НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

«До захисту в ЕК»	**«До захисту допущено»**
Директор інституту(декан факультету)	Завідувач кафедри
_____ ___Андрій ФОРСЮК__	_____ _Сергій ГРИБКОВ__
(підпис) (ім'я та прізвище)	(підпис) (ім'я та прізвище)
«__2_» __червня_____ 2025р.	«_2__» ____червня_____ 2025р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

зі спеціальності 122 «Комп'ютерні науки»
(код та назва спеціальності)
освітньо-професійної програми _Інформаційні системи та штучний інтелект___

на тему: Розроблення веб-застосунку для студентської спільноти Київського індустріального фахового коледжу КНУБА

Виконав: здобувач _4_ курсу, групи __КН-4-4___

_____Смілянський Михайло Юрійович_____ _____
(прізвище, ім'я по батькові повністю) (підпис)

Керівник __Ліманська Наталія Володимирівна_____ _____
(прізвище , ім'я та по батькові повністю) (підпис)

Консультанти_____ _____
(ім'я та прізвище) (підпис)

_____ _____
(ім'я та прізвище) (підпис)

_____ _____
(ім'я та прізвище) (підпис)

Рецензент _____ _____
(ім'я та прізвище) (підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволеної допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач_____
(підпис)

Київ – 2025 р

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки
Освітній ступінь бакалавр
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма Інформаційні системи та штучний інтелект

ЗАТВЕРДЖУЮ

Завідувач кафедри
Інформаційних технологій, штучного інтелекту і кібербезпеки
Сергій ГРИБКОВ

" 28 " квітня 2025 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА
Смілянського Михайла Юрійовича
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення веб-застосунку для студентської спільноти Київського індустріального фахового коледжу КНУБА

керівник роботи Ліманська Наталія Володимирівна, старший викладач
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 28 квітня 2025 року № 254-кс
2. Строк подання здобувачем роботи 30.05.2025 р.
3. Вихідні дані до роботи
Загальна інформація про коледж КІФК КНУБА, організаційна структура

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
1. Дослідження предметної області
2. Технічне завдання та проектування
3. Проектування, створення та апробація програмного забезпечення
5. Перелік графічного матеріалу
1. Схеми бази даних
2. Діаграма Ганта
3. Скріншоти інтерфейсу користувача
4.Фрагменти коду інформаційної системи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Ліманська Наталія Володимирівна	28.04.2025	13.05.2025
2	Ліманська Наталія Володимирівна	28.04.2025	19.05.2025
3	Ліманська Наталія Володимирівна	28.04.2025	26.05.2025

7. Дата видачі завдання_28.04.2025_____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Постановка задачі. Визначення цілей, завдань та вимог до системи	28.04.2025 – 30.04.2025	Виконано
2	Аналіз предметної області. Огляд систем-аналогів	01.05.2025– 04.05.2025	Виконано
3	Реалізація основних модулів системи та проєктування бази даних	05.05.2025– 18.05.2025	Виконано
4	Підготовка пояснювальної записки, презентації	19.05.2025– 23.05.2025	Виконано
5	Оформлення відгуку, рецензії, подання роботи	24.05.2025– 26.05.2025	Виконано

Здобувач _____ _____
(підпис) (прізвище та ініціали)

Керівник роботи _____ _____
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота: «Розроблення веб-застосунку для студентської спільноти Київського індустріального фахового коледжу КНУБА», 74 с., 3 таблиць, 24 ілюстрацій, 30 використаних джерел, 12 додатків.

У кваліфікаційній роботі здійснено проектування та розробку веб-застосунку для студентської спільноти Київського індустріального фахового коледжу КНУБА. Робота складається з трьох розділів, у яких послідовно розглянуто теоретичні основи розробки веб-застосунків для освітніх закладів, формування технічного завдання та проектування системи, а також практичну реалізацію і тестування розробленого програмного продукту.

У першому розділі проаналізовано сучасні підходи до створення веб-застосунків для студентських спільнот, досліджено існуючі рішення та визначено основні функціональні можливості, необхідні для успішної комунікації та взаємодії між студентами.

Другий розділ присвячено аналізу вимог та розробці технічного завдання для веб-застосунку. Детально описано функціональні та нефункціональні вимоги, спроектовано структуру системи, інформаційну модель, архітектуру веб-застосунку та інтерфейси користувачів.

У третьому розділі представлено практичну реалізацію проекту з використанням сучасних веб-технологій, таких як Python, Flask та SQLite. Описано процес розгортання системи, проведення тестування.

Розроблений веб-застосунок надає студентам можливість створювати профілі, залишати коментарі, переглядати оголошення, навчальні матеріали та розклад занять. Користувач з правами адміністратора забезпечує управління контентом, модерацію модулів.

Ключові слова: ВЕБ-ЗАСТОСУНОК, СТУДЕНТСЬКА СПІЛЬНОТА, ОСВІТНІ ТЕХНОЛОГІЇ, ІНФОРМАЦІЙНІ СИСТЕМИ, SQLITE, PYTHON, ТЕХНОЛОГІЧНИЙ СТЕК.

ABSTRACT

Qualification work: “Development of a web application for the student community of the Kyiv Industrial Vocational College of KNUBA”, 74 p., 3 tables, 24 illustrations, 30 references, 12 appendices.

In the qualification work, the design and development of a web application for the student community of the Kyiv Industrial Vocational College of KNUBA was carried out. The work consists of three chapters, which consistently consider the theoretical foundations of developing web applications for educational institutions, the formation of technical specifications and system design, as well as the practical implementation and testing of the developed software product.

The first section analyzes modern approaches to creating web applications for student communities, examines existing solutions, and identifies the main functionalities necessary for successful communication and interaction between students.

The second section is devoted to the analysis of requirements and the development of technical specifications for the web application. The functional and non-functional requirements are described in detail, and the system structure, information model, web application architecture, and user interfaces are designed.

The third section presents the practical implementation of the project using modern web technologies such as Python, Flask, and SQLite. The process of system deployment and testing is described.

The developed web application provides students with the ability to create profiles, leave comments, view announcements, educational materials, and class schedules. A user with administrator rights provides content management and module moderation.

Keywords: WEB APPLICATION, STUDENT COMMUNITY, EDUCATIONAL TECHNOLOGIES, INFORMATION SYSTEMS, SQLITE, PYTHON, TECHNOLOGY STACK.

.

ЗМІСТ

ВСТУП..... 7
Розділ 1 Дослідження предметної області та постановка задачі..... 9
1.1. Загальна характеристика студентської спільноти КІФК КНУБА 9
1.2. Організаційна структура коледжу та роль студентської спільноти 10
1.3. Аналіз нинішнього стану комп'ютеризації та комунікації в коледжі 11
1.4. Функціональне моделювання та аналіз існуючих бізнес-процесів комунікації 13
1.5. Огляд існуючих рішень для розв'язання виявлених проблем..... 18
1.6. Техніко-економічне обґрунтування впровадження веб-сайту..... 22
1.7. Обґрунтування доцільності проєктування і розроблення веб-сайту..... 25
Розділ 2 Технічне завдання на проєктування 28
2.1. Загальні положення 28
2.2. Призначення і цілі створення системи 29
2.3. Характеристика об'єкта автоматизації 30
2.4. Вимоги до системи 31
2.5. Діаграма Ганта та календарний план розроблення проєкту 34
Розділ 3 Проєктування, створення та апробація програмного забезпечення . 37
3.1. Опис та обґрунтування вибору програмно-технічних засобів розроблення..... 37
3.2. Проєктування та створення бази даних..... 38
3.3. Реалізація функцій системи 45
3.4. Інструкція користувача 59
3.5. Тестування програмного продукту. 65
ВИСНОВКИ..... 69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 71
ДОДАТКИ..... 75

ВСТУП

У сучасному освітньому середовищі ефективна комунікація та обмін інформацією між усіма учасниками освітнього процесу є одним із ключових факторів успішного функціонування навчального закладу. Особливо гостро ця потреба проявляється в студентських спільнотах, де активне студентське життя, різноманітні заходи, культурно-масові події та щоденна взаємодія потребують налагодженої системи інформаційного обміну.

Київський індустріальний фаховий коледж КНУБА (КІФК КНУБА) є сучасним навчальним закладом з багатою історією та розвиненою студентською спільнотою. Проте, як показав проведений аналіз, існуюча система комунікації в коледжі характеризується розрізненістю інформаційних каналів, відсутністю централізованої платформи для обміну інформацією та недостатньою оперативністю поширення важливих повідомлень. Це призводить до комунікаційних розривів, втрати актуальної інформації, низького рівня залученості студентів до позанавчальних активностей та інших проблем, що негативно впливають на якість освітнього середовища в цілому.

Розробка веб-сайту для студентської спільноти КІФК КНУБА спрямована на вирішення цих проблем шляхом створення єдиної інформаційної платформи, яка забезпечить централізований, оперативний та зручний обмін інформацією між студентами, викладачами та адміністрацією коледжу. Цей проєкт має стратегічне значення для підвищення якості комунікації, покращення інформаційної інфраструктури та формування активної студентської спільноти в коледжі.

Актуальність даної роботи зумовлена зростаючою роллю інформаційних технологій в освітньому процесі, підвищенням вимог до швидкості та якості комунікації в сучасному інформаційному суспільстві, а також необхідністю створення комфортного інформаційного середовища для забезпечення повноцінного розвитку студентів. Розробка спеціалізованого веб-сайту дозволить не лише вирішити існуючі комунікаційні проблеми, але й створити

основу для подальшого розвитку інформаційної інфраструктури коледжу, підвищення рівня цифрової грамотності студентів та впровадження сучасних освітніх технологій.

Метою даної кваліфікаційної роботи є проєктування та розробка веб-сайту для студентської спільноти Київського індустріального фахового коледжу КНУБА, який забезпечить ефективну комунікацію між учасниками освітнього процесу, централізоване розміщення та управління різними типами інформації, а також створить умови для формування активної та згуртованої студентської спільноти.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Провести аналіз існуючого стану інформаційного обміну в коледжі та виявити основні проблеми комунікації.
2. Дослідити існуючі аналогічні системи та визначити їх переваги та недоліки.
3. Розробити технічне завдання та спроєктувати архітектуру системи.
4. Обрати оптимальні технології та інструменти для реалізації проєкту.
5. Розробити базу даних та програмні компоненти системи.
6. Створити зручний та інтуїтивно зрозумілий інтерфейс користувача.
7. Провести тестування системи та виправити виявлені помилки.
8. Підготувати документацію та інструкції для користувачів та адміністраторів системи.

Результатом роботи стане повноцінний веб-сайт для студентської спільноти КІФК КНУБА, який включатиме модулі оголошень, розкладу заходів, коментування, анонімних повідомлень та скарг, загублених/знайдених речей, а також систему аутентифікації та авторизації з різними рівнями доступу. Розроблена система матиме зручний інтерфейс, адаптований для різних пристроїв, та забезпечить ефективне вирішення виявлених комунікаційних проблем в коледжі.

РОЗДІЛ 1
ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1.Загальна характеристика студентської спільноти КІФК КНУБА

Київський індустріальний фаховий коледж КНУБА (КІФК КНУБА) [2] є сучасним навчальним закладом, який поєднує традиції якісної освіти та інноваційні підходи до навчального процесу. Коледж має багаторічну історію та є структурним підрозділом Київського національного університету будівництва та архітектури, що дозволяє забезпечувати наступність освіти для студентів. Основною місією закладу є підготовка висококваліфікованих фахівців у галузі будівництва, архітектури, інформаційних технологій та економіки.

Студентська спільнота КІФК КНУБА є важливою складовою освітнього середовища коледжу і налічує понад 1200 студентів різних спеціальностей. Активне студентське життя проявляється у діяльності студентського самоврядування, організації культурно-масових заходів, спортивних змагань, наукових конференцій та творчих проєктів. Студентська спільнота характеризується різноманіттям інтересів та прагненням до професійного розвитку, що відображається у великій кількості гуртків, клубів за інтересами та секцій.

Особливістю студентської спільноти коледжу є її активна громадянська позиція та залученість до вирішення актуальних питань навчального процесу. Студентське самоврядування коледжу має представництво у Педагогічній раді та бере участь у розробці нормативних документів, що стосуються освітнього процесу та студентського життя. Важливим напрямком діяльності студентської спільноти є волонтерство та благодійність, що сприяє формуванню соціальної відповідальності у майбутніх фахівців

Студентська спільнота КІФК КНУБА має значний творчий потенціал, який реалізується через театральну студію, музичні гуртки, проведення тематичних вечорів та конкурсів талантів. У коледжі функціонують спортивні секції з різних

видів спорту, де студенти мають можливість розвивати свої фізичні здібності та представляти заклад на різноманітних змаганнях. Усі ці активності створюють насичене студентське життя, яке потребує ефективної системи комунікації та інформування.

Культурно-масова діяльність є однією з візитних карток коледжу, де регулярно проводяться концерти, вистави, конкурси та фестивалі. Кожного семестру організовуються прослуховування для бажаючих долучитися до театральних постановок, що дає можливість студентам проявити свої таланти. Ця активна творча діяльність потребує налагодженої системи анонсування подій, реєстрації учасників та ефективного обміну інформацією між організаторами та потенційними учасниками заходів.

1.2.Організаційна структура коледжу та роль студентської спільноти

Організаційна структура Київського індустріального фахового коледжу КНУБА побудована за ієрархічним принципом і включає адміністрацію, навчальну частину, методичний відділ, відділення за напрямками підготовки, циклові комісії, бібліотеку, господарську частину та інші підрозділи. На чолі закладу стоїть директор, якому підпорядковуються заступники з навчальної, виховної та господарської роботи. Така структура забезпечує комплексне функціонування освітнього закладу та координацію всіх напрямків його діяльності.

Важливу роль в організаційній структурі коледжу відіграє студентська спільнота, представлена органами студентського самоврядування. Студентське самоврядування має власну ієрархію, що включає студентську раду коледжу, студентські ради відділень та старостати академічних груп. Голова студентської ради входить до складу адміністративної ради коледжу, що забезпечує представництво інтересів студентів на найвищому рівні управління закладом. Студентське самоврядування взаємодіє з різними структурними підрозділами коледжу, зокрема з виховним, навчальним відділами та цикловими комісіями [3]

Функціональна взаємодія між студентською спільнотою та адміністрацією коледжу реалізується через систему зборів, засідань та нарад, де обговорюються актуальні питання студентського життя та навчального процесу. Студентська рада координує діяльність секторів за напрямками: навчально-науковий, культурно-масовий, спортивний, інформаційний та соціально-побутовий. Кожен сектор має власного керівника, який відповідає за організацію відповідних заходів та інформування студентів про свою діяльність.

Циклові комісії коледжу, як об'єднання викладачів певного напрямку, також тісно співпрацюють зі студентською спільнотою при організації навчально-виховних заходів, конференцій, олімпіад та конкурсів. Представники студентства залучаються до обговорення змісту освітніх програм, формування розкладу та визначення форм контролю знань. Така співпраця сприяє формуванню освітнього середовища, орієнтованого на потреби студентів та вимоги сучасного ринку праці.

У структурі коледжу функціонують різноманітні гуртки та секції, які очолюються викладачами, але активно залучають студентів до організації своєї діяльності. Театральна студія, спортивні секції, наукові гуртки стають осередками студентської активності та самореалізації. Координація їхньої роботи потребує налагодженої системи комунікації між адміністрацією, викладачами та студентами, що наразі є одним із викликів для коледжу .

Взаємодія між різними ланками організаційної структури коледжу реалізується через офіційні канали комунікації (накази, розпорядження, оголошення) та неформальні канали (соціальні мережі, месенджери). Однак, відсутність єдиної інформаційної платформи ускладнює координацію дій та своєчасне інформування всіх учасників освітнього процесу про актуальні події та заходи.

