуальні компоненти включають картки для відображення інформації, модальні вікна для додаткових дій, сповіщення про успішні операції та попередження про помилки. Кольорова схема використовує контрастні кольори для розрізнення типів повідомлень (відправлені – сині, отримані – сірі) та статусів (нові повідомлення – червоні рамки).

JavaScript функціонал реалізований без використання зовнішніх бібліотек для мінімізації залежностей. Основний функціонал включає динамічне оновлення контенту, валідацію форм на стороні клієнта, обробку подій та покращення користувацького досвіду через плавні анімації.

Система логування реалізована на двох рівнях: логування в файл та логування в базу даних. Файлове логування використовується для технічних помилок та критичних подій, тоді як логування в базу даних застосовується для відстеження дій користувачів.

Функція `logEvent()` забезпечує стандартизований формат записів логів, що включає `timestamp`, ідентифікатор користувача (якщо доступний), тип дії та детальний опис. Це дозволяє ефективно аналізувати активність в системі та виявляти потенційні проблеми безпеки.

Моніторинг безпеки включає відстеження підозрілих дій, таких як множинні невдалі спроби входу, спроби доступу до неавторизованих ресурсів та незвичні шаблони поведінки користувачів.

Система конфігурації реалізована через централізований файл `config.php`, який містить всі критичні налаштування системи. Використання констант забезпечує типобезпеку та запобігає випадковим змінам значень під час виконання програми.

Динамічні налаштування зберігаються в таблиці `system_settings`, що дозволяє змінювати параметри системи без необхідності редагування коду. Це включає такі налаштування як максимальна довжина повідомлень, обмеження спроб входу та інші параметри, які можуть потребувати коригування під час експлуатації системи.

Централізована обробка помилок реалізована через налаштування рівня помилок та створення користувацьких обробників помилок. Система розрізняє технічні помилки (помилки бази даних, проблеми з підключенням) та бізнес–помилки (невірні дані, порушення правил).

При виникненні помилок система відображає зрозумілі повідомлення для користувачів, уникаючи технічних деталей, які можуть становити загрозу безпеці. Одночасно детальна інформація про помилку записується в логи для аналізу розробниками.

Основні фрагменти програмного коду розробленого вебзастосунку, включаючи модулі авторизації, інтерфейси користувача та структуру бази даних, з метою зручності подання винесено та наведено в додатках до кваліфікаційної роботи.

3.3. Реалізація групового чату взаємодії куратора та студентів

У межах розширення функціональних можливостей вебзастосунку електронної взаємодії куратора та студентів було реалізовано модуль групового чату. Його впровадження зумовлене потребою оперативного інформування студентів, організації колективного обговорення та централізованого збереження історії комунікації в межах академічної групи.

На відміну від індивідуального листування, груповий чат забезпечує одночасну взаємодію куратора з усіма студентами групи. Це дозволяє швидко повідомляти про зміни розкладу, організаційні заходи, терміни виконання завдань та іншу важливу інформацію. Студенти, у свою чергу, отримують можливість ставити запитання й отримувати відповіді в єдиному інформаційному середовищі.

До основних функціональних можливостей реалізованого модуля належать:

- відкриття групового чату відповідно до академічної групи;
- автоматичне створення чату куратором під час першого відкриття;

- надсилання повідомлень куратором і студентами;
- відображення історії повідомлень;
- фіксація автора та часу надсилання;
- розмежування доступу відповідно до ролей користувачів.

Сценарії використання. Система передбачає два основні сценарії роботи з груповим чатом.

Перший сценарій реалізовано для куратора. Після авторизації в системі користувач переходить до розділу «Груповий чат», де отримує можливість обрати академічну групу зі списку доступних. У разі відсутності відповідного чату система автоматично створює його. Після цього куратор може надсилати повідомлення, які стають доступними всім студентам обраної групи.

Другий сценарій передбачає роботу студента. Після входу до системи користувач відкриває розділ групового чату, де автоматично відображається чат його академічної групи. Студент має змогу переглядати історію повідомлень та надсилати власні повідомлення.

Архітектура реалізації. Модуль групового чату реалізовано в межах клієнт-серверної архітектури вебзастосунку. Серверну частину розроблено мовою PHP із використанням системи керування базами даних MySQL, клієнтську засобами HTML, CSS та JavaScript.

Основним елементом інтерфейсу є вебсторінка `group_chat.php`, яка забезпечує:

- відображення історії повідомлень;
- форму введення тексту;
- елементи навігації між розділами системи.

Серверна логіка відповідає за обробку запитів користувачів, перевірку ролей доступу, створення чатів та взаємодію з базою даних.

Структура бази даних. Для реалізації групового чату структуру бази даних було розширено шляхом створення додаткових таблиць.

Таблиця `group_chats` містить інформацію про створені чати та включає такі поля:

- унікальний ідентифікатор чату; – ідентифікатор куратора;
- назву чату; – дату створення.

Таблиця `group_chat_messages` призначена для збереження повідомлень і містить:

- ідентифікатор повідомлення; – ідентифікатор чату;
- тип відправника; – ідентифікатор користувача;
- текст повідомлення; – дату й час надсилання.

Таблиця `group_chat_members` призначена для збереження складу учасників групових чатів і містить:

- ідентифікатор запису; – ідентифікатор чату;
- тип учасника; – ідентифікатор користувача;
- дату приєднання.

Між таблицями встановлено зв'язок типу «один до багатьох», де один чат може містити необмежену кількість повідомлень.

Алгоритм роботи модуля. Алгоритм функціонування групового чату включає такі етапи:

- авторизація користувача в системі;
- визначення ролі користувача (куратор або студент);
- отримання інформації про академічну групу;
- пошук відповідного чату в базі даних;
- автоматичне створення чату (за потреби);
- завантаження історії повідомлень;
- надсилання нового повідомлення;
- оновлення відображення чату.

Інтерфейс користувача. Інтерфейс групового чату інтегровано в загальний дизайн вебзастосунку. Сторінка містить блок вибору групи (для куратора), заголовок чату, область відображення повідомлень та форму введення тексту. Повідомлення відображаються у вигляді діалогових блоків із зазначенням автора та часу надсилання. Візуально розрізняються власні повідомлення користувача та повідомлення інших учасників чату.

Результати реалізації. У процесі тестування перевірено коректність створення чату, надсилення повідомлень різними ролями, відображення історії та збереження даних у базі. Результати підтвердили стабільну роботу модуля та відповідність функціональних можливостей поставленим вимогам.

3.4. Розгортання та запуск системи

Розроблений вебзастосунок управління комунікацією між куратором та студентами з модулем групового чату передбачає можливість розгортання як у локальному середовищі, так і на віддаленому вебсервері. Для тестування та демонстрації працездатності системи використано локальне серверне середовище XAMPP.

Підготовка серверного середовища. На першому етапі виконано встановлення програмного пакета XAMPP, який містить необхідні компоненти для функціонування вебзастосунку:

- вебсервер Apache; – систему керування базами даних MySQL;
- інтерпретатор PHP; – вебінтерфейс адміністрування phpMyAdmin.

Після встановлення середовища виконано запуск служб Apache та MySQL через панель керування XAMPP, що забезпечило готовність серверної інфраструктури до розгортання застосунку.

Розміщення файлів проекту. Файли вебзастосунку було розміщено в кореневому каталозі вебсерверу:

xampp/htdocs/university_messages/

У директорії проекту знаходяться:

- PHP-файли серверної логіки; – таблиці стилів CSS;
- HTML-шаблони інтерфейсу; – JavaScript-скрипти;
- конфігураційні файли.

Така структура забезпечує коректну маршрутизацію HTTP-запитів і взаємодію клієнтської та серверної частин системи.

Створення бази даних. Для забезпечення зберігання даних у середовищі phpMyAdmin створено окрему базу даних, наприклад:

university_messaging_system

Після створення бази виконано імпорт SQL-скрипту, який містить структуру таблиць:

- curators;
- students;
- messages;
- group_chats;
- system_settings.
- group_chat_members;
- group_chat_messages;
- login_attempts;
- system_logs;

Імпорт здійснювався через вкладку «Import» вебінтерфейсу phpMyAdmin шляхом завантаження SQL-файлу структури бази даних.

Налаштування конфігурації підключення. Параметри підключення до бази даних задаються у файлі config.php. У конфігураційному файлі визначено такі параметри:

- адреса сервера бази даних;
- ім'я користувача;
- пароль доступу;
- назва бази даних.

Приклад базових параметрів підключення:

localhost сервер БД

root користувач (без пароля для локального середовища)

university_messaging_system назва бази

Централізоване зберігання конфігурації забезпечує спрощене адміністрування та можливість швидкої зміни середовища розгортання.

Ініціалізація та запуск системи. Після завершення налаштувань система запускається через веббраузер за адресою: http://localhost/university_messages/

Головною точкою входу є файл index.php, який виконує перевірку автентифікації користувача та перенаправляє його до відповідного кабінету залежно від ролі.

Після авторизації користувач отримує доступ до функціоналу системи, зокрема:

- індивідуального обміну повідомленнями;
- перегляду історії листування;
- роботи з модулем групового чату;
- надсилання та отримання повідомлень у межах академічної групи.

Перевірка працездатності. Коректність розгортання системи перевіряється шляхом:

- реєстрації тестових користувачів;
- входу до системи під різними ролями;
- надсилання індивідуальних повідомлень;
- створення та відкриття групового чату;
- обміну повідомленнями між учасниками.

Результати перевірки підтвердили стабільну роботу вебзастосунку в локальному середовищі та готовність системи до подальшого розгортання на віддаленому сервері.

РОЗДІЛ 4. ТЕСТУВАННЯ ТА ОЦІНКА

4.1. Методи тестування

Тестування прототипу виконувалось з метою перевірки коректності ключових сценаріїв взаємодії «куратор ↔ студент» та функціонування групового чату.

Функціональне тестування охоплювало: реєстрацію та авторизацію користувачів, перевірку ролей і обмеження доступу, створення та перегляд профілів, надсилання й отримання індивідуальних повідомлень, а також роботу вкладених відповідей.

Для групового чату перевірено відкриття чату за групою, його створення куратором під час першого відкриття, надсилання повідомлень куратором і студентами, відображення історії повідомлень у хронологічному порядку та коректне визначення автора і часу надсилання.

Безпекове тестування включало валідацію введених даних, використання підготовлених SQL-запитів і обмеження доступу до сторінок на основі сесійної авторизації.

Тестування проводилось у локальному середовищі XAMPP із використанням браузера Google Chrome та інструмента phpMyAdmin для контролю стану даних у базі.

4.2. Результати тестування

Після запуску системи відображається головна сторінка вебзастосунку (рис. 4.1).

При запуску системи користувач бачить головну сторінку, яка виконує функцію точки входу до системи. У верхній частині сторінки розміщено заголовок «Система повідомлень університету» та підзаголовок «Ефективна

комунікація між кураторами та студентами», що відображають призначення розробленого програмного рішення.

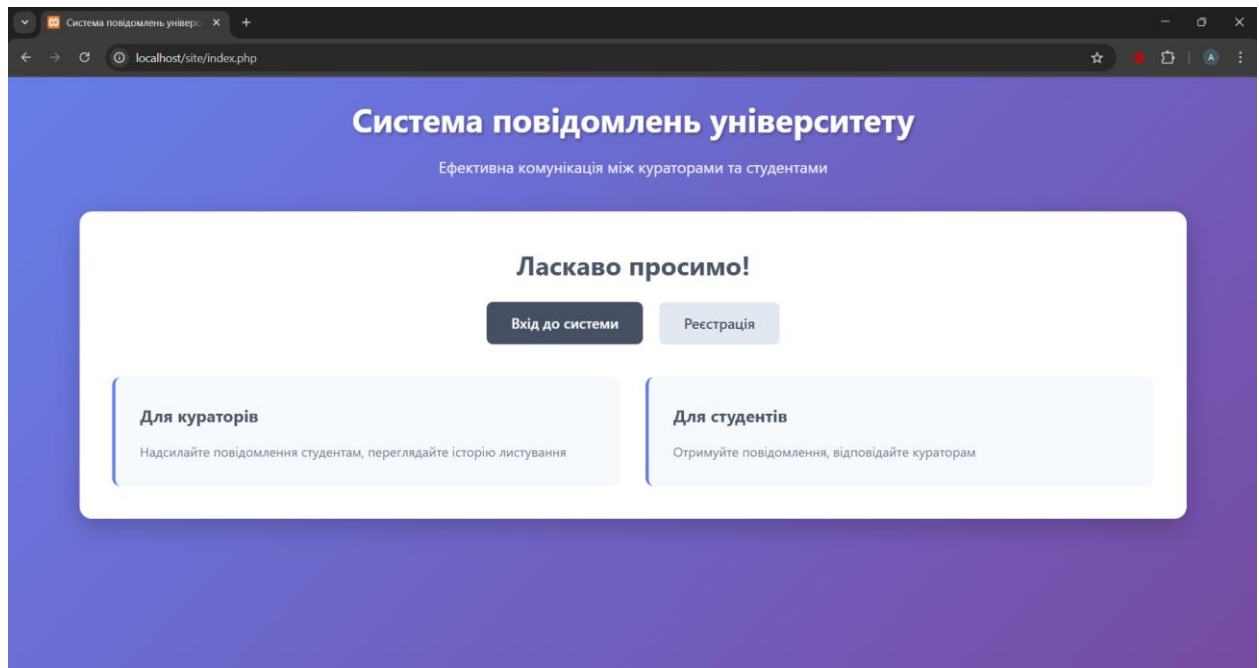


Рисунок 4.1. Головна сторінка системи

У центральній частині розташована панель привітання з великим текстом "Ласкаво просимо!" та двома основними кнопками:

- кнопка "Вхід до системи" – для авторизованих користувачів;
- кнопка "Реєстрація" – для нових користувачів.