1.3.Аналіз нинішнього стану комп'ютеризації та комунікації в коледжі

Київський індустріальний фаховий коледж КНУБА має базову інформаційно-технічну інфраструктуру, що включає комп'ютерні класи,

локальну мережу, доступ до інтернету та офіційний веб-сайт закладу. Комп'ютерні класи обладнані сучасною технікою та використовуються переважно для проведення занять з інформатики та спеціалізованих дисциплін. Локальна мережа охоплює адміністративні підрозділи та навчальні аудиторії, забезпечуючи базовий рівень електронної взаємодії між підрозділами коледжу.

Офіційний веб-сайт коледжу виконує переважно презентаційну функцію і містить загальну інформацію про заклад, спеціальності, умови вступу та основні новини. Однак, він не є інтерактивною платформою і не забезпечує повноцінної комунікації між учасниками освітнього процесу. Оновлення інформації на сайті відбувається централізовано адміністратором, що не дозволяє оперативно висвітлювати всі події та заходи, особливо ті, що організовуються студентською спільнотою.

Для комунікації між студентами та викладачами використовуються різноманітні канали: електронна пошта, месенджери (Viber, Telegram), соціальні мережі (Facebook, Instagram), а також традиційні оголошення на інформаційних стендах. Така розрізненість комунікаційних каналів призводить до дублювання інформації та її фрагментації, коли частина студентів може не отримати важливі повідомлення через те, що не користується певним каналом комунікації. Крім того, відсутня єдина база даних студентів, що ускладнює адресну розсилку повідомлень та інформування цільових груп .

Для організації навчального процесу в коледжі частково використовуються системи дистанційного навчання, зокрема Google Classroom та Moodle, проте ці платформи орієнтовані переважно на навчальну діяльність і не охоплюють інші аспекти студентського життя. Відсутність єдиної інформаційної системи для анонсування культурно-масових заходів, спортивних змагань, конкурсів та прослуховувань ускладнює залучення студентів до позанавчальної діяльності та знижує їхню інформованість про можливості самореалізації в коледжі.

Окремою проблемою є організація обміну інформацією щодо загублених та знайдених речей. Наразі ця взаємодія відбувається стихійно через особисті

контакти, інформаційні стенди або групи в соціальних мережах, що не забезпечує системного підходу до вирішення цього питання. Студенти часто стикаються з труднощами при спробі знайти загублені речі або повернути знайдені власникам через відсутність централізованої системи обліку таких випадків.

У коледжі відсутня також єдина система для збору зворотного зв'язку від студентів, зокрема анонімних повідомлень та скарг, що ускладнює виявлення проблемних питань в організації навчального процесу та студентського життя. Це обмежує можливості адміністрації оперативно реагувати на проблеми та покращувати умови навчання та проживання студентів.

1.4. Функціональне моделювання та аналіз існуючих бізнес-процесів комунікації

Функціональна модель інформаційного обміну в коледжі

Функціональна модель інформаційного обміну в КІФК КНУБА представляє собою складну систему взаємодії між різними суб'єктами освітнього процесу та адміністративними підрозділами. На вершині цієї моделі знаходиться адміністрація коледжу, яка генерує офіційну інформацію у вигляді наказів, розпоряджень і офіційних оголошень. Ця інформація передається до навчальної частини, яка, в свою чергу, розповсюджує її через завідувачів відділень до кураторів груп і безпосередньо до студентів. Така вертикальна модель комунікації забезпечує передачу офіційної інформації, але характеризується значними часовими затримками та можливим спотворенням інформації на кожному етапі її передачі.

Паралельно з вертикальною існує горизонтальна модель комунікації, яка забезпечує обмін інформацією між студентами різних груп та курсів, а також між студентами та викладачами. Ця модель реалізується переважно через неформальні канали комунікації – соціальні мережі, месенджери та особисте спілкування [4]. Такий спосіб обміну інформацією є більш оперативним, але

менш надійним з точки зору достовірності інформації та охоплення всієї цільової аудиторії.

Особливим елементом функціональної моделі є студентське самоврядування, яке виступає як комунікаційний міст між адміністрацією та студентською спільнотою. Воно генерує власні інформаційні повідомлення стосовно студентських ініціатив, заходів та проектів, а також передає запити та пропозиції від студентства до адміністрації. Ця комунікація здійснюється як через офіційні канали (звернення, протоколи засідань), так і через неформальне спілкування з представниками адміністрації .

У сфері організації культурно-масових заходів інформаційний обмін відбувається від організаторів (студентська рада, викладачі-куратори творчих гуртків) до потенційних учасників та глядачів. Цей процес включає анонсування заходів, реєстрацію учасників, координацію підготовки та звітування про результати. Наразі ці функції реалізуються через різноманітні канали комунікації без єдиної системи, що призводить до дублювання інформації та неповного охоплення цільової аудиторії.

Функціональна модель інформаційного обміну щодо загублених та знайдених речей є найменш структурованою. Вона базується на особистих контактах, оголошеннях на інформаційних стендах та в групах соціальних мереж. Відсутність централізованого реєстру таких випадків та єдиної платформи для розміщення відповідних оголошень знижує ефективність цього процесу та ускладнює повернення речей їхнім власникам.

Аналіз функціональної моделі інформаційного обміну в коледжі виявляє її фрагментованість, відсутність єдиної платформи для комунікації всіх учасників освітнього процесу та недостатню інтеграцію різних інформаційних потоків. Це обґрунтовує необхідність створення єдиної інформаційної системи, яка б забезпечила централізацію та структуризацію комунікаційних процесів в коледжі.

Функціональні моделі наведено у Додатку А

Виявлені проблеми

В результаті аналізу функціональної моделі інформаційного обміну в КІФК КНУБА було виявлено ряд суттєвих проблем, які знижують ефективність комунікації та негативно впливають на якість освітнього процесу та студентського життя. Основною проблемою є розрізненість інформаційних каналів, що призводить до фрагментації інформаційного простору коледжу. Студенти та викладачі вимушені використовувати різні платформи для отримання та передачі різних типів інформації: офіційний сайт для загальних новин, соціальні мережі для анонсів заходів, месенджери для оперативної комунікації, електронну пошту для офіційних повідомлень. Це вимагає постійного моніторингу всіх каналів та призводить до інформаційного перевантаження.

Друга важлива проблема – несвоєчасність поширення інформації. Відсутність єдиної платформи для оперативного розміщення оголошень та анонсів призводить до того, що інформація про важливі події часто доходить до цільової аудиторії із запізненням. Особливо гостро ця проблема стоїть щодо інформування про зміни в розкладі, перенесення занять, проведення додаткових консультацій та оголошення про прослуховування для участі в творчих заходах. Несвоєчасне отримання такої інформації знижує рівень участі студентів у позанавчальних активностях та може негативно впливати на навчальний процес.

Відсутність системи зворотного зв'язку та можливості анонімного звернення є ще однією критичною проблемою. Студенти не мають зручного та безпечного каналу для висловлення своїх скарг, пропозицій та зауважень щодо організації навчального процесу та студентського життя. Це призводить до накопичення невирішених проблем та зниження рівня довіри між студентами та адміністрацією коледжу. Особливо актуальною є проблема неможливості анонімного повідомлення про випадки порушення етичних норм, булінгу або інших конфліктних ситуацій.

Проблема обліку та координації заходів також виявлена в процесі аналізу. Відсутність єдиного календаря подій призводить до накладок у розкладі різних

заходів, що знижує рівень участі студентів та ефективність використання ресурсів коледжу. Організатори заходів не мають можливості оперативно перевірити зайнятість приміщень, обладнання та потенційних учасників, що ускладнює планування та реалізацію культурно-масових, спортивних та навчальних активностей.

Проблема обліку загублених та знайдених речей є однією з найбільш актуальних для студентської спільноти. Відсутність централізованої системи реєстрації таких випадків призводить до того, що значна кількість загублених речей не повертається до власників, навіть якщо вони були знайдені іншими студентами або працівниками коледжу. Це створює додатковий стрес для студентів та може призводити до матеріальних втрат.

Нарешті, відсутність єдиної бази даних контактів студентів та викладачів ускладнює адресну комунікацію та цільове інформування. Інформація про зміни, що стосуються конкретних груп або окремих студентів, часто не доходить до адресатів через відсутність налагодженого каналу прямої комунікації. Це призводить до комунікаційних розривів та знижує ефективність управління освітнім процесом.

Пропозиції щодо усунення наявних проблем

Для вирішення виявлених проблем інформаційного обміну в КІФК КНУБА пропонується створення єдиної веб-орієнтованої інформаційної системи, яка централізує комунікаційні процеси та забезпечить інтеграцію різних інформаційних потоків. Така система повинна стати основною платформою для обміну інформацією між всіма учасниками освітнього процесу, замінивши розрізнені канали комунікації єдиним інформаційним простором. Центральним елементом цієї системи має стати веб-сайт студентської спільноти, доступний з різних пристроїв та зручний для використання як студентами, так і викладачами.

Для вирішення проблеми розрізненості інформаційних каналів пропонується створити модульну структуру сайту, де кожен модуль відповідатиме за певний тип інформації: модуль оголошень для офіційної інформації, модуль розкладу заходів для анонсування культурно-масових та

спортивних подій, модуль загублених/знайдених речей для обліку відповідних випадків. Така структура дозволить користувачам швидко знаходити потрібну інформацію, не переходячи між різними платформами, та забезпечить єдиний стандарт представлення різних типів даних.

Для забезпечення оперативності поширення інформації пропонується впровадити систему повідомлень, яка дозволить автоматично інформувати користувачів про нові оголошення, заходи та інші важливі події. Передбачається можливість налаштування фільтрів для отримання лише релевантної інформації, що дозволить уникнути інформаційного перевантаження. Крім того, система повинна мати функцію push-повідомлень для оперативного інформування про термінові зміни та важливі анонси.

Для вирішення проблеми зворотного зв'язку пропонується створити модуль анонімних повідомлень, який дозволить студентам безпечно висловлювати свої зауваження, скарги та пропозиції. Передбачається система модерації таких повідомлень, щоб уникнути зловживань, але зберегти можливість анонімного звернення у випадках, коли це необхідно. Цей модуль стане важливим інструментом для виявлення проблемних моментів в організації навчального процесу та студентського життя.

Для вирішення проблеми обліку та координації заходів пропонується впровадження інтерактивного календаря подій, який дозволить організаторам заздалегідь планувати заходи, перевіряти доступність приміщень та обладнання, а також інформувати потенційних учасників. Календар повинен мати функцію фільтрації за типами заходів, що дозволить студентам швидко знаходити події, які відповідають їхнім інтересам. Крім того, передбачається функція онлайн-реєстрації на заходи та прослуховування, що спростить організаційні процеси.

Для вирішення проблеми обліку загублених та знайдених речей пропонується створення відповідного модуля з можливістю розміщення детальних описів та фотографій предметів, а також контактної інформації для зв'язку. Цей модуль повинен мати систему пошуку за різними параметрами, що дозволить швидко знаходити інформацію про конкретні предмети.

Передбачається також функція автоматичного сповіщення користувачів про появу нових оголошень, що відповідають їхнім запитам.

Для вирішення проблеми адресної комунікації пропонується створення бази даних користувачів з різними рівнями доступу та можливістю групування за різними параметрами (курс, група, спеціальність тощо). Це дозволить адміністрації та студентському самоврядуванню ефективно комунікувати з цільовими групами та забезпечить персоналізацію інформаційних потоків.

1.5.Огляд існуючих рішень для розв'язання виявлених проблем

Аналіз конкурентних веб-платформ

Для визначення оптимальних рішень при розробці веб-сайту для студентської спільноти КІФК КНУБА було проведено аналіз існуючих аналогічних платформ, що використовуються в інших навчальних закладах. Першою аналізованою системою є "KPI Campus" - інформаційна система Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського". Ця платформа забезпечує комплексну інформаційну підтримку студентського життя, включаючи електронний розклад, новини, оголошення та модуль електронного навчання. Перевагами системи є інтеграція з навчальним процесом, наявність мобільного додатку та зручний інтерфейс. Однак, система є доволі складною для впровадження в невеликому навчальному закладі та вимагає значних ресурсів для підтримки (Рис. 1.1).

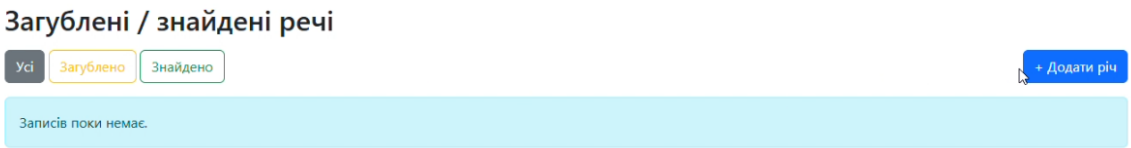

Рисунок 1.1 – Початковий інтерфейс модуля розкладу заходів із фільтрами та кнопкою додавання

Другою аналізованою платформою є "StudWay" - веб-ресурс для студентів Львівського національного університету імені Івана Франка. Ця система фокусується на позанавчальній активності студентів і включає модулі для анонсування заходів, обміну навчальними матеріалами та комунікації між

студентами. Перевагами платформи є простота використання, активне залучення студентського самоврядування до наповнення контентом та наявність функціоналу для обговорення актуальних питань студентського життя. Недоліками є обмежена інтеграція з офіційними інформаційними системами університету та відсутність мобільної версії.

Третьою розглянутою системою є "Campus" - інформаційна платформа Харківського національного університету радіоелектроніки. Ця система поєднує функції електронного деканату, новинного порталу та соціальної мережі для студентів і викладачів. Серед переваг - наявність особистих кабінетів з різними рівнями доступу, можливість формування електронного портфоліо студента та інтеграція з системою електронного навчання. Недоліками є складність адміністрування та висока вартість впровадження та підтримки.

Четвертою аналізованою платформою є "Student's Life" - веб-сайт студентського самоврядування Одеського національного університету імені І. І. Мечникова. Ця система фокусується на висвітленні студентського життя, анонсуванні заходів та підтримці студентських ініціатив. Перевагами є простота адміністрування, наявність модуля для збору зворотного зв'язку від студентів та інтеграція з соціальними мережами. Недоліками є обмежений функціонал щодо навчального процесу та відсутність мобільної версії.

Також було розглянуто міжнародний досвід, зокрема платформу "Unibuddy" - соціальну мережу для студентів британських університетів. Ця система дозволяє студентам комунікувати між собою, ділитися досвідом та знаходити однодумців для реалізації проектів. Особливістю платформи є акцент на формуванні студентської спільноти та підтримці нових студентів у процесі адаптації до університетського життя. Система має зручний інтерфейс, мобільний додаток та потужні інструменти для комунікації, однак орієнтована переважно на соціальну взаємодію, а не на інформаційну підтримку навчального процесу.

Проведений аналіз конкурентних веб-платформ дозволив виявити найбільш успішні рішення та функціональні можливості, які доцільно

імплементувати у веб-сайті для студентської спільноти КІФК КНУБА, а також врахувати недоліки існуючих систем для створення більш ефективного рішення для конкретних потреб коледжу.

Порівняння систем-аналогів

Для об'єктивного порівняння проаналізованих систем-аналогів було визначено ряд критеріїв, що відображають функціональні та технічні вимоги до веб-сайту для студентської спільноти КІФК КНУБА (Табл. 1.1).

Таблиця 1.1. Порівняння систем-аналогів за ключовими характеристиками

Критерій	**KPI Campus**	**StudWay**	**Campus**	**Student's Life**	**Unibuddy**
Функціональність	Висока	Середня	Висока	Середня	Середня
Зручність інтерфейсу	Середня	Висока	Середня	Висока	Дуже висока
Масштабованість	Висока	Середня	Висока	Низька	Середня
Інтеграція з іншими системами	Висока	Низька	Висока	Низька	Середня
Вартість впровадження	Висока	Низька	Висока	Низька	Середня
Вартість підтримки	Висока	Середня	Висока	Низька	Висока
Наявність мобільної версії	Так	Ні	Так	Ні	Так
Адаптивність до потреб закладу	Низька	Середня	Середня	Висока	Низька
Залучення студентів до наповнення	Низька	Висока	Середня	Висока	Дуже висока

Основними критеріями для порівняння стали: функціональність системи, зручність інтерфейсу, масштабованість, інтеграція з іншими системами, вартість впровадження та підтримки, наявність мобільної версії, можливість адаптації до специфічних потреб коледжу та залучення студентів до наповнення контентом.

За критерієм функціональності найбільш повним рішенням є система "KPI Campus", яка має найширший набір модулів для підтримки різних аспектів студентського життя. Однак, більшість цих функцій орієнтовані на потреби великого університету і можуть бути надлишковими для коледжу. Системи "StudWay" та "Student's Life" мають більш обмежений, але достатній для невеликого закладу функціонал, що фокусується на позанавчальних активностях. Система "Campus" займає проміжне положення, пропонуючи збалансований набір функцій для навчального процесу та студентського життя.

Щодо зручності інтерфейсу, найбільш інтуїтивно зрозумілим є інтерфейс системи "Unibuddy", орієнтований на молодіжну аудиторію та соціальну взаємодію. Системи "StudWay" та "Student's Life" також мають досить простий та зручний інтерфейс, тоді як "KPI Campus" та "Campus" характеризуються більш складною структурою, що вимагає певного часу для освоєння. За критерієм масштабованості лідерами є "KPI Campus" та "Campus", архітектура яких дозволяє легко додавати нові модулі та функції без суттєвої перебудови системи.

За критерієм інтеграції з іншими системами лідирують "KPI Campus" та "Campus", які мають APIs для взаємодії з іншими інформаційними системами навчального закладу. Системи "StudWay" та "Student's Life" мають обмежені можливості інтеграції, але пропонують базове підключення до соціальних мереж для поширення контенту. "Unibuddy" має середній рівень інтеграції, переважно з популярними соціальними платформами.

Суттєві відмінності спостерігаються за критерієм вартості впровадження та підтримки. Найбільш економічними рішеннями є "StudWay" та "Student's Life", які мають відкритий код та можуть бути впроваджені власними силами IT-відділу закладу. Натомість, "KPI Campus", "Campus" та "Unibuddy" вимагають значних фінансових вкладень для придбання ліцензій та технічної підтримки.

Щодо наявності мобільної версії, системи "KPI Campus", "Campus" та "Unibuddy" пропонують повноцінні мобільні додатки, що забезпечує зручний доступ до інформації з мобільних пристроїв. Системи "StudWay" та "Student's

Life" не мають спеціалізованих мобільних додатків, але їхні веб-інтерфейси частково адаптовані для мобільних пристроїв.

За критерієм адаптивності до специфічних потреб навчального закладу найбільш гнучкою є система "Student's Life", яка дозволяє легко налаштовувати структуру та функціонал відповідно до конкретних вимог. Системи "StudWay" та "Campus" також пропонують певний рівень кастомізації, тоді як "KPI Campus" та "Unibuddy" мають більш жорстку архітектуру, що ускладнює їх адаптацію.

Нарешті, за критерієм залучення студентів до наповнення контентом лідерами є "Unibuddy", "StudWay" та "Student's Life", які мають зручні інструменти для створення та редагування контенту користувачами з різними рівнями доступу. Системи "KPI Campus" та "Campus" орієнтовані більше на централізоване наповнення контентом адміністраторами системи.

Проведений порівняльний аналіз дозволяє зробити висновок, що жодна з розглянутих систем повністю не відповідає специфічним потребам КІФК КНУБА. Оптимальним рішенням буде розробка власної системи з урахуванням найкращих практик та функціональних можливостей проаналізованих аналогів, але з фокусом на конкретні потреби коледжу та його студентської спільноти.

1.6. Техніко-економічне обґрунтування впровадження веб-сайту

Впровадження веб-сайту для студентської спільноти КІФК КНУБА вимагає обґрунтування з точки зору технічної доцільності та економічної ефективності. Для оцінки економічної ефективності проєкту необхідно розрахувати витрати на розробку та впровадження системи, а також оцінити потенційні вигоди від її використання. Витрати на розробку включають оплату праці розробників, витрати на обладнання та програмне забезпечення, а також витрати на тестування та впровадження системи.

Розрахунок витрат на розробку базується на оцінці трудомісткості проєкту. З урахуванням функціональних вимог до системи, загальна трудомісткість розробки оцінюється в 280 людино-годин, з яких 70 годин припадає на аналіз та проєктування, 140 годин на розробку програмного забезпечення, 35 годин на

тестування та 35 годин на впровадження та документування. При середній ставці оплати праці розробника веб-додатків на Python 350 грн/год, загальні витрати на оплату праці становлять 98 000 грн.

Витрати на програмне забезпечення для розробки системи мінімізуються завдяки використанню відкритих технологій та інструментів, таких як Python, Flask, SQLite та Bootstrap. Усі необхідні компоненти (Python, Flask, SQLAlchemy, Bootstrap) є безкоштовними та мають відкритий вихідний код. Витрати на хостинг та доменне ім'я для розміщення веб-сайту становлять приблизно 2 500 грн на рік. Додаткові витрати на технічне забезпечення мінімальні, оскільки для розробки та підтримки системи використовуватиметься існуюча інфраструктура коледжу. Витрати на розробку та впровадження зображені нижче (Табл. 1.2).

Таблиця 1.2. Витрати на розробку та впровадження веб-сайту

Стаття витрат	**Сума, грн**
Оплата праці розробників	96 000
Хостинг та доменне ім'я (на рік)	3 000
Витрати на тестування	8 000
Витрати на впровадження та навчання персоналу	10 000
Загальні витрати	117 000

Вигоди від впровадження системи можна розділити на прямі економічні вигоди та непрямі соціальні та освітні вигоди. До прямих економічних вигод відносяться економія часу працівників коледжу на інформування студентів, зменшення витрат на друк оголошень та інформаційних матеріалів, а також зниження адміністративних витрат на організацію заходів завдяки автоматизації процесів реєстрації та координації.

Оцінка економії часу базується на припущенні, що впровадження системи дозволить скоротити час на інформування студентів та координацію заходів на 35%. При середньому часі, витраченому на ці завдання, у 18 годин на тиждень та середній ставці оплати праці адміністративного персоналу 220 грн/год, річна економія становить приблизно 58 500 грн (18 годин × 0.35 × 220 грн × 52 тижні).

Зменшення витрат на друк оголошень та інформаційних матеріалів оцінюється в 8 500 грн на рік, виходячи з поточних річних витрат коледжу на ці цілі. Зниження адміністративних витрат на організацію заходів оцінюється в 12 000 грн на рік, враховуючи економію на оплаті праці організаторів та зменшення витрат на комунікацію. Річні економічні вигоди від впровадження наведені у таблиці (Табл. 1.3).

Таблиця 1.3. Річні економічні вигоди від впровадження веб-сайту

Вид вигоди	Сума, грн/рік
Економія часу працівників	62 400
Зменшення витрат на друк	10 000
Зниження адміністративних витрат	15 000
Загальні річні вигоди	87 400

Розрахунок окупності інвестицій показує, що при загальних витратах на розробку та впровадження 116 000 грн та річних вигодах 79 000 грн, період окупності проєкту становить приблизно 1,47 роки (116 000 / 79 000). З урахуванням постійних витрат на підтримку системи (хостинг, технічна підтримка) у розмірі 12 000 грн на рік, чиста річна економія становить 67 000 грн (79 000 - 12 000), що дає скоригований період окупності 1,73 роки.

Особливою перевагою обраного технологічного стеку (Python + Flask + SQLite) є його економічність та простота підтримки. Відсутність ліцензійних витрат на програмне забезпечення, мінімальні вимоги до апаратного забезпечення завдяки ефективності SQLite, та можливість підтримки системи власними силами IT-відділу коледжу значно знижують загальну вартість володіння системою.

Крім прямих економічних вигод, впровадження веб-сайту матиме значні непрямі вигоди, які складно оцінити кількісно: підвищення якості комунікації між адміністрацією та студентами, покращення інформованості студентів про події та можливості самореалізації, підвищення рівня залученості студентів до позанавчальних активностей, зменшення кількості конфліктних ситуацій через

неправильну або несвоєчасну інформацію, а також підвищення загального рівня задоволеності студентів освітнім середовищем коледжу.

Техніко-економічне обґрунтування показує, що розробка та впровадження веб-сайту для студентської спільноти КІФК КНУБА з використанням відкритих технологій Python та Flask є економічно доцільним проєктом з прийнятним періодом окупності та значними потенційними вигодами як економічного, так і соціально-освітнього характеру.

1.7.Обґрунтування доцільності проєктування і розроблення веб-сайту

Проведений аналіз поточного стану інформаційного обміну в КІФК КНУБА, виявлені проблеми комунікації, порівняння існуючих аналогічних систем та техніко-економічне обґрунтування дозволяють зробити висновок про доцільність проєктування і розроблення веб-сайту для студентської спільноти коледжу. Основним аргументом на користь реалізації проєкту є наявність суттєвих комунікаційних проблем, які негативно впливають на якість освітнього процесу та студентського життя, і які можуть бути ефективно вирішені за допомогою запропонованої системи.

Розрізненість інформаційних каналів та відсутність єдиної платформи для комунікації створюють значні незручності для всіх учасників освітнього процесу. Впровадження веб-сайту студентської спільноти дозволить централізувати інформаційні потоки, забезпечити єдиний стандарт представлення різних типів даних та спростити доступ до актуальної інформації. Це особливо важливо в контексті активного студентського життя коледжу, яке включає різноманітні культурно-масові, спортивні та освітні заходи.

Проблема несвоєчасності поширення інформації також може бути ефективно вирішена за допомогою веб-сайту, який забезпечить оперативне розміщення оголошень та анонсів з можливістю автоматичного сповіщення користувачів про нові публікації. Це підвищить рівень інформованості студентів про важливі події та зміни в розкладі, що позитивно вплине на рівень їхньої участі в різних активностях та на загальну організацію навчального процесу.

Впровадження модуля анонімних повідомлень та скарг створить важливий канал зворотного зв'язку, який дозволить адміністрації коледжу своєчасно виявляти та вирішувати проблемні питання. Це сприятиме формуванню довірливих відносин між студентами та адміністрацією, а також підвищить загальний рівень задоволеності студентів освітнім середовищем коледжу.

Створення модуля обліку загублених та знайдених речей дозволить вирішити одну з найбільш актуальних проблем студентського життя. Централізована система реєстрації таких випадків підвищить ймовірність повернення загублених речей їхнім власникам, що зменшить стрес та можливі матеріальні втрати для студентів.

Важливим аргументом на користь розробки власного веб-сайту, а не використання існуючих систем-аналогів, є можливість максимальної адаптації функціоналу до специфічних потреб коледжу. Проведений аналіз показав, що жодна з існуючих систем не відповідає повністю вимогам КІФК КНУБА: системи великих університетів мають надмірний функціонал та високу вартість впровадження, а більш прості системи не забезпечують необхідної інтеграції з навчальним процесом.

Техніко-економічне обґрунтування підтвердило економічну доцільність проєкту з прийнятним періодом окупності інвестицій. Крім прямих економічних вигод, впровадження веб-сайту матиме значні непрямі вигоди, які сприятимуть підвищенню якості освітнього середовища та формуванню активної студентської спільноти.

Суттєвим аргументом є також те, що розробка веб-сайту силами студентів-програмістів коледжу в рамках дипломного проєктування дозволить значно зменшити витрати на розробку та забезпечить максимальне врахування потреб цільової аудиторії. Крім того, це створить цінний досвід практичного застосування знань та навичок для студентів, залучених до проєкту.

Таким чином, проєктування і розроблення веб-сайту для студентської спільноти КІФК КНУБА є доцільним та обґрунтованим рішенням, яке дозволить

підвищити якість комунікації, покращити інформованість студентів, забезпечити зворотний зв'язок та вирішити ряд практичних проблем студентського життя.

РОЗДІЛ 2
ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ

2.1.Загальні положення

Технічне завдання на розробку веб-сайту для студентської спільноти Київського індустріального фахового коледжу КНУБА (далі - Система) розроблено відповідно до результатів дослідження предметної області та виявлених проблем інформаційного обміну в коледжі. Дане технічне завдання визначає вимоги та порядок розробки Системи, її призначення, функціональні можливості, технічні параметри та інші характеристики, необхідні для успішної реалізації проєкту.

Основою для розробки Системи є результати аналізу існуючих аналогічних платформ, техніко-економічне обґрунтування та пропозиції щодо усунення виявлених проблем комунікації в коледжі. Система розробляється з урахуванням сучасних вимог до веб-додатків, з використанням відкритих технологій та інструментів, що забезпечують надійність, безпеку та зручність використання.

Результати розробки Системи оформлюються у вигляді повністю функціонального веб-сайту, розміщеного на хостингу з доменним ім'ям, що відповідає назві та призначенню ресурсу. Також до результатів розробки відносяться: програмний код, база даних, технічна документація, інструкції для користувачів різних категорій та матеріали для навчання адміністраторів системи.

У випадку необхідності внесення змін до технічного завдання в процесі розробки, такі зміни повинні бути узгоджені з зацікавленими сторонами та документально оформлені у вигляді додатків до даного технічного завдання. Таким чином буде забезпечено відповідність кінцевого продукту актуальним потребам коледжу та його студентської спільноти.

2.2. Призначення і цілі створення системи

Система призначена для забезпечення централізованого інформаційного обміну між усіма учасниками освітнього процесу Київського індустріального фахового коледжу КНУБА з фокусом на потреби студентської спільноти. Веб-сайт має стати основною платформою для комунікації, анонсування заходів, обміну інформацією та зворотного зв'язку, замінивши розрізнені канали комунікації єдиним інформаційним простором.

Основною метою створення Системи є підвищення якості комунікації між адміністрацією, викладачами та студентами коледжу шляхом забезпечення оперативного доступу до актуальної інформації про навчальний процес, позанавчальні заходи та інші аспекти студентського життя. Додатковими цілями є спрощення процесів організації та координації заходів, забезпечення прозорості інформаційних потоків та створення ефективного зворотного зв'язку між студентами та адміністрацією коледжу.

Система має вирішити ряд конкретних завдань, зокрема: централізувати розміщення оголошень та анонсів, забезпечити оперативне інформування про зміни в розкладі та організацію заходів, створити єдиний календар подій, налагодити облік загублених та знайдених речей, забезпечити можливість анонімного зворотного зв'язку, та надати платформу для обговорення актуальних питань студентського життя. Всі ці завдання спрямовані на формування ефективної інформаційної інфраструктури коледжу.

Створення веб-сайту має забезпечити формування активної студентської спільноти, підвищення рівня залученості студентів до позанавчальних активностей, покращення організації студентського життя та створення комфортного інформаційного середовища. Це сприятиме загальному підвищенню якості освітнього процесу та формуванню позитивного іміджу коледжу як сучасного навчального закладу, що використовує інноваційні підходи до організації комунікації.