Нижче розміщено інформаційні блоки, які описують переваги системи для різних груп користувачів:

- для кураторів – можливість надсилати повідомлення студентам та переглядати історію листування;
- для студентів – отримання повідомлень та можливість відповідати кураторам.

Дизайн виконано у сучасному стилі з використанням градієнтного фону, заокруглених кутів та тіней, що створює приємний візуальний ефект.

Зареєструємося (рис. 4.2). Оберемо куратора.

При натисканні кнопки "Реєстрація" система перенаправляє користувача на сторінку реєстрації. Ця сторінка містить форму з обов'язковими полями для заповнення:

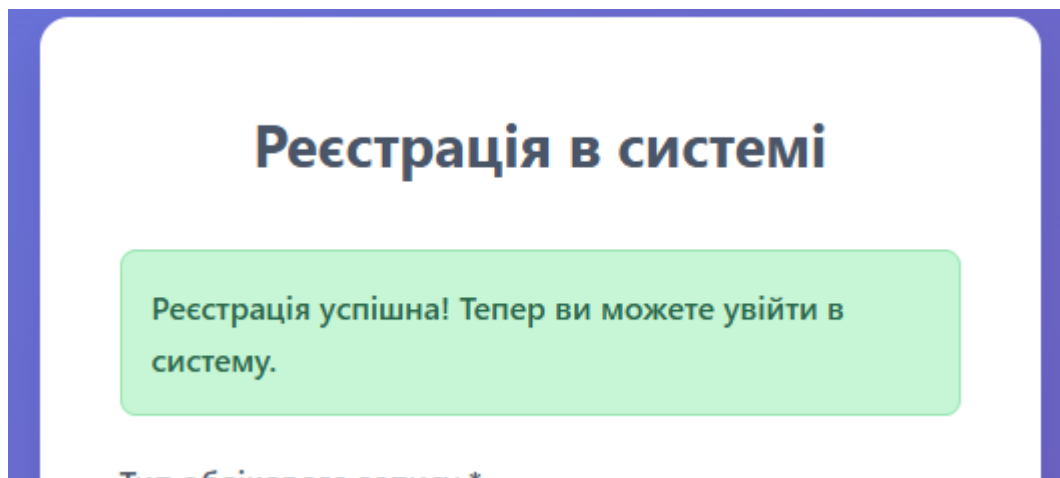


Рисунок 4.2. Вікно реєстрації куратора

Основні поля форми:

- тип облікового запису – випадаючий список з варіантами "куратор" або "студент";
- повне ім'я – текстове поле для введення ПІБ;
- ім'я користувача – унікальний логін для входу;
- email – електронна пошта для зв'язку;
- пароль – поле з прихованим вводом;
- підтвердження пароля – повторний введення пароля для перевірки.

Процес реєстрації куратора:

1. користувач обирає "куратор" у полі "тип облікового запису";
2. заповнює всі обов'язкові поля;
3. система автоматично приховує поля, специфічні для студентів (назва групи, вибір куратора);
4. при натисканні кнопки "zareestruватися" система перевіряє:
 - унікальність імені користувача та email;
 - відповідність паролів вимогам безпеки;
 - коректність формату email.

5. після успішної реєстрації система автоматично авторизує користувача та перенаправляє в кабінет куратора.

Виконаємо авторизацію користувача з роллю куратора (рис. 4.3).

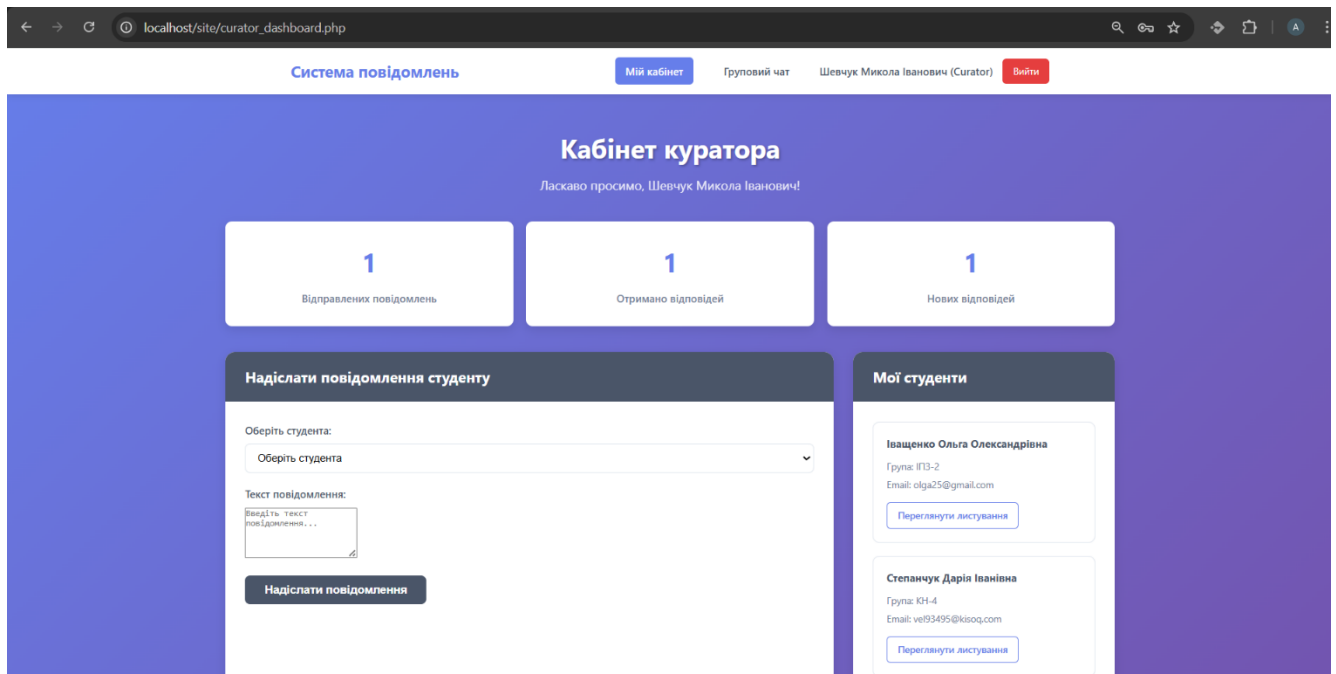


Рисунок 4.3. Авторизація куратора

Сторінка входу містить спрощену форму авторизації з трьома полями:

Поля форми входу:

- тип облікового запису – вибір між "куратор" та "студент";
- ім'я користувача – введення зареєстрованого логіну;
- пароль – введення пароля.