2.3.Характеристика об'єкта автоматизації

Об'єктом автоматизації є процеси інформаційного обміну в Київському індустріальному фаховому коледжі КНУБА, зокрема комунікація між адміністрацією, викладачами та студентами, анонсування та координація заходів, облік загублених та знайдених речей, збір зворотного зв'язку та обговорення актуальних питань студентського життя. Наразі ці процеси реалізуються через розрізнені канали комунікації без єдиної системи координації та стандартизації.

КІФК КНУБА є навчальним закладом з контингентом близько 1200 студентів та 120 викладачів і співробітників. Коледж має розвинену інфраструктуру, що включає навчальні корпуси, гуртожитки, спортивні та культурні об'єкти. Активне студентське життя представлене діяльністю студентського самоврядування, різноманітними гуртками, спортивними секціями, творчими колективами та волонтерськими ініціативами. Всі ці активності генерують значні інформаційні потоки, які потребують ефективної системи управління та координації.

Поточний стан комп'ютеризації коледжу включає комп'ютерні класи, локальну мережу, доступ до інтернету та офіційний веб-сайт закладу. Однак, ці ресурси не забезпечують повноцінної інтерактивної комунікації між усіма учасниками освітнього процесу. Комунікація відбувається через різноманітні канали: електронну пошту, месенджери, соціальні мережі, інформаційні стенди, що призводить до фрагментації інформаційного простору та ускладнює доступ до актуальної інформації.

Особливістю об'єкта автоматизації є наявність різних категорій користувачів з різними потребами та рівнями доступу: адміністрація коледжу, викладачі, представники студентського самоврядування, студенти різних відділень та курсів. Кожна категорія має специфічні функціональні потреби, які повинні бути враховані при розробці Системи. Крім того, об'єкт автоматизації характеризується динамічністю інформаційних процесів, необхідністю

оперативного оновлення інформації та забезпечення її актуальності для всіх користувачів.

2.4. Вимоги до системи

Вимоги до системи в цілому

Система повинна бути реалізована у вигляді веб-додатку з адаптивним дизайном, що забезпечує коректне відображення на різних пристроях (комп'ютери, планшети, смартфони). Архітектура системи має бути клієнт-серверною з використанням сучасних веб-технологій та фреймворків, що забезпечують надійність, безпеку та масштабованість. Програмна реалізація серверної частини повинна бути здійснена на мові Python з використанням фреймворку Flask, клієнтська частина - з використанням HTML5, CSS3 та фреймворку Bootstrap.

Система повинна забезпечувати багатокористувацький режим роботи з розмежуванням прав доступу залежно від ролі користувача. Передбачається наявність таких ролей: адміністратор, модератор, представник студентського самоврядування, викладач, студент, гість. Кожна роль повинна мати відповідний набір прав для роботи з різними модулями системи. Автентифікація користувачів повинна здійснюватися через форму входу з використанням електронної пошти та паролю, з можливістю відновлення забутого паролю через розширення Flask-Login.

Система повинна забезпечувати високий рівень безпеки даних, включаючи захист від несанкціонованого доступу, хакерських атак та втрати інформації. Всі паролі користувачів повинні зберігатися в зашифрованому вигляді з використанням werkzeug.security, а передача чутливих даних має здійснюватися через захищені канали зв'язку (HTTPS). Система повинна мати функціонал резервного копіювання бази даних та можливість відновлення даних у випадку збоїв.

Вимоги до функцій системи

Система повинна забезпечувати реєстрацію та авторизацію користувачів з різними рівнями доступу. Реєстрація нових користувачів має здійснюватися з обов'язковою верифікацією електронної пошти. Для студентів та викладачів коледжу повинна бути можливість автоматичного підтвердження статусу через домен електронної пошти коледжу. Система повинна забезпечувати управління профілем користувача, включаючи налаштування отримання повідомлень та сповіщень про нові публікації.

Модуль оголошень повинен забезпечувати можливість створення, редагування та видалення оголошень різних категорій: загальні, навчальні, культурно-масові, спортивні та інші. Кожне оголошення має містити заголовок, текст, дату публікації, термін актуальності, категорію та теги для зручного пошуку. Користувачі з відповідними правами повинні мати можливість модерувати оголошення. Система повинна забезпечувати пошук оголошень за різними параметрами (назва, категорія, дата, ключові слова) та сортування результатів. Також має бути реалізована функція автоматичного сповіщення користувачів про нові оголошення відповідно до їхніх налаштувань.

Модуль розкладу заходів повинен забезпечувати створення та управління єдиним календарем подій коледжу. Кожна подія повинна містити назву, опис, дату та час початку і закінчення, місце проведення, категорію та контактну інформацію організаторів. Повинна бути можливість додавання подій, що повторюються з певною періодичністю. Система має забезпечувати різні представлення календаря: щоденне, тижневе, місячне, а також список подій за категоріями. Для культурно-масових заходів повинна бути реалізована функція онлайн-реєстрації учасників та прослуховувань, з можливістю встановлення обмеження на кількість учасників.

Модуль загублених/знайдених речей повинен забезпечувати можливість розміщення оголошень про загублені або знайдені предмети з детальним описом, фотографіями та контактною інформацією. Система повинна забезпечувати пошук за категоріями предметів, датою та місцем знахідки/втрати. Користувачі

повинні мати можливість підписатися на сповіщення про нові оголошення, що відповідають їхнім пошуковим запитам. Також система повинна забезпечувати автоматичне архівування неактуальних оголошень після певного періоду часу.

Модуль анонімних повідомлень та скарг повинен забезпечувати можливість відправки повідомлень адміністрації коледжу без розкриття особистості відправника. Кожне повідомлення повинно мати тему, текст та категорію (пропозиція, скарга, повідомлення про порушення тощо). Система має забезпечувати модерацію повідомлень для уникнення зловживань, але зберігати анонімність відправника. Адміністратори повинні мати можливість відповідати на повідомлення через систему, не розкриваючи особистість відправника. Також система повинна забезпечувати статистику та аналітику за типами повідомлень для виявлення системних проблем.

Вимоги до видів забезпечення

Інформаційне забезпечення системи повинно базуватися на реляційній базі даних SQLite версії 3.0 або вище. База даних повинна містити таблиці для зберігання інформації про користувачів, оголошення, заходи, загублені/знайдені речі, повідомлення та інші дані, необхідні для функціонування системи. Структура бази даних повинна забезпечувати цілісність даних, мінімізацію дублювання інформації та оптимальну продуктивність при роботі з даними. Для забезпечення безпеки даних повинна бути реалізована система резервного копіювання бази даних з можливістю відновлення даних у випадку збоїв.

Програмне забезпечення системи повинно бути реалізовано з використанням сучасних технологій веб-розробки. Серверна частина повинна бути розроблена на мові Python версії 3.8 або вище з використанням фреймворку Flask версії 2.0 або вище. Клієнтська частина повинна бути розроблена з використанням HTML5, CSS3 та фреймворку Bootstrap версії 5.0 або вище. Для забезпечення функціональності системи повинні бути використані розширення Flask-Login для автентифікації, Flask-SQLAlchemy для роботи з базою даних, Jinja2 для шаблонізації, а також werkzeug.security для безпечного хешування паролів.

Технічне забезпечення системи повинно включати веб-сервер з наступними мінімальними характеристиками: процесор з частотою не менше 2 ГГц (2 ядра або більше), оперативна пам'ять не менше 4 ГБ, дисковий простір не менше 20 ГБ. Веб-сервер повинен працювати під управлінням операційної системи Linux (Ubuntu 20.04 LTS або вище) з встановленим WSGI-сервером (наприклад, Gunicorn) для розгортання Flask-додатку. Для забезпечення безпеки передачі даних повинен бути встановлений SSL-сертифікат з підтримкою протоколу HTTPS.

Лінгвістичне забезпечення системи повинно включати україномовний інтерфейс користувача з можливістю додавання інших мовних пакетів у майбутньому. Всі повідомлення системи, елементи інтерфейсу, довідкова інформація повинні бути представлені українською мовою [5]. Система повинна коректно обробляти та відображати символи української мови, включаючи введення, зберігання та виведення текстової інформації з використанням кодування UTF-8. Також система повинна забезпечувати можливість пошуку інформації українською мовою з урахуванням особливостей морфології.

2.5.Діаграма Ганта та календарний план розроблення проєкту

Розробка веб-сайту для студентської спільноти КІФК КНУБА передбачає виконання комплексу робіт, які можна розділити на кілька послідовних етапів. Для ефективного управління проєктом та контролю за виконанням робіт розроблено календарний план та діаграму Ганта, що відображає послідовність та тривалість кожного етапу. Загальна тривалість проєкту складає 12 тижнів, починаючи з 1 червня 2025 року і закінчуючи 23 серпня 2025 року. Діаграму Ганта, що відображає послідовність та тривалість кожного етапу, наведено на рисунку 2.1.

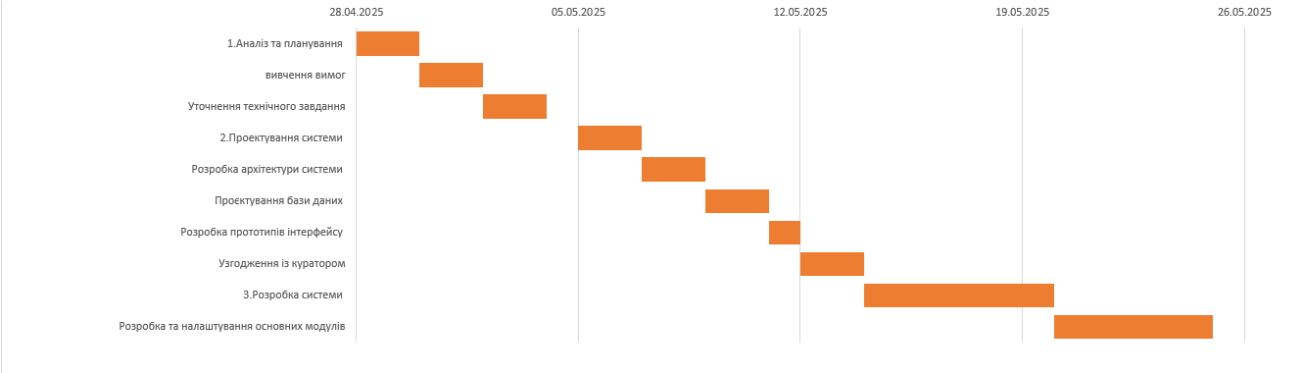

Рисунок 2.1 - Діаграма Ганта розроблення веб-сайту студентської спільноти КІФК КНУБА

Перший етап – аналіз та планування включає детальне вивчення вимог, уточнення технічного завдання, вибір технологій та інструментів розробки, а також планування архітектури системи. Цей етап є критично важливим для успіху проєкту, оскільки визначає основні технічні та функціональні параметри майбутньої системи. Результатом етапу є затверджене технічне завдання та детальний план розробки.

Другий етап – проєктування системи включає розробку архітектури системи, проєктування бази даних, розробку прототипів інтерфейсу користувача та узгодження їх із замовником. Цей етап передбачає створення UML-діаграм, схеми бази даних, wireframes та mockups інтерфейсу користувача. Результатом етапу є затверджені технічні рішення та дизайн-макети системи.

Третій етап – розробка системи є найбільш тривалим та включає програмування серверної та клієнтської частин системи, розробку бази даних, інтеграцію компонентів та реалізацію функціональних модулів. Цей етап розбито на підетапи відповідно до основних модулів системи: система авторизації та управління користувачами, модуль оголошень, модуль розкладу заходів, модуль загублених/знайдених речей, модуль анонімних повідомлень та скарг. Кожен підетап включає розробку, тестування та інтеграцію відповідного модуля.

Четвертий етап – тестування та оптимізація включає різні види тестування: функціональне, інтеграційне, навантажувальне, тестування безпеки та юзабіліті-тестування. За результатами тестування проводиться оптимізація системи,

виправлення виявлених помилок та покращення продуктивності. Цей етап є критичним для забезпечення якості та надійності системи перед введенням її в експлуатацію.

П'ятий етап – впровадження та документування включає розгортання системи на виробничому сервері, міграцію даних (якщо необхідно), навчання адміністраторів та ключових користувачів, а також розробку технічної документації та інструкцій для користувачів. Результатом етапу є повністю функціональна система, готова до експлуатації, та комплект документації для різних категорій користувачів.

Діаграма Ганта наочно відображає послідовність та взаємозв'язок всіх етапів та підетапів проєкту, визначає критичний шлях та ключові точки контролю. Цей інструмент дозволяє ефективно управляти ресурсами проєкту та контролювати дотримання термінів виконання робіт. Календарний план та діаграма Ганта є гнучкими документами, які можуть бути адаптовані відповідно до змін у вимогах або умовах реалізації проєкту.

РОЗДІЛ 3
ПРОЄКТУВАННЯ, СТВОРЕННЯ ТА АПРОБАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Опис та обґрунтування вибору програмно-технічних засобів розроблення

Вибір мови програмування та фреймворку

Для розробки веб-сайту студентської спільноти КІФК КНУБА було обрано мову програмування Python [1] версії 3.x та веб-фреймворк Flask [6]. Python є однією з найпопулярніших мов програмування для веб-розробки завдяки своїй простоті, читабельності коду та великій кількості бібліотек.

Flask є легким і гнучким веб-фреймворком для Python, який дозволяє швидко створювати веб-додатки. Основними перевагами Flask є:

- Мінімалістичний підхід та простота використання
- Гнучкість в архітектурних рішеннях
- Велика кількість розширень
- Відмінна документація та активна спільнота

Для роботи з автентифікацією використовується розширення Flask-Login, для роботи з базами даних - Flask-SQLAlchemy, що надає зручний ORM для взаємодії з базою даних.

Вибір системи управління базами даних

Для веб-сайту студентської спільноти КІФК КНУБА було обрано СУБД SQLite [7] - компактну, вбудовану реляційну базу даних, яка зберігає всі дані в одному файлі на диску.

Переваги SQLite для даного проєкту:

- Простота розгортання та налаштування
- Відсутність необхідності в окремому сервері БД
- Достатня продуктивність для очікуваної кількості користувачів (до 1500)
- Повна підтримка SQL стандартів
- Надійність та стабільність

- Відмінна інтеграція з Python через вбудований модуль sqlite3

SQLite підходить для проєктів малого та середнього масштабу, забезпечуючи всі необхідні функції реляційної СУБД без складності адміністрування.

Вибір допоміжних технологій та інструментів

Для створення інтерфейсу користувача використовується CSS [8] - фреймворк Bootstrap версії 5.3.3, підключений через CDN. Bootstrap забезпечує:

– Адаптивний дизайн для різних пристроїв
– Готові компоненти інтерфейсу
– Консистентний вигляд елементів

Шаблони створюються за допомогою Jinja2 - вбудованого шаблонізатора Flask.

Структура проєкту організована за принципами Flask:

– models.py - моделі даних (SQLAlchemy ORM)
– auth.py - модуль автентифікації (Blueprint)
– app.py - основний файл додатку з маршрутами
– templates/ - HTML шаблони
– static/ - статичні файли (CSS, зображення)

Для роботи з паролями використовується werkzeug.security з функціями generate_password_hash() та check_password_hash() для безпечного хешування та перевірки паролів. Система автентифікації побудована на основі розширення Flask-Login, що забезпечує управління сесіями користувачів та контроль доступу до захищених ресурсів.

3.2.Проєктування та створення бази даних

Логічна модель даних

Логічна модель даних [9] для веб-сайту студентської спільноти КІФК КНУБА відображає структуру та взаємозв'язки між різними сутностями системи на концептуальному рівні, незалежно від конкретної СУБД. Модель розроблена

з урахуванням функціональних вимог до системи та особливостей роботи з SQLAlchemy ORM у фреймворку Flask.

Центральною сутністю логічної моделі є "Користувач" (User), яка представляє всіх зареєстрованих користувачів системи та містить такі атрибути: унікальний ідентифікатор (id), ім'я користувача (username), електронна пошта (email), пароль (password, зберігається в зашифрованому вигляді за допомогою werkzeug.security), роль користувача (role: admin, moderator, student, teacher). Ця сутність пов'язана з більшістю інших сутностей системи через зовнішні ключі, оскільки практично всі дії в системі асоціюються з конкретним користувачем.

Сутність "Оголошення" (Announcement) представляє інформаційні повідомлення різних типів і містить атрибути: унікальний ідентифікатор (id), заголовок (title), зміст (content), дата створення (created_at з автоматичним встановленням поточної дати через datetime.utcnow). Ця сутність не має прямого зв'язку з користувачами в поточній реалізації, що дозволяє створювати оголошення від імені системи.

Сутність "Заняття" (Lesson) представляє елементи розкладу навчальних занять і містить атрибути: унікальний ідентифікатор (id), назва предмету (subject), викладач (teacher), аудиторія (classroom), назва групи (group_name, опціонально), дата проведення (date), дата закінчення (date_end, для періодичних занять), час проведення (time). Ця сутність дозволяє створювати як разові заняття, так і періодичні з вказанням діапазону дат.

Сутність "Подія" (Event) представляє різноманітні заходи, що проводяться в коледжі, і містить атрибути: унікальний ідентифікатор (id), назва заходу (title), опис (description), дата проведення (date), час проведення (time), місце проведення (location). Ця сутність пов'язана з сутністю "Коментар до події" (EventComment) відношенням "один до багатьох", що дозволяє користувачам обговорювати заходи.

Сутність "Коментар до події" (EventComment) представляє коментарі користувачів до подій і містить атрибути: унікальний ідентифікатор (id), зміст коментаря (content), дата створення (created_at), ідентифікатор події (event_id як

зовнішній ключ), ідентифікатор користувача (user_id як зовнішній ключ). Ця сутність пов'язана з сутностями "Користувач" та "Подія" відношеннями "багато до одного", що дозволяє відстежувати авторство коментарів та їх приналежність до конкретних подій.

Сутність "Загублена/знайдена річ" (LostFoundItem) представляє інформацію про загублені або знайдені предмети і містить атрибути: унікальний ідентифікатор (id), тип оголошення (type: 'lost' або 'found'), назва предмету (title), опис (description), контактна інформація (contact), дата створення (created_at), ідентифікатор користувача (user_id як зовнішній ключ). Ця сутність пов'язана з сутністю "Користувач" відношенням "багато до одного", що дозволяє відстежувати, хто розмістив оголошення.

Сутність "Спортивна подія" (SportEvent) представляє спортивні заходи та екскурсії і містить атрибути: унікальний ідентифікатор (id), назва заходу (title), опис (description), дата (date як рядок для гнучкості), місце проведення (place), час початку (start), іконка (icon для візуального представлення), дата створення (created_at). Ця сутність не має прямих зв'язків з іншими сутностями в поточній реалізації.

Сутність "Анонімне повідомлення" (AnonymousMessage) представляет повідомлення, надіслані користувачами без розкриття їхньої особистості, і містить атрибути: унікальний ідентифікатор (id), тема повідомлення (subject), зміст (message), дата створення (created_at). Ця сутність навмисно не має зв'язку з сутністю "Користувач", щоб забезпечити повну анонімність відправників.

Логічна модель даних характеризується простотою та ефективністю, що відповідає архітектурі Flask та можливостям SQLAlchemy ORM. Використання автоматичних первинних ключів (INTEGER PRIMARY KEY AUTOINCREMENT в SQLite) та зовнішніх ключів забезпечує цілісність даних та ефективну навігацію між пов'язаними записами, модель даних з відображенням всіх сутностей та зв'язків між ними представлена на рис. 3.1.

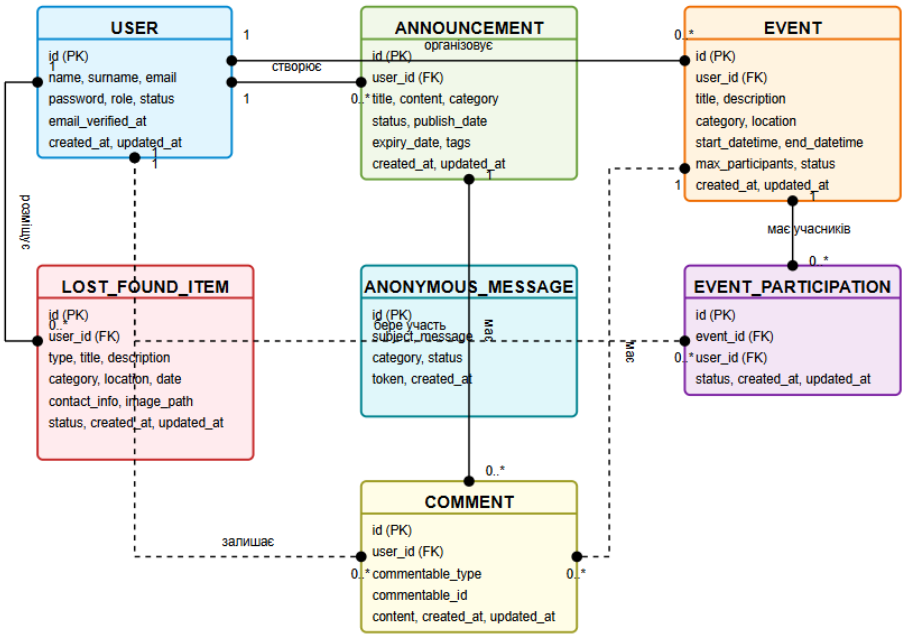

Рисунок 3.1 – Логічна модель даних веб-сайту студентської спільноти КІФК КНУБА

Створення та налаштування бази даних

. Процес створення та налаштування бази даних для веб-сайту студентської спільноти КІФК КНУБА включав кілька етапів, що враховували особливості роботи з SQLite та SQLAlchemy ORM у фреймворку Flask. Вся робота з базою даних організована через об'єктно-реляційне відображення (ORM), що забезпечує безпеку, зручність розробки та підтримки системи.

На першому етапі було налаштовано конфігурацію Flask-додатку для роботи з базою даних SQLite. У файлі app.py встановлено наступну конфігурацію:

```
pythonfrom flask import Flask
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['SECRET_KEY'] = 'supersecretkey123'
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///db.sqlite3'

db = SQLAlchemy()
```

```
db.init_app(app)
```

Параметр SQLALCHEMY_DATABASE_URI вказує на файл db.sqlite3 у кореневій директорії проєкту, де зберігатимуться всі дані системи. SECRET_KEY необхідний для роботи сесій та захисту форм від CSRF-атак.

На другому етапі було створено моделі даних у файлі models.py з використанням SQLAlchemy ORM. Кожна модель представляє таблицю в базі даних та наслідує від db.Model:

```
pythonfrom flask_sqlalchemy import SQLAlchemy
from flask_login import UserMixin
from datetime import datetime

db = SQLAlchemy()

class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(50), unique=True, nullable=False)
    email = db.Column(db.String(120), unique=True, nullable=False)
    password = db.Column(db.String(200), nullable=False)
    role = db.Column(db.String(20), nullable=False)
```

Кожна модель містить визначення полів через db.Column() з вказанням типу даних, обмежень та індексів. Використання nullable=False забезпечує обов'язковість заповнення полів, unique=True - унікальність значень, а primary_key=True визначає первинний ключ таблиці.

Третій етап включав налаштування зв'язків між таблицями через зовнішні ключі та відношення. Наприклад, для зв'язку коментарів з подіями та користувачами:

```
pythonclass EventComment(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    content = db.Column(db.Text, nullable=False)
    created_at = db.Column(db.DateTime, default=datetime.utcnow)
    event_id = db.Column(db.Integer, db.ForeignKey('event.id'))
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'))
    user = db.relationship('User', backref='comments')
```

Функція db.ForeignKey() створює зовнішній ключ, а db.relationship() встановлює відношення між моделями з автоматичним створенням зворотного зв'язку через backref.

Четвертий етап включав ініціалізацію бази даних та створення таблиць. Це відбувається автоматично при запуску додатку через виклик db.create_all() у контексті додатку:

```
pythonif __name__ == '__main__':
    with app.app_context():
        db.create_all()
        print("Кількість оголошень у базі:", Announcement.query.count())
    app.run(debug=True)
```

Функція db.create_all() аналізує всі визначені моделі та створює відповідні таблиці в базі даних SQLite, якщо вони ще не існують. Це забезпечує автоматичну ініціалізацію структури бази даних при першому запуску додатку.

Особливості роботи з SQLite в даному проєкті включають:

1. Автоматичне створення файлу бази даних - SQLite створює файл db.sqlite3 автоматично при першому зверненні до бази даних.

2. Вбудована підтримка типів даних - SQLite автоматично перетворює типи даних SQLAlchemy (Integer, String, Text, DateTime) у відповідні типи SQLite.

3. ACID-сумісність - SQLite забезпечує атомарність, консистентність, ізоляцію та довговічність транзакцій.

4. Простота резервного копіювання - вся база даних зберігається в одному файлі, що спрощує створення резервних копій.

П'ятий етап включав налаштування механізмів забезпечення цілісності даних через обмеження на рівні моделей та валідацію на рівні додатку. SQLAlchemy автоматично генерує SQL-запити з proper escaping, що захищає від SQL-ін'єкцій.

Для підтримки роботи з базою даних в процесі розробки використовувалися наступні команди:

```
python# Створення всіх таблиць
db.create_all()

# Видалення всіх таблиць (для повного перестворення)
db.drop_all()
db.create_all()

# Перевірка кількості записів
Announcement.query.count()
```

Налаштована система забезпечує надійне зберігання даних, ефективну роботу запитів та простоту адміністрування, що відповідає вимогам проєкту та очікуваному навантаженню системи.

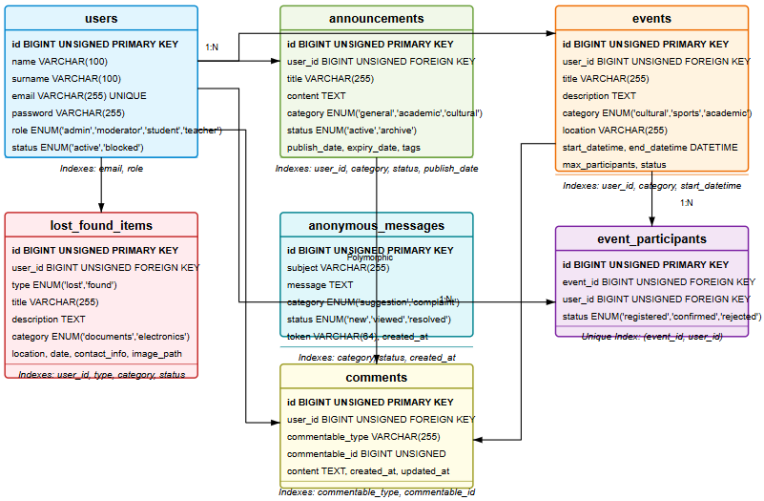

Рисунок 3.2 – Фізична модель даних веб-сайту студентської спільноти КІФК КНУБА

Для роботи з базою даних SQLite використовується ORM SQLAlchemy через розширення Flask-SQLAlchemy. Це дозволяє:

- Працювати з базою даних через об'єктно-орієнтований підхід
- Автоматично створювати таблиці на основі моделей Python
- Забезпечувати безпеку запитів та захист від SQL-ін'єкцій
- Легко мігрувати зміни в структурі бази даних

3.3.Реалізація функцій системи

Розробка системи аутентифікації та авторизації

Система аутентифікації та авторизації є одним з ключових компонентів веб-сайту студентської спільноти КІФК КНУБА, забезпечуючи безпечну ідентифікацію користувачів та контроль доступу до різних функцій системи. Розробка цього компоненту була виконана з використанням розширення Flask-Login та вбудованих можливостей Flask для забезпечення високого рівня безпеки та зручності використання.

Архітектура системи аутентифікації

Система аутентифікації побудована на основі розширення Flask-Login, яке надає готові інструменти для управління сесіями користувачів. Основні компоненти системи включають:

1. LoginManager - центральний компонент для управління автентифікацією
2. Blueprint auth - модульна організація маршрутів автентифікації
3. Модель User з UserMixin - інтеграція з Flask-Login
4. Хешування паролів - безпечне зберігання паролів
5. Контроль доступу - перевірка ролей та прав користувачів

Конфігурація Flask-Login

Ініціалізація системи аутентифікації відбувається в основному файлі додатку app.py:

```
pythonfrom flask_login import LoginManager, login_required, current_user

# Ініціалізація LoginManager
login_manager = LoginManager()
login_manager.login_view = 'auth.login'  # маршрут для перенаправлення неавторизованих користувачів
login_manager.init_app(app)

# Функція завантаження користувача за ID
@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))
```

Параметр login_view вказує маршрут, на який перенаправлятимуться неавторизовані користувачі при спробі доступу до захищених ресурсів. Функція user_loader використовується Flask-Login для завантаження об'єкта користувача з бази даних за його ідентифікатором.

Модель користувача

Модель User успадковує від UserMixin, що автоматично додає необхідні методи для роботи з Flask-Login:

```
pythonfrom flask_login import UserMixin

class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(50), unique=True, nullable=False)
    email = db.Column(db.String(120), unique=True, nullable=False)
    password = db.Column(db.String(200), nullable=False)
    role = db.Column(db.String(20), nullable=False)
```

UserMixin забезпечує реалізацію методів is_authenticated, is_active, is_anonymous та get_id(), які використовуються Flask-Login для управління станом автентифікації користувача.

Модуль автентифікації (Blueprint)

Логіка автентифікації винесена в окремий Blueprint auth.py для модульності та зручності підтримки:

```
pythonfrom flask import Blueprint, render_template, request, redirect, url_for, flash
from werkzeug.security import generate_password_hash, check_password_hash
from flask_login import login_user, logout_user, login_required

auth_bp = Blueprint('auth', __name__)
```

Реєстрація користувачів

Процес реєстрації включає валідацію даних, хешування пароля та створення нового користувача:

```
python@auth_bp.route('/register', methods=['GET', 'POST'])
def register():
```

```
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = generate_password_hash(request.form['password'])
        role = request.form['role']

        # Перевірка унікальності email та username відбувається на рівні бази даних
        new_user = User(username=username, email=email, password=password, role=role)
        db.session.add(new_user)
        db.session.commit()

        flash('Успішна реєстрація! Увійдіть.')
        return redirect(url_for('auth.login'))

    return render_template('register.html')
```

Використовується функція generate_password_hash() з бібліотеки werkzeug.security для створення безпечного хешу пароля з використанням алгоритму PBKDF2.

Вхід користувачів

Процес автентифікації включає перевірку облікових даних та встановлення сесії:

```
python@auth_bp.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
```

```
        # Пошук користувача за email
        user = User.query.filter_by(email=email).first()

        # Перевірка пароля
        if user and check_password_hash(user.password, password):
            login_user(user)  # Встановлення сесії користувача
            return redirect(url_for('home'))
        else:
            flash('Невірний логін або пароль')

    return render_template('login.html')
```

Функція check_password_hash() безпечно порівнює введений пароль з збереженим хешем. При успішній автентифікації login_user() встановлює сесію користувача.