Процес авторизації:

1. користувач обирає "куратор" як тип облікового запису;
2. вводить свої облікові дані;
3. система виконує перевірку:
 - існування користувача в таблиці кураторів;
 - відповідність пароля;
 - активність облікового запису;
 - відсутність блокування через занадто багато невдалих спроб входу.

4. при успішній авторизації система:

- створює сесію користувача;
- оновлює час останнього входу;
- перенаправляє в кабінет куратора.

Процес реєстрації студента має додаткові поля порівняно з куратором:

Додаткові поля для студента:

- група – назва навчальної групи (наприклад, "КН-4");
- куратор – випадючий список з доступними кураторами.

Особливості реєстрації студента:

1. при виборі "студент" у полі типу облікового запису система автоматично відображає додаткові поля;
2. поле "група" стає обов'язковим для заповнення;
3. поле "куратор" вимагає вибору одного з зареєстрованих кураторів;
4. система перевіряє, щоб усі поля були заповнені коректно;
5. після реєстрації студент автоматично прив'язується до обраного куратора.

Зареєструємо обліковий запис студента (рис. 4.4).

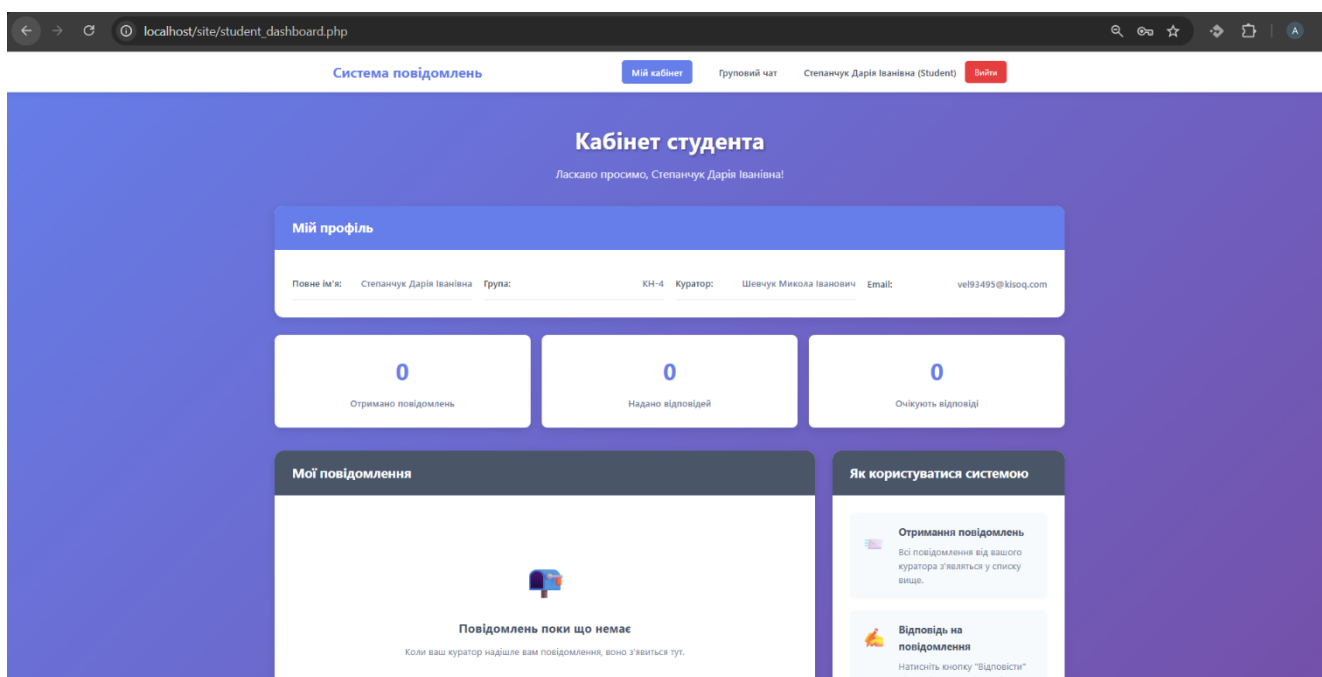


Рисунок 4.4. Авторизація студента

Як куратор напишемо студенту повідомлення (рис. 4.5).