Вихід користувачів

Процес виходу забезпечує безпечне завершення сесії:

```
python@auth_bp.route('/logout')
@login_required
def logout():
    logout_user()  # Видалення сесії користувача
    return redirect(url_for('auth.login'))
```

Контроль доступу та авторизація

Захист маршрутів здійснюється через декоратор @login_required:

```
python@app.route('/announcements')
@login_required
def announcements():
    # Цей маршрут доступний лише авторизованим користувачам
```

```
    all_announcements = Announcement.query.order_by(Announcement.created_at.desc()).all()
    return render_template('announcements.html', announcements=all_announcements)
```

Для перевірки ролей користувачів використовується атрибут current_user.role:

```
python@app.route('/announcements/create', methods=['GET', 'POST'])
@login_required
def create_announcement():
    # Перевірка прав доступу
    if current_user.role not in ['admin', 'moderator']:
        flash("У вас немає прав для створення оголошення.")
        return redirect(url_for('announcements'))

    # Логіка створення оголошення
    if request.method == 'POST':
        title = request.form['title']
        content = request.form['content']
        new_announcement = Announcement(title=title, content=content)
        db.session.add(new_announcement)
        db.session.commit()
        flash("Оголошення опубліковано.")
        return redirect(url_for('announcements'))

    return render_template('create_announcement.html')
```

Система ролей

Реалізована ієрархічна система ролей:

– admin - повний доступ до всіх функцій системи
– moderator - створення та модерація контенту

– teacher - доступ до навчальних функцій

– student - базовий доступ до перегляду та коментування

Безпека системи

Система забезпечує наступні аспекти безпеки:

– Хешування паролів - використання PBKDF2 через werkzeug.security

– Захист сесій - SECRET_KEY для підпису сесійних даних

– Валідація на стороні сервера - перевірка всіх вхідних даних

– Захист від CSRF - автоматичний захист форм Flask

– Контроль доступу - перевірка ролей для кожної дії

Розроблена система аутентифікації та авторизації забезпечує надійний захист веб-сайту студентської спільноти, зручність використання для користувачів різних категорій та гнучкість для подальшого розширення функціоналу. Скріншоти з програми наведено у Додатку Б

Реалізація модуля оголошень

Модуль оголошень є одним із ключових компонентів веб-сайту студентської спільноти КІФК КНУБА, який забезпечує централізоване розміщення та управління різними типами інформаційних повідомлень. Реалізація цього модуля була спрямована на створення зручного та функціонального інструменту для ефективної комунікації між адміністрацією, викладачами та студентами коледжу.

Архітектура модуля оголошень базується на паттерні MVC (Model-View-Controller), що забезпечує чітке розділення логіки, даних та представлення. Модель представлена класом Announcement, контролер AnnouncementController містить методи для обробки користувацьких запитів, а представлення реалізовані у вигляді Blade-шаблонів для різних сторінок модуля (Рис. 3.3).

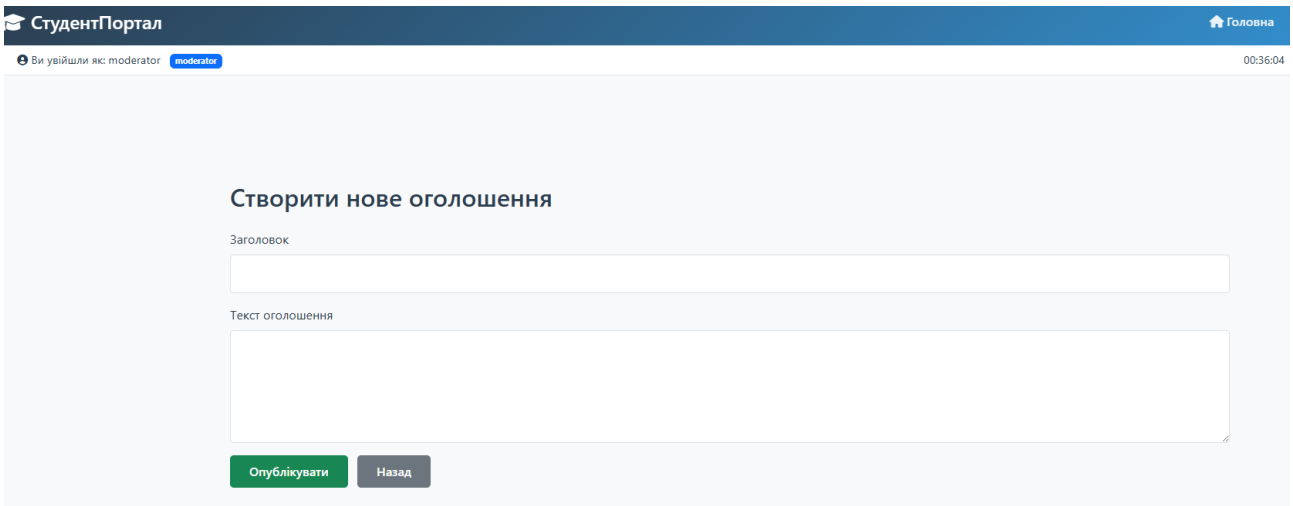

Рисунок 3.3 – Форма для створення нового оголошення в модулі оголошень

Функціонал модуля включає створення нових оголошень із зазначенням заголовка, змісту, категорії, терміну актуальності, тегів та статусу. Для редагування текстового контенту використовується WYSIWYG-редактор CKEditor. Для зручного доступу реалізовано функціонал пошуку та фільтрації оголошень за різними параметрами, а також система сортування результатів (Рис. 3.4 - 3.5).

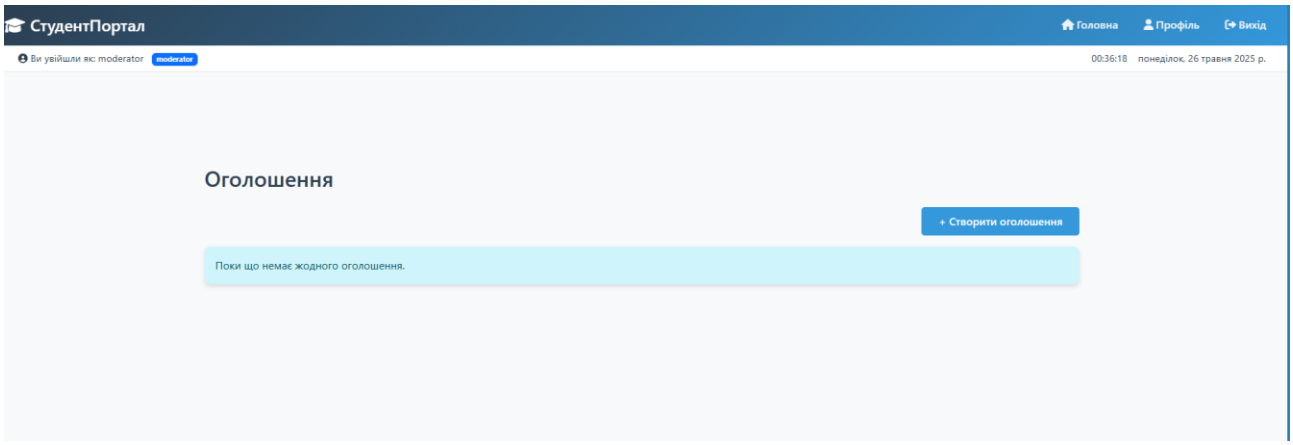

Рисунок 3.4 – Інтерфейс модуля оголошень при відсутності оголошень

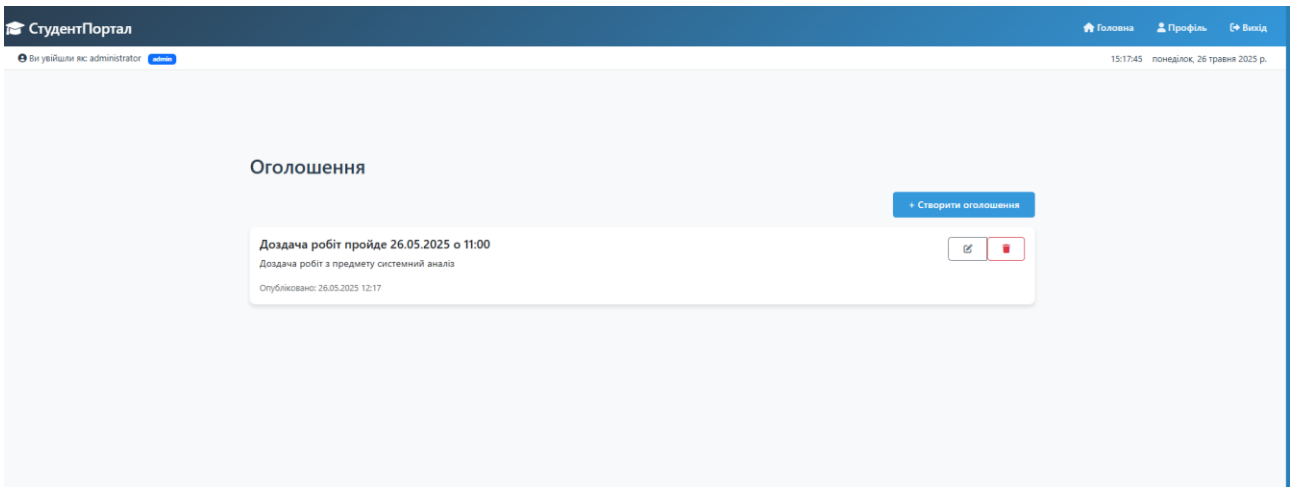

Рисунок 3.5 – Інтерфейс модуля оголошень із опублікованим оголошенням

Модуль інтегрований з системою сповіщень для автоматичного інформування користувачів про нові оголошення. Безпека та модерація забезпечуються системою прав доступу, яка визначає, хто може створювати, редагувати та видаляти оголошення, а також системою модерації контенту перед публікацією.

Реалізація модуля розкладу заходів

Модуль розкладу заходів є важливою складовою веб-сайту студентської спільноти КІФК КНУБА, що забезпечує централізоване управління інформацією про різноманітні події, які проводяться в коледжі. Цей модуль надає зручний інструмент для планування, анонсування та координації культурно-масових, спортивних, навчальних та інших заходів.

Архітектура модуля базується на паттерні MVC. Модель представлена класом Event та зв'язаними моделями EventParticipation та EventCategory. Контролер EventController містить методи для основних операцій з заходами. Представлення реалізовані за допомогою Blade-шаблонів та бібліотеки FullCalendar для інтерактивного відображення календаря (Рис. 3.6-3.8).

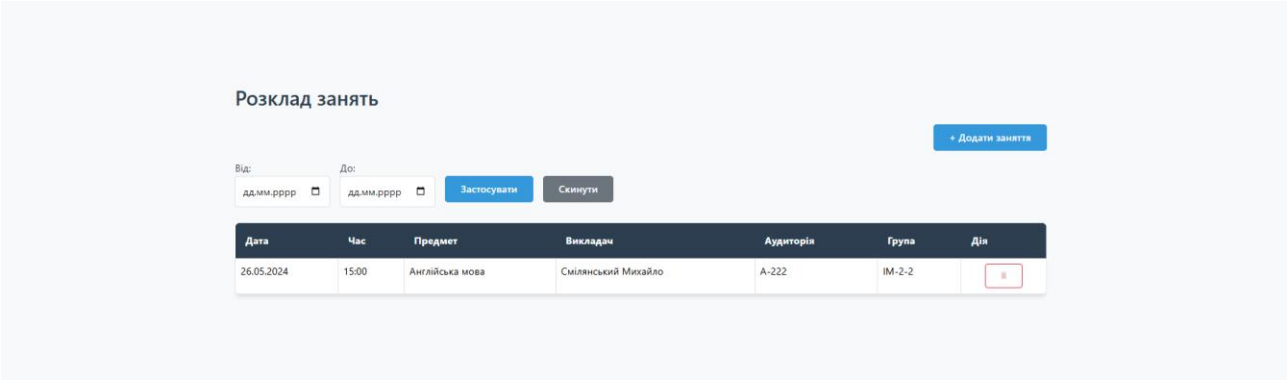

Рисунок 3.6 – Інтерфейс модуля розкладу занять із формою створення та таблицею занять

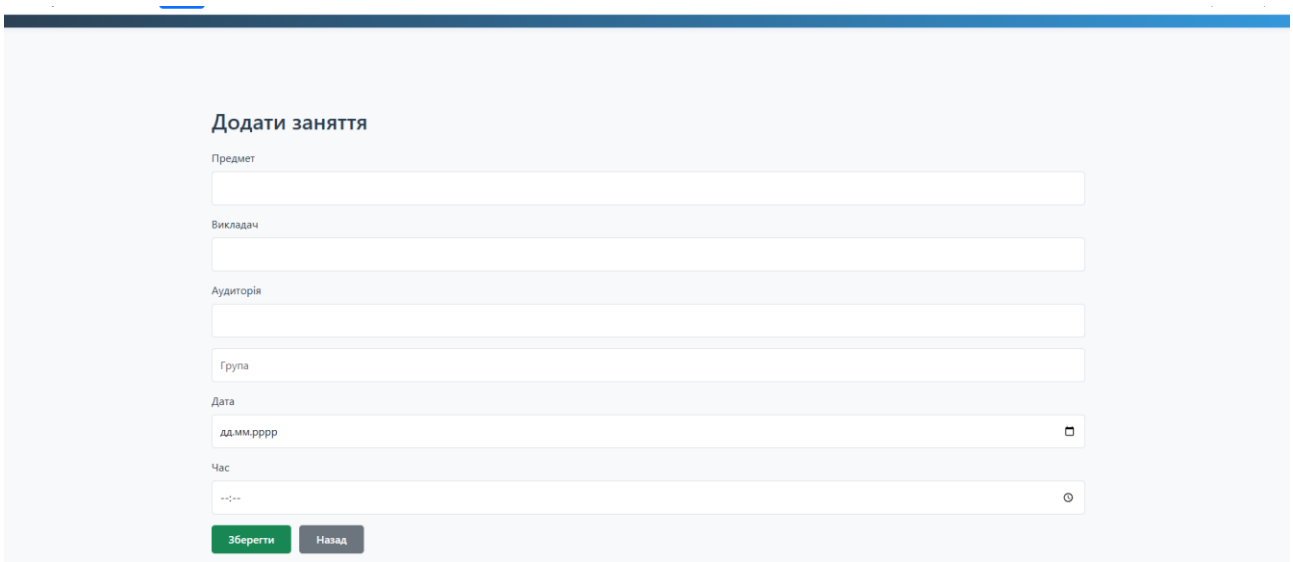

Рисунок 3.7 – Форма для додавання нового заняття в модулі розкладу заходів

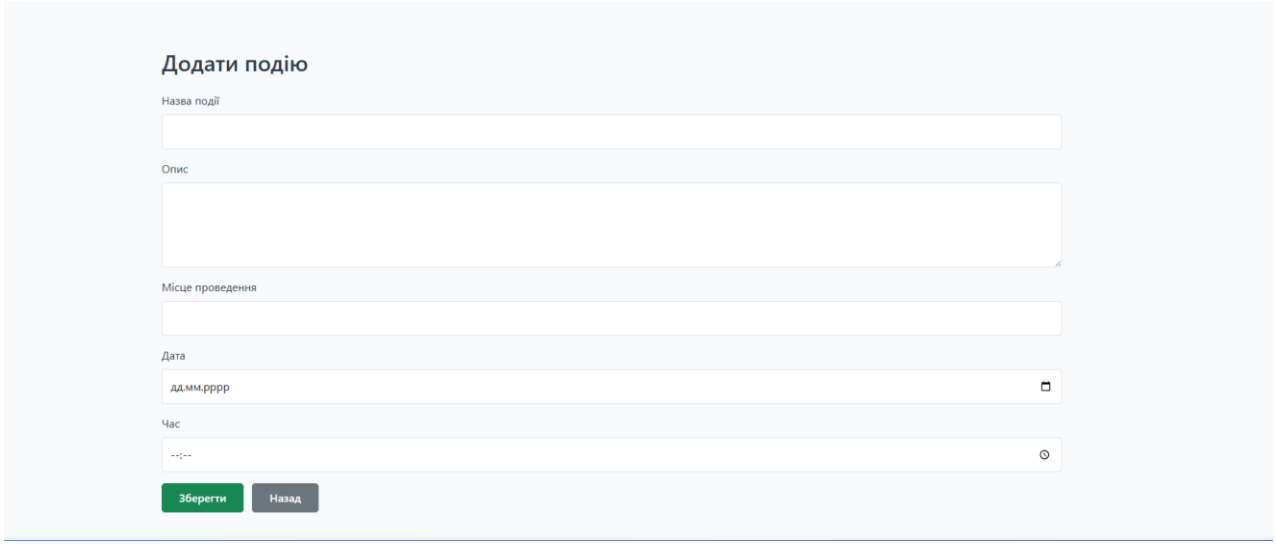

Рисунок 3.8 – Форма для додавання нового поля в модулі подій

Центральним компонентом модуля є інтерактивний календар з різними представленнями (день, тиждень, місяць, список) та можливістю фільтрації заходів за категоріями. Функціонал створення та управління заходами дозволяє авторизованим користувачам з відповідними правами створювати нові заходи з усіма необхідними параметрами, включаючи повторювані події (Рис.3.9).