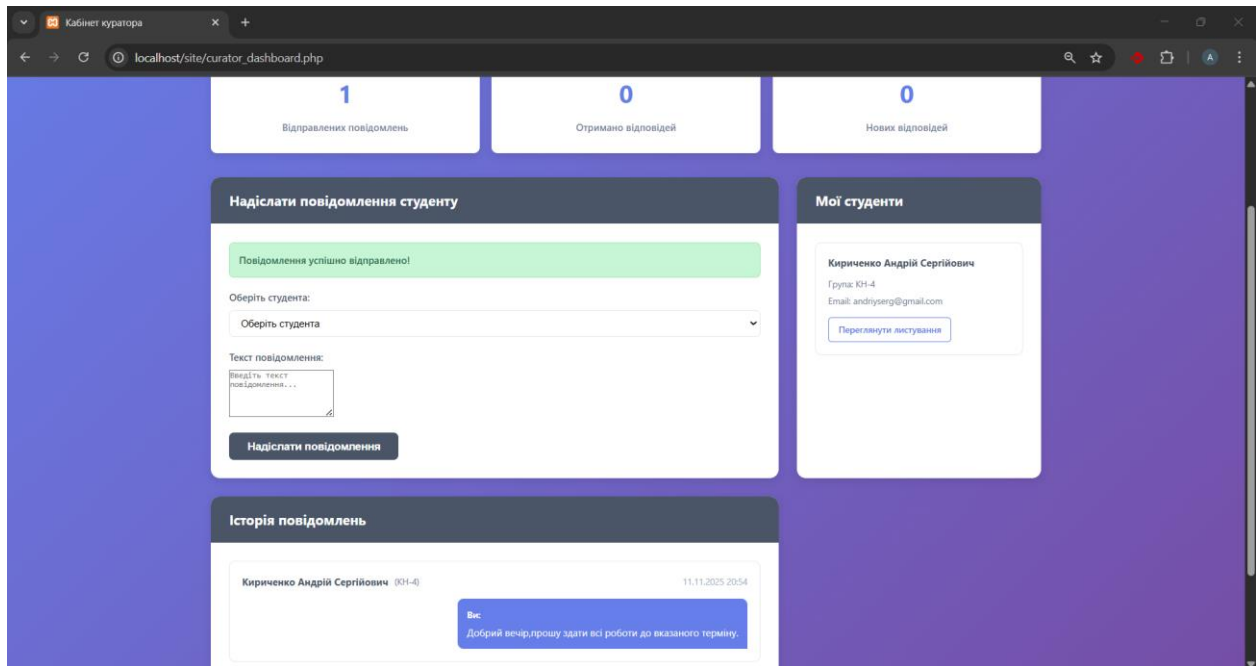


Рисунок 4.5. Повідомлення студенту

Кабінет куратора має декілька основних секцій:

Панель статистики:

- відображає загальну кількість відправлених повідомлень;
- показує кількість отриманих відповідей;
- відображає кількість нових (непрочитаних) відповідей.

Форма відправки повідомлення:

- Випадаючий список студентів – перелік усіх студентів, закріплених за куратором
- Текстове поле повідомлення – область для введення тексту
- Кнопка "Надіслати повідомлення" – для відправки

Процес відправки повідомлення:

1. куратор обирає студента з випадаючого списку;
2. вводить текст повідомлення у текстове поле;
3. натискає кнопку відправки;
4. система виконує перевірки:
 - вибір студента (поле не може бути порожнім);
 - наявність тексту повідомлення;

- довжина тексту (не перевищує максимально дозволєну).

5. після успішної відправки:

- повідомлення зберігається в базі даних;
- система відображає повідомлення про успіх;
- оновлюється статистика на панелі.

Перевіримо функціонування системи з облікового запису студента (рис. 4.6).

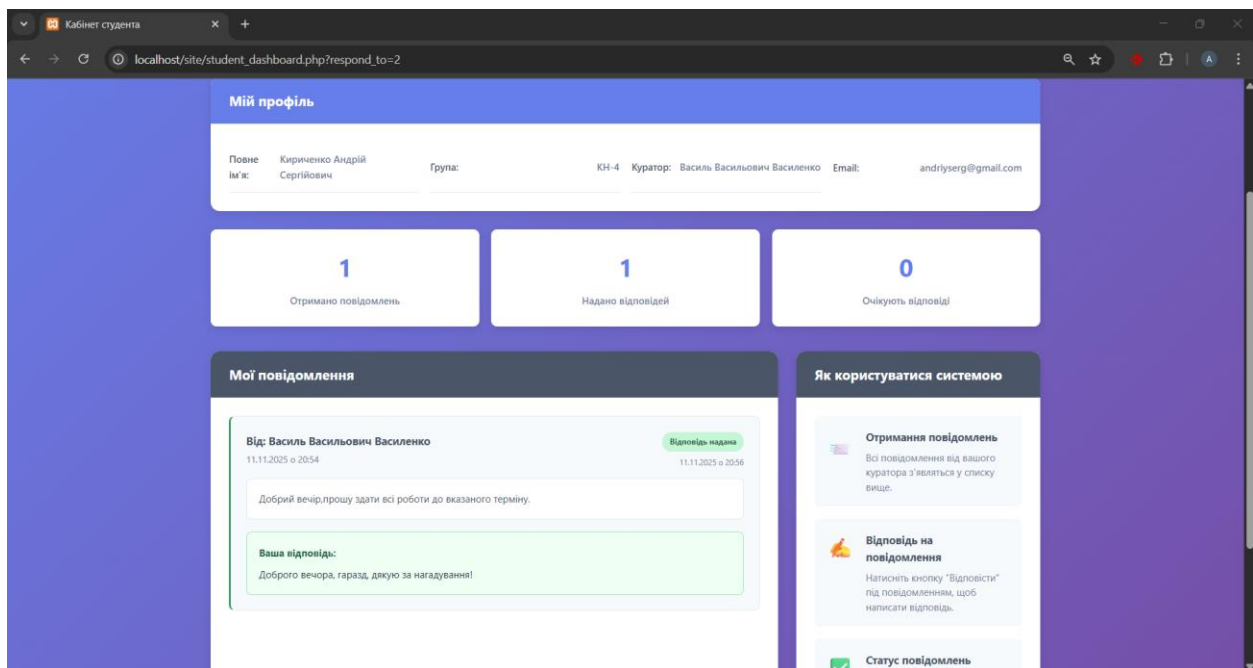


Рисунок 4.6. Повідомлення від викладача

Кабінет студента організований наступним чином:

Інформація про профіль:

- повне ім'я студента;
- ім'я куратора;
- назва групи;
- електронна пошта.

Панель статистики студента:

- кількість отриманих повідомлень;
- кількість наданих відповідей;
- кількість повідомлень, що очікують відповіді.

Список повідомлень:

- відображається у хронологічному порядку;
- кожне повідомлення представлено у вигляді картки;
- індикатори статусу:
 - "Очікує відповіді" – червоний значок для нових повідомлень;
 - "Відповідь надана" – зелений значок для відповідей;

Перегляд повідомлення:

1. студент бачить повідомлення від куратора з датою відправки;
2. текст повідомлення чітко відформатований та легко читається;
3. якщо студент ще не відповів – відображається кнопка "відповісти";
4. при натисканні кнопки "відповісти" з'являється текстова область для введення відповіді.

5. після введення тексту та натискання кнопки "надіслати відповідь":
 - відповідь зберігається в базі даних;
 - статус повідомлення змінюється на "відповідь надана";
 - куратор отримує сповіщення про нову відповідь.

Система сповіщень:

- При отриманні нового повідомлення студент бачить візуальний індикатор

- Непрочитані повідомлення можуть виділятися кольором або іконкою
- Система забезпечує своєчасне сповіщення про нові повідомлення.

Куратор обирає академічну групу для відкриття групового чату (рис. 4.7).

Після вибору групи система відкриває відповідний чат, у якому відображається історія повідомлень. У правій частині інтерфейсу показано діалогові блоки з іменем відправника, датою та часом надсилання, а також текстом повідомлення. Нові повідомлення можуть бути додані через форму введення, розміщену під історією листування.

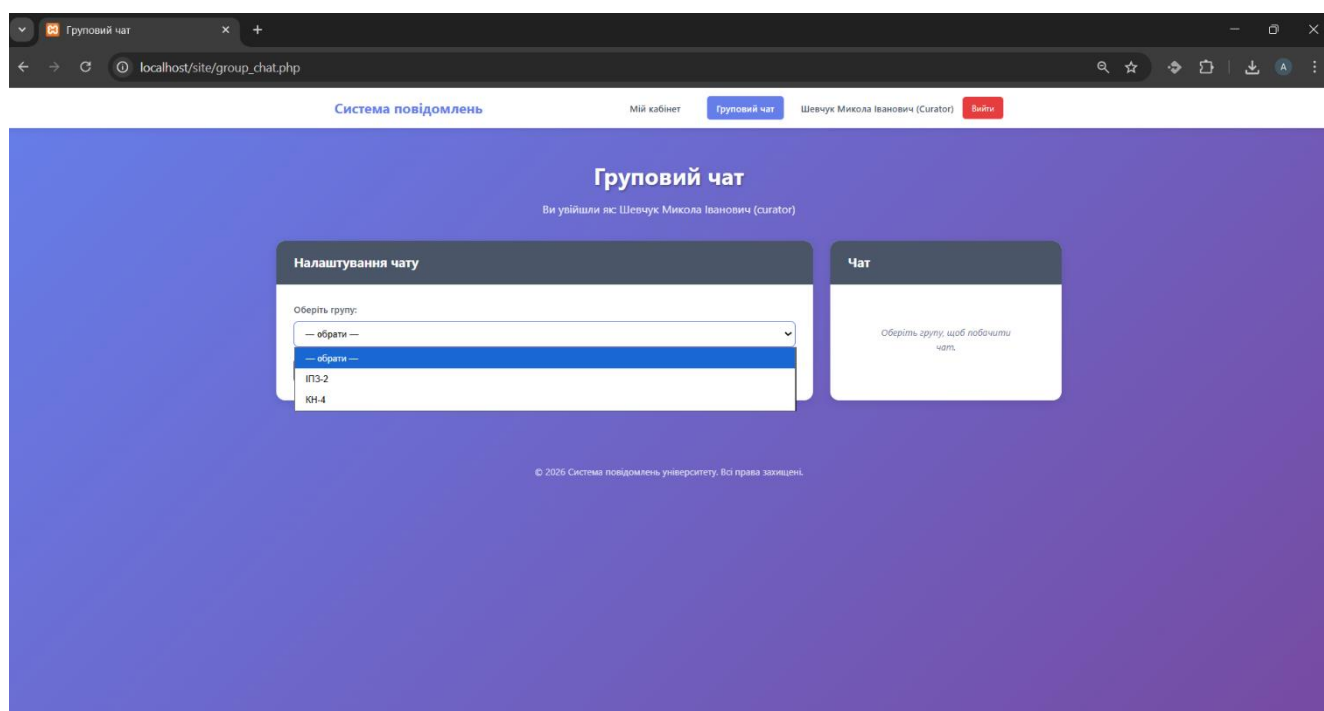


Рисунок 4.7. Вибір академічної групи куратором

У груповому чаті відображається історія повідомлень із зазначенням автора та часу надсилання (рис. 4.8).

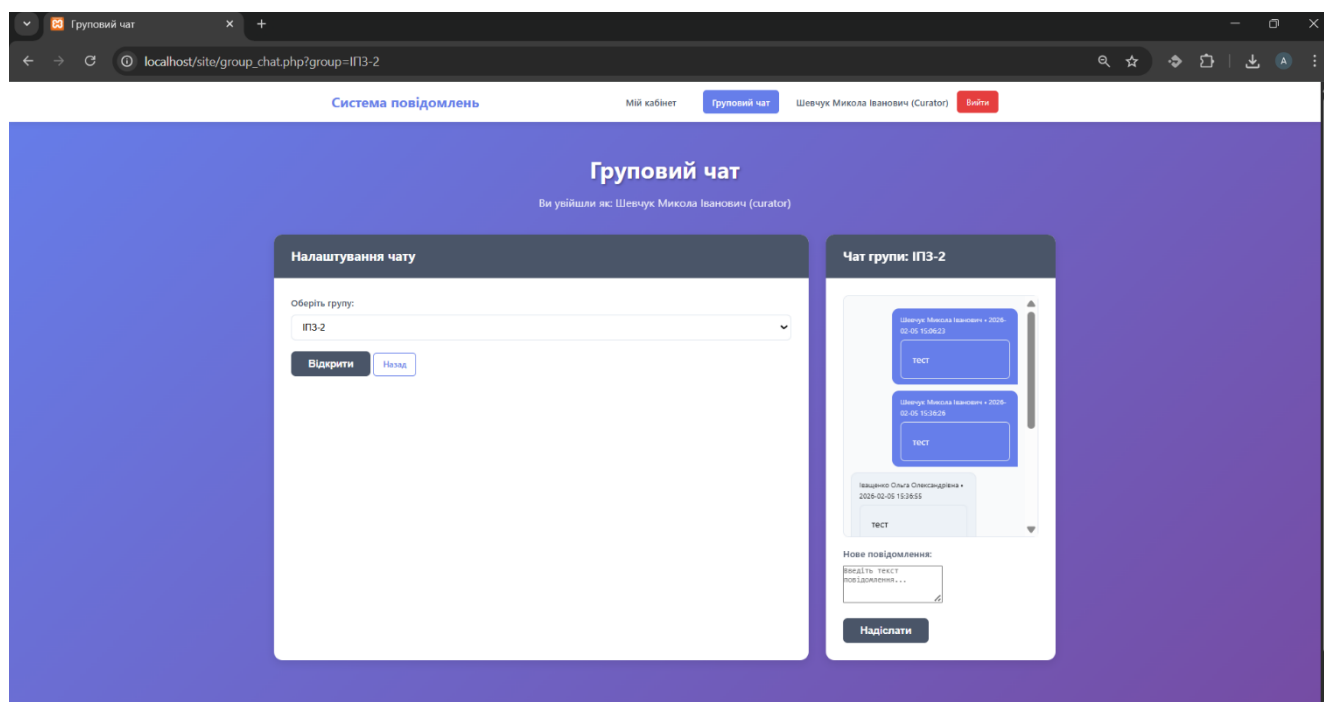


Рисунок 4.8. Інтерфейс групового чату та відображення повідомлень

На рисунку представлено інтерфейс групового чату після відкриття академічної групи куратором. У правій частині сторінки відображається історія повідомлень у хронологічному порядку. Повідомлення містять інформацію про автора, дату й час надсилання. Власні повідомлення користувача візуально відокремлюються кольором, що спрощує сприйняття діалогу. У нижній частині розміщено форму введення тексту та кнопку надсилання повідомлення.