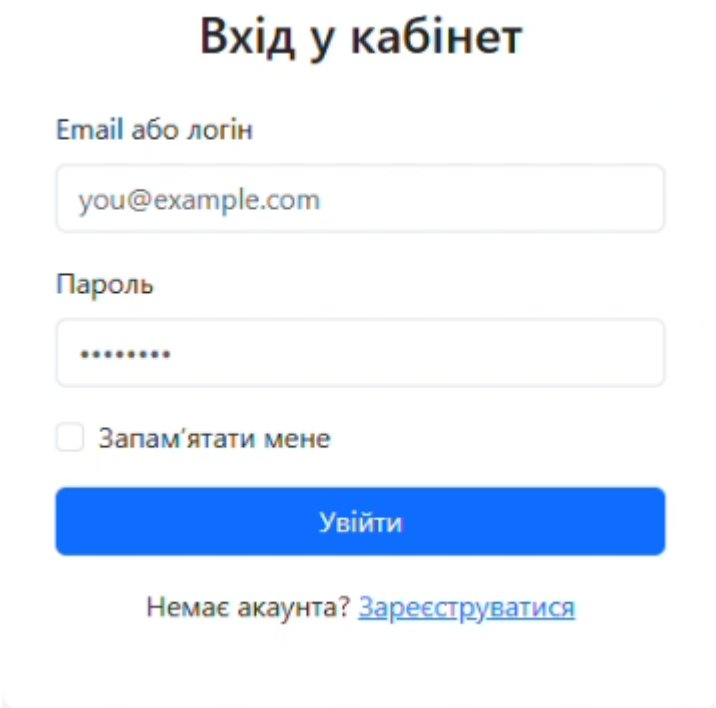

Рисунок 3.9 – Форма створення спортивного заходу в модулі розкладу заходів

Для заходів з обмеженою кількістю учасників реалізовано систему онлайн-реєстрації. Модуль інтегрований з системою сповіщень для інформування користувачів про нові заходи, зміни в розкладі та нагадування про майбутні події, а також підтримує експорт у популярні формати календарів.

Реалізація модуля коментування

Модуль коментування є важливою складовою веб-сайту студентської спільноти КІФК КНУБА, що забезпечує інтерактивну взаємодію між користувачами системи та сприяє обговоренню опублікованої інформації. Цей модуль інтегрований з іншими компонентами системи, дозволяючи користувачам залишати коментарі до різних типів публікацій.

Архітектура модуля коментування базується на поліморфному зв'язку, що дозволяє асоціювати коментарі з різними типами об'єктів системи. Модель

Comment містить необхідні атрибути та використовує єдину таблицю для зберігання коментарів до різних типів контенту, забезпечуючи гнучкість та ефективність роботи з даними.

Функціонал додавання коментарів реалізований через AJAX-запити, що дозволяє користувачам коментувати без перезавантаження сторінки. Для зручного перегляду коментарі відображаються у вигляді сортованого списку з інформацією про автора та можливістю згортання/розгортання великих текстів, а також реалізовано можливість відповідей на коментарі (Рис. 3.10).

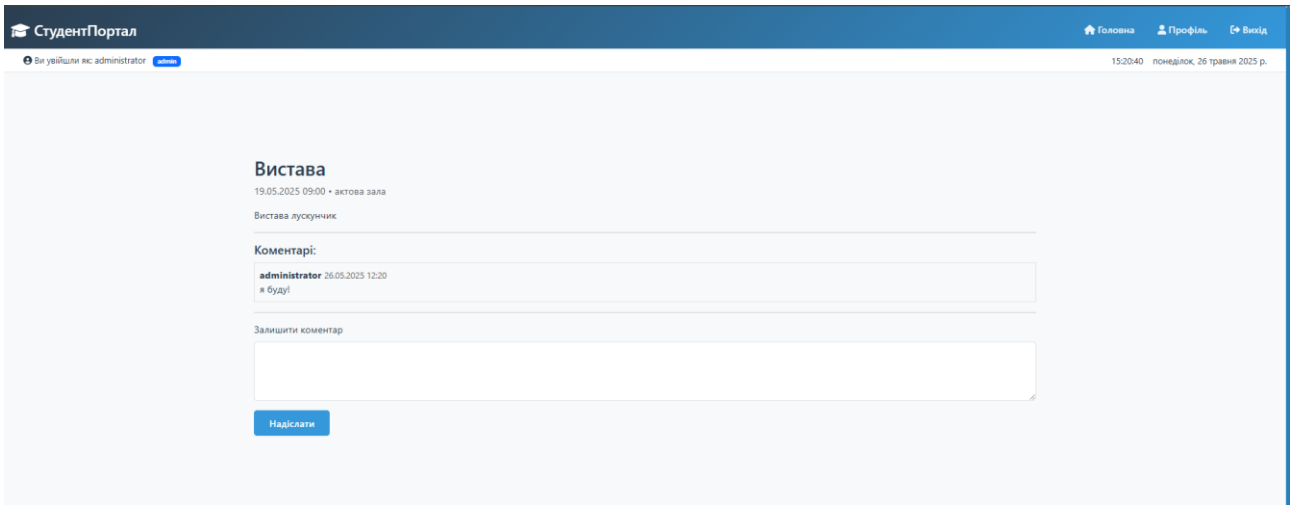

Рисунок 3.10 – Інтерфейс модуля коментарів для додавання нового коментаря

Модерація коментарів забезпечується системою прав доступу та функцією повідомлення про неприйнятний контент. Модуль інтегрований з системою сповіщень, яка інформує користувачів про нові коментарі до їхніх публікацій або відповіді на їхні коментарі, з можливістю налаштування отримання таких сповіщень.

Реалізація модуля анонімних повідомлень та скарг

Модуль анонімних повідомлень та скарг є важливим інструментом для забезпечення зворотного зв'язку між студентами та адміністрацією коледжу, дозволяючи користувачам безпечно висловлювати свої зауваження, скарги та пропозиції без розкриття своєї особистості. Цей модуль створений з метою підвищення рівня довіри та прозорості процесів управління (Рис. 3.11).

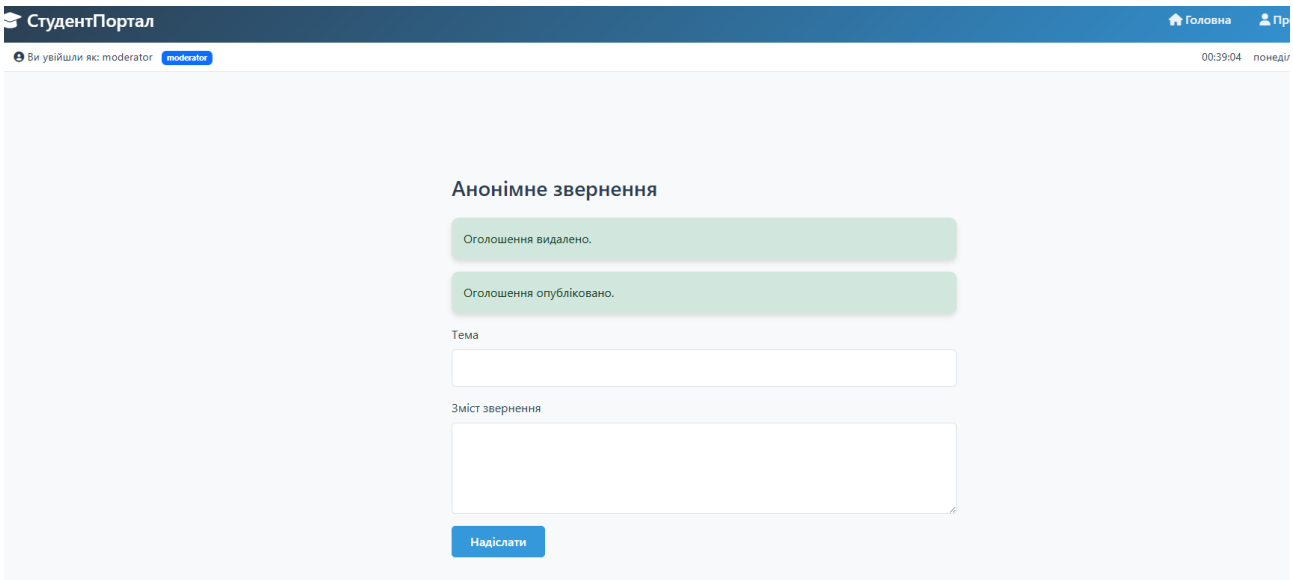

Рисунок 3.11 – Список коментарів до запису в модулі коментарів

Архітектура модуля базується на моделі AnonymousMessage з атрибутами для теми, тексту, категорії, статусу та унікального токена для анонімної комунікації. Ключовою особливістю є відсутність прямого зв'язку з моделлю користувача, що забезпечує анонімність відправника. Контролер AnonymousMessageController містить методи для всіх основних операцій з повідомленнями.

Функціонал створення анонімних повідомлень доступний навіть неавторизованим користувачам. Для запобігання зловживанням реалізовано обмеження на кількість повідомлень з однієї IP-адреси, використання reCAPTCHA та фільтрацію за ключовими словами. Система генерує унікальний токен для перевірки статусу повідомлення та отримання відповіді.

Адміністративний інтерфейс модуля надає доступ до перегляду та управління анонімними повідомленнями, дозволяючи змінювати статус повідомлень та надавати відповіді без розкриття особистості відправника. Особлива увага приділена безпеці та конфіденційності даних, включаючи захищене з'єднання та автоматичне видалення повідомлень після певного періоду часу.

Реалізація модуля загублених/знайдених речей

Модуль загублених/знайдених речей є практичним компонентом веб-сайту студентської спільноти КІФК КНУБА, що дозволяє ефективно вирішувати

проблему пошуку загублених предметів та повернення знайдених речей їхнім власникам. Цей модуль надає централізовану платформу для розміщення відповідних оголошень.

Архітектура модуля базується на моделі LostFoundItem з атрибутами для опису предмета, включаючи тип оголошення, назву, опис, категорію, місце, дату, контактну інформацію та шлях до фотографії. Модель пов'язана з моделлю User для ідентифікації автора оголошення. Контролер LostFoundItemController обробляє всі запити для створення, редагування, видалення оголошень та пошуку серед них.

Функціонал створення нових оголошень дозволяє авторизованим користувачам розміщувати інформацію про загублені або знайдені предмети з детальним описом та можливістю завантаження фотографії. Для ефективного пошуку реалізовано функціонал фільтрації та пошуку оголошень за різними параметрами, включаючи повнотекстовий пошук.

Система автоматичних сповіщень інформує користувачів про нові оголошення, що відповідають їхнім інтересам, на основі підписок за певними параметрами. Для забезпечення актуальності інформації реалізовано функціонал зміни статусу та автоматичної архівації оголошень після певного періоду часу.

Розробка інтерфейсу користувача

Розробка інтерфейсу користувача для веб-сайту студентської спільноти КІФК КНУБА була спрямована на створення зручного, інтуїтивно зрозумілого та естетично привабливого середовища для взаємодії користувачів з системою. При проєктуванні враховувалися принципи user-centered design та сучасні тенденції веб-дизайну.

Основою інтерфейсу є адаптивний дизайн, реалізований з використанням фреймворку Bootstrap 5, який забезпечує коректне відображення веб-сайту на різних пристроях. Верстка сторінок виконана за принципом "mobile-first" для оптимального користувацького досвіду на всіх типах пристроїв. Візуальний стиль розроблений відповідно до фірмового стилю коледжу з використанням корпоративних кольорів, логотипу та шрифтів.

Структура інтерфейсу включає навігаційну панель у верхній частині сторінки, основний контент у центральній частині та додаткову інформацію у бічній панелі або під основним контентом. Для кожного модуля системи розроблено спеціалізований інтерфейс, який враховує специфіку взаємодії користувача з відповідним функціоналом.

Особлива увага була приділена інтерактивним елементам та зворотному зв'язку для користувача. Реалізовано систему повідомлень про результати дій, використано анімації для плавної зміни станів інтерфейсу, забезпечено валідацію форм з інформативними повідомленнями про помилки та автозбереження даних для запобігання їх втраті.

3.4.Інструкція користувача

Реєстрація та вхід

Процеси реєстрації та входу в систему є першими етапами взаємодії користувача з веб-сайтом студентської спільноти КІФК КНУБА. Особлива увага була приділена зручності, безпеці та інтуїтивності цих процесів для забезпечення надійної ідентифікації користувачів та захисту їхніх облікових даних.

Реєстрація нових користувачів здійснюється через спеціальну форму, що містить поля для основної інформації. Після заповнення форми система перевіряє унікальність електронної пошти та створює новий обліковий запис з зашифрованим паролем (Рис.3.12).

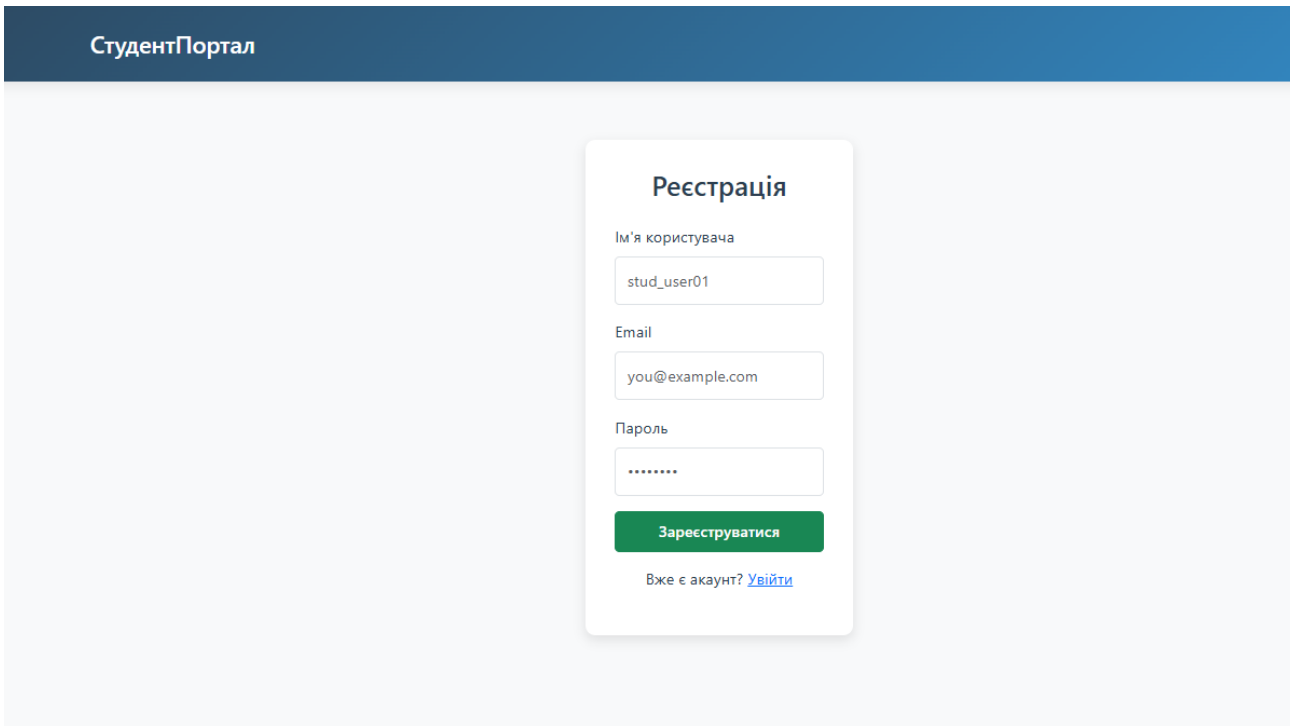

Рисунок 3.12 – Форма реєстрації нового користувача в системі аутентифікації

Вхід в систему здійснюється через форму з полями для електронної пошти та паролю, а також опцією "Запам'ятати мене".