Студент має доступ до групового чату своєї академічної групи, де відображається історія повідомлень та форма для надсилання нових (рис. 4.9).

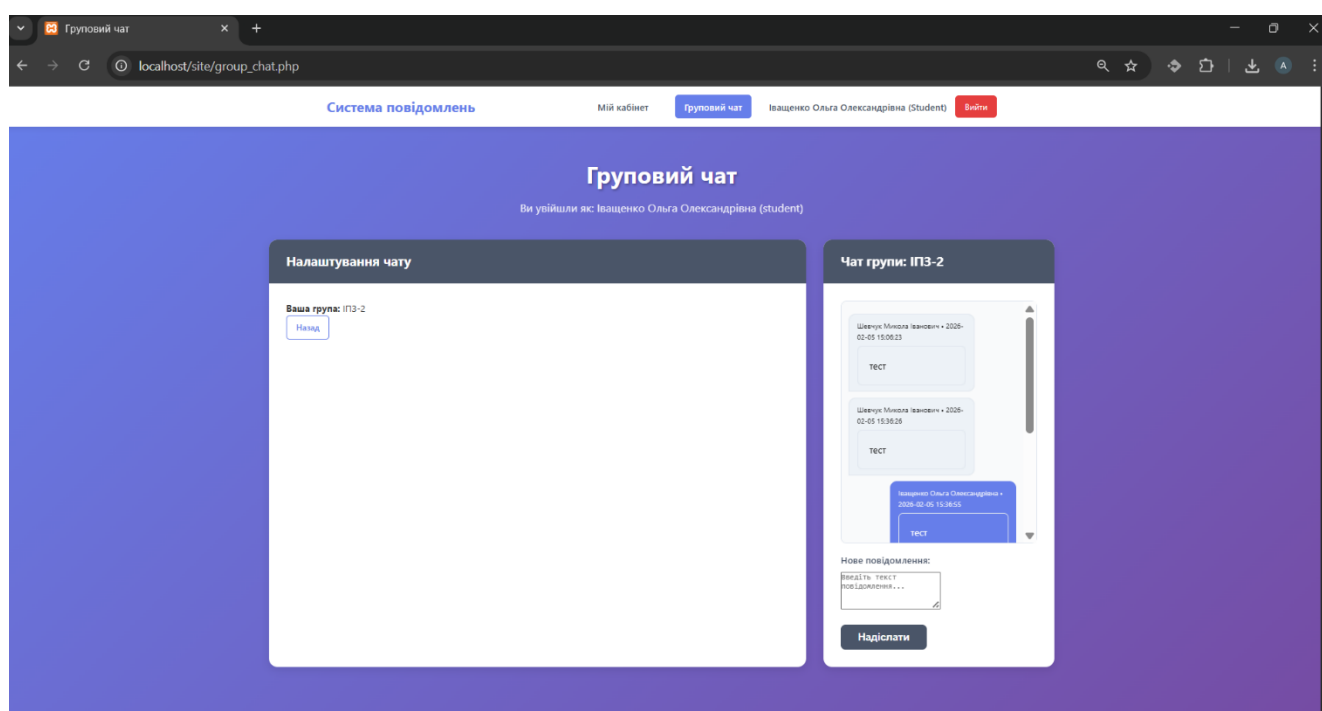


Рисунок 4.9. Інтерфейс групового чату для студента

На екрані відображається чат академічної групи студента із зазначенням автора та часу надсилання повідомлень. Повідомлення користувача візуально відрізняються від повідомлень інших учасників, що полегшує сприйняття діалогу. У нижній частині інтерфейсу розташовано текстове поле для введення нового повідомлення та кнопку його надсилання.

4.3. Тестування безпеки системи

Окрему увагу під час тестування приділено перевірці механізмів захисту вебзастосунку. Зокрема, було протестовано:

- захист від SQL-ін'єкцій шляхом введення спеціальних символів у поля авторизації та реєстрації;
- захист від XSS-атак через спроби вставки HTML- і JavaScript-коду в текст повідомлень;
- перевірку CSRF-захисту через підробку запитів без валідного токена;
- обмеження кількості спроб входу для запобігання brute-force атакам.

Результати тестування показали, що система коректно обробляє небезпечні запити, відхиляє невалідні дані та забезпечує належний рівень захисту облікових записів і повідомлень користувачів.

4.4. Оцінка продуктивності системи

Оцінка продуктивності проводилась у локальному середовищі шляхом імітації одночасної роботи кількох користувачів (куратора та студентів).

Перевірялась швидкість:

- авторизації користувачів;
- відкриття кабінетів;
- завантаження історії повідомлень;
- відображення групового чату.

Середній час завантаження сторінок не перевищував 1–2 секунд, що відповідає встановленим технічним вимогам. Затримок при надсиланні повідомлень або оновленні історії чату не зафіксовано.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було успішно розроблено та реалізовано веборієнтовану систему управління повідомленнями між кураторами та студентами, що повністю відповідає поставленій меті, завданням та вихідним вимогам дослідження. Розроблений програмний продукт реалізовано у вигляді вебзастосунку, доступного через браузер без необхідності встановлення додаткового програмного забезпечення.

Науково-технічні результати роботи включають проектування та реалізацію архітектури системи, що базується на багаторівневому підході з чітким розділенням обов'язків між компонентами. Було сформовано модель даних із використанням реляційної бази даних MySQL, що забезпечує ефективне зберігання, обробку та структурування інформації про користувачів, індивідуальні повідомлення та групові комунікації. Особливу увагу приділено розробці інтерфейсу користувача, який поєднує функціональність, логічну навігацію та зручність використання для обох категорій користувачів кураторів і студентів.

Головні досягнення роботи полягають у створенні цілісного програмного рішення, що включає:

- систему авторизації та розмежування прав доступу;
- функціонал індивідуального обміну повідомленнями з підтримкою відповідей;
- модуль групового чату для взаємодії куратора зі студентами академічної групи;
- інтуїтивний користувацький інтерфейс з адаптивним дизайном;
- комплексний захист від основних вебзагроз;
- систему моніторингу та логування подій.

Реалізований модуль групового чату розширює можливості системи та забезпечує оперативну колективну взаємодію в межах академічної групи. Його функціонал дозволяє куратору здійснювати швидке інформування студентів,

організовувати обговорення організаційних питань і централізовано зберігати історію комунікації. Студенти, у свою чергу, отримують можливість переглядати повідомлення, брати участь у спільному обговоренні та оперативно реагувати на інформацію куратора.

Технічні переваги розробленої системи проявляються у використанні сучасних технологій веброзробки. Система реалізована із використанням мови програмування PHP, СУБД MySQL та клієнтських технологій HTML, CSS і JavaScript, що забезпечило стабільність роботи, безпеку даних та масштабованість рішення. Застосування підготовлених SQL-запитів, механізмів валідації введених даних, захисту форм та хешування паролів дозволило створити надійну систему, захищену від поширених типів вебатак.

Практична цінність роботи полягає у можливості використання розробленої системи в освітньому процесі закладів вищої освіти для організації ефективної комунікації між кураторами та студентами. Система дозволяє впорядкувати процес обміну повідомленнями, забезпечує швидкий доступ до історії листування, а також підтримує як індивідуальну, так і групову взаємодію учасників освітнього середовища.

Використання розробленої системи дозволяє зменшити часові витрати на інформування студентів, централізувати комунікацію та підвищити оперативність управлінських рішень у межах академічних груп, що сприяє оптимізації організаційних процесів освітнього закладу.

Проведене тестування підтвердило коректність функціонування всіх модулів системи, включаючи індивідуальні повідомлення та груповий чат. Система демонструє стабільну роботу, швидкий час відгуку та правильність обробки користувацьких запитів у різних сценаріях використання.

Перспективи подальшого розвитку системи полягають у розширенні її функціональних можливостей, зокрема шляхом впровадження push- та email-сповіщень, додавання обміну файлами в чатах, інтеграції з існуючими системами управління навчанням, а також розробки мобільного застосунку для забезпечення доступу з портативних пристроїв.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Best Learning Management System (LMS) Solution for Schools, Colleges, and Universities. URL: <https://www.instructure.com/lms-learning-management-system> (дата звернення: 10.05.2026).
2. Tutor Platform. URL: <https://www.tutor-platform.com/> (дата звернення: 10.05.2026).
3. 30 Best Corporate Learning Management Systems of 2025. URL: <https://peoplemanagingpeople.com/tools/best-corporate-learning-management-systems/> (дата звернення: 10.05.2026).
4. Система електронної взаємодії державних електронних інформаційних ресурсів «Трембіта». URL: <https://trembita.gov.ua/> (дата звернення: 10.05.2026).
5. Kotenok A., Kulaha I., Klivak V., Tkachenko O. The E-Government's Influence on the Country's Economy (at the Example of Ukraine and Estonia). Proceedings of the III International Scientific Congress «Society of Ambient Intelligence 2020». Atlantis Press, 2020.P. 175–182. DOI: <https://doi.org/10.2991/aebmr.k.200318.022> (дата звернення: 10.05.2026).
6. HTML: HyperText Markup Language. URL: <https://developer.mozilla.org/docs/Web/HTML> (дата звернення: 10.05.2026).
7. PHP Manual. URL: <https://www.php.net/> (дата звернення: 10.05.2026).
8. CSS Cascading Style Sheets. URL: <https://www.w3.org/Style/CSS/Overview.en.html> (дата звернення: 10.05.2026).
9. MySQL The World's Most Popular Open Source Database. URL: <https://www.mysql.com/> (дата звернення: 10.05.2026).
10. Open Server Panel: Local Web Server for Windows. URL: <https://ospanel.io/> (дата звернення: 10.05.2026).
11. JavaScript Guide. URL: <https://developer.mozilla.org/docs/Web/JavaScript> (дата звернення: 10.05.2026).

12. Рівненський державний гуманітарний університет : методичні рекомендації до виконання кваліфікаційних робіт бакалавра (спеціальність 122 «Комп'ютерні науки»).Рівне:РДГУ,2023.URL:
<http://repository.rshu.edu.ua/id/eprint/15039/> (дата звернення: 10.05.2026).
13. Закон України «Про вищу освіту» від 01.07.2014 № 1556-VII. Редакція від 01.01.2024. URL: <https://zakon.rada.gov.ua/laws/show/1556-18>. (дата звернення: 10.05.2026).
14. Гуржій А. М., Овчарук О. В. Інформаційно-комунікаційні технології в освіті : навч. посіб. Київ : Центр навчальної літератури, 2020. 204 с. URL: <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://kubg.edu.ua/images/stories/Departaments/nmc.nd/aspirantura/entrance/13-00-10.pdf> (дата звернення: 10.05.2026).
15. Глоба Л. С. Розробка інформаційних ресурсів та систем : підручник. Київ : КПІ ім. Ігоря Сікорського,2021.320с. URL:
<https://duikt.edu.ua/ua/lib/1/category/96/view/1690> (дата звернення: 10.05.2026).
16. ДСТУ ISO/IEC 25010:2016. Системна та програмна інженерія. Вимоги до якості систем і програмного забезпечення та їх оцінювання. Київ:ДП «УкрНДНЦ»,2017.
URL:https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=69134
(дата звернення: 10.05.2026).
17. Лавріщева К. М. Програмна інженерія : підручник. Київ : Академперіодика, 2020.280с.URL:<https://books.chic.cv.ua/k-m-lavrishhevaprogramna-inzheneriyapidruchnyk/> (дата звернення: 10.05.2026).
18. Березко О. Л., Пастернак Я. В. Вебтехнології : навчальний посібник. Львів : Видавництво Львівської політехніки, 2023. 244 с. URL:
<https://vlp.com.ua/taxonomy/term/3560> (дата звернення: 10.05.2026).
19. Стратегія цифровізації освіти і науки України на період до 2025 року. Розпорядження КМУ від 14 квітня 2023 р. № 329-р. URL:<chrome->

[extension://efaidnbmnnnibpcajpcgklclefindmkaj/https://auc.org.ua/sites/default/files/projekt_strategiya.pdf](chromeextension://efaidnbmnnnibpcajpcgklclefindmkaj/https://auc.org.ua/sites/default/files/projekt_strategiya.pdf) (дата звернення: 10.05.2026).

20. Батрак Є. О. Архітектура комп'ютерних систем : навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2020.195с.

URL:<chromeextension://efaidnbmnnnibpcajpcgklclefindmkaj/https://ela.kpi.ua/server/api/core/bitstreams/180a1bfe-cfc6-488e-a874-53cb8c00d267/content>(дата звернення: 10.05.2026).