Робота з основними модулями

Після авторизації користувач потрапляє до головного меню, отримує доступ до основних модулів системи: оголошень, розкладу заходів, модуля загублених/знайдених речей, анонімних повідомлень та інших компонентів (Рис. 3.13). Взаємодія з цими модулями організована для забезпечення інтуїтивності, зручності та ефективності виконання типових задач.

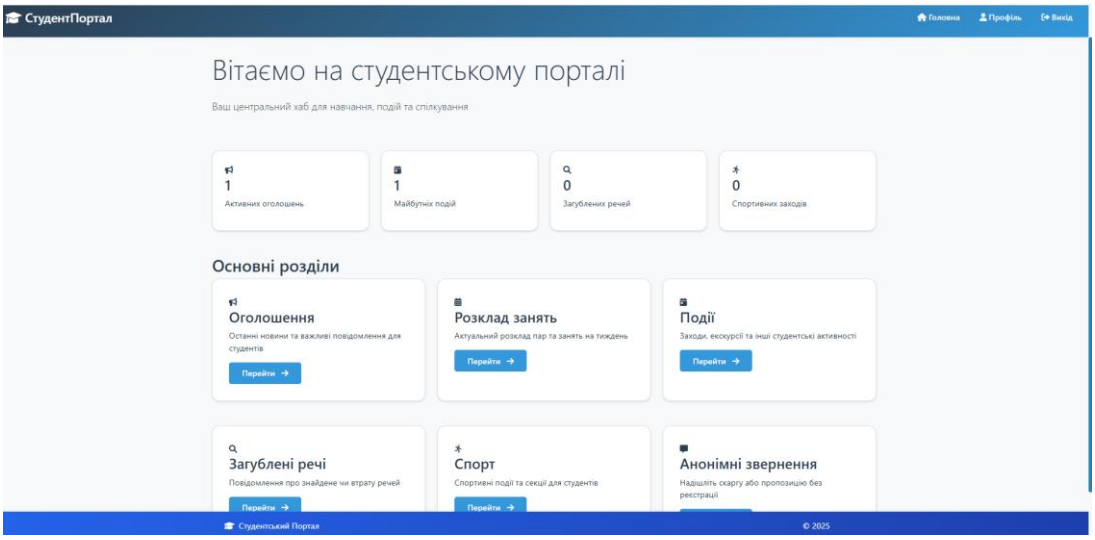

Рисунок 3.13 – Головне меню веб-сайту

Робота з модулем оголошень починається з перегляду списку актуальних оголошень (Рис. 3.14). Користувачі з відповідними правами можуть створювати нові оголошення, заповнюючи форму з необхідними полями (Рис. 3.15).

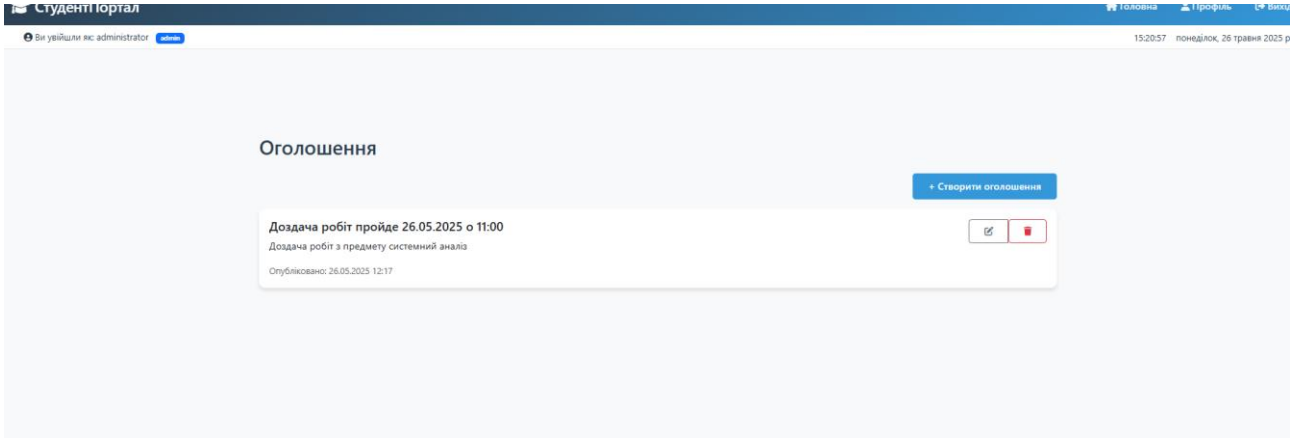

Рисунок 3.14 – Модуль оголошення

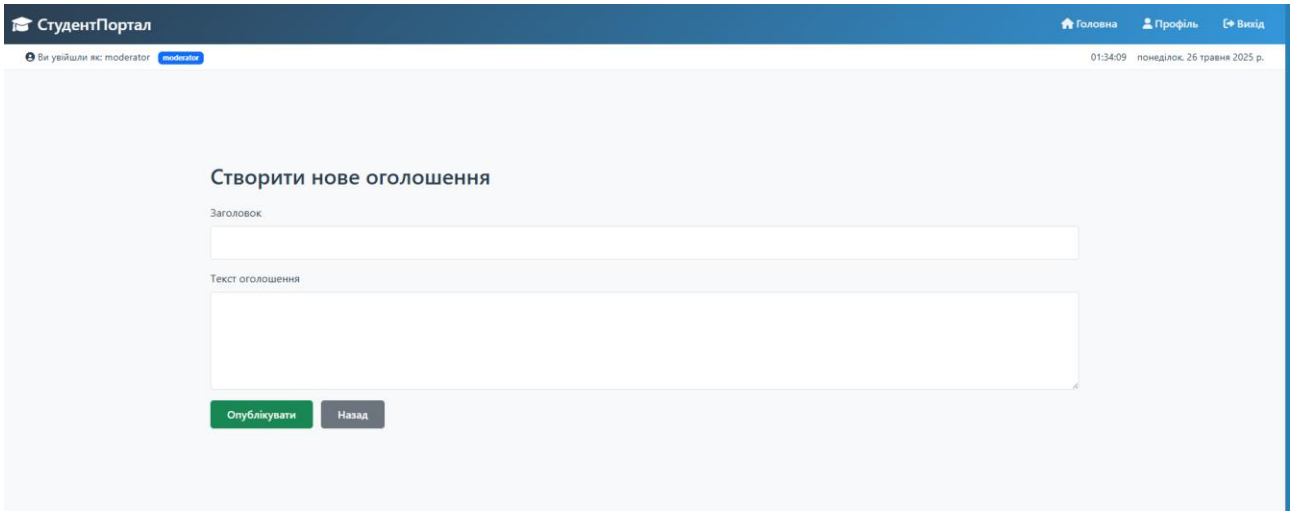

Рисунок 3.15 – Сторінка зі створенням нових оголошень

Взаємодія з модулем розкладу заходів відбувається через інтерактивний календар з можливістю перемикання між різними представленнями та фільтрації за категоріями (Рис. 3.16). Користувачі з відповідними правами можуть редагувати розклад (Рис. 3.17).

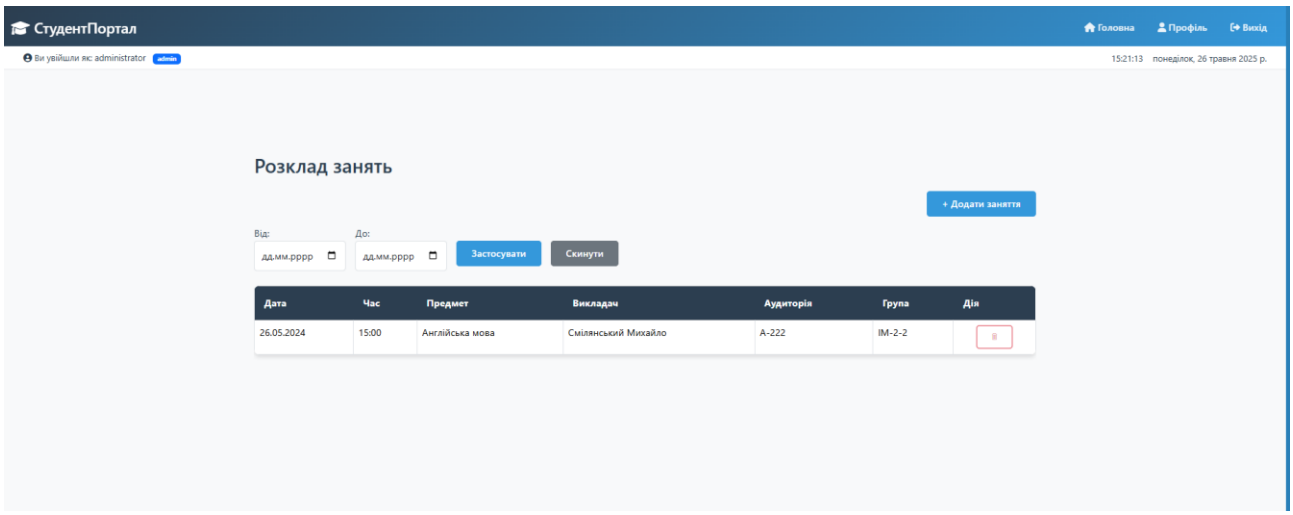

Рисунок 3.16 – Модуль розкладу занять

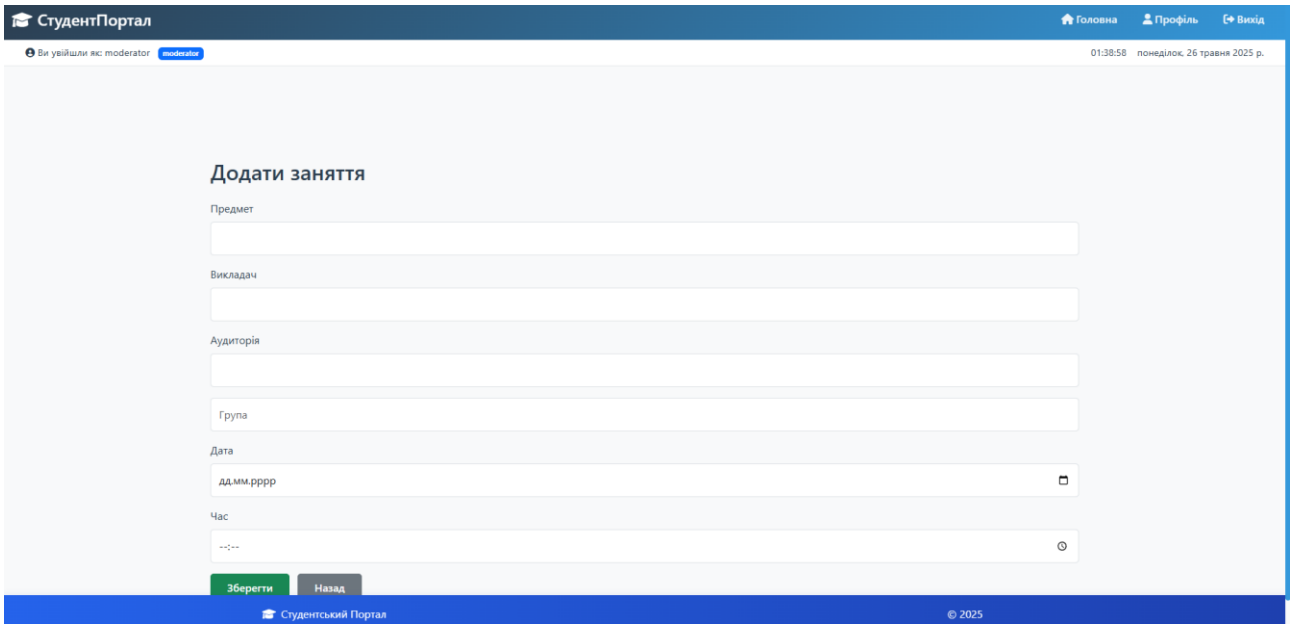

Рисунок 3.17 – Створення нових занять для розкладу

При натисканні на події/заходи відкривається детальна інформація з можливістю передивлятись деталі та коментувати (Рис. 3.18). Користувачі з відповідними правами можуть створювати нові заходи з різними параметрами (Рис. 3.19).

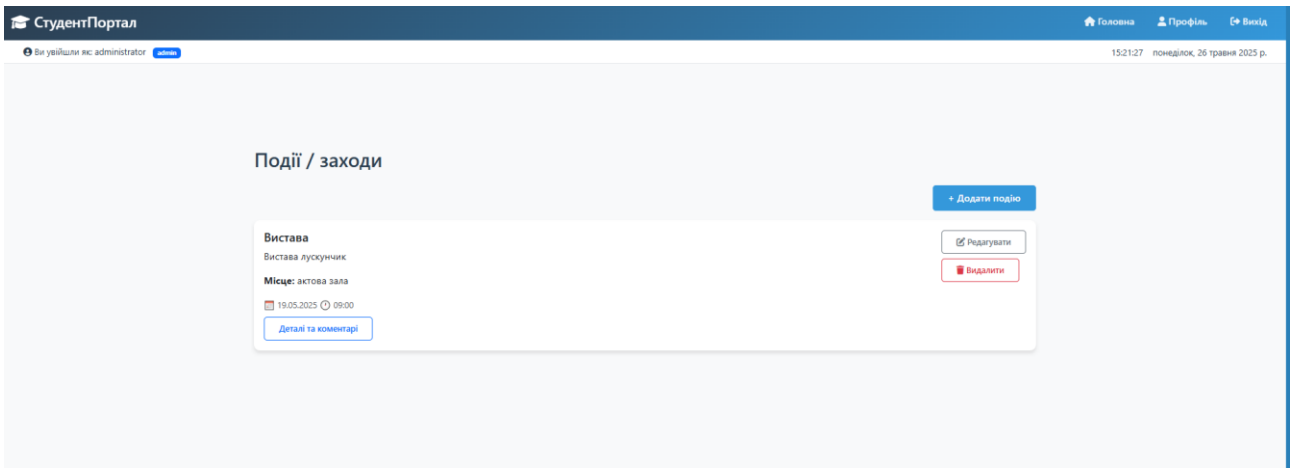

Рисунок 3.18 – Модуль події/заходи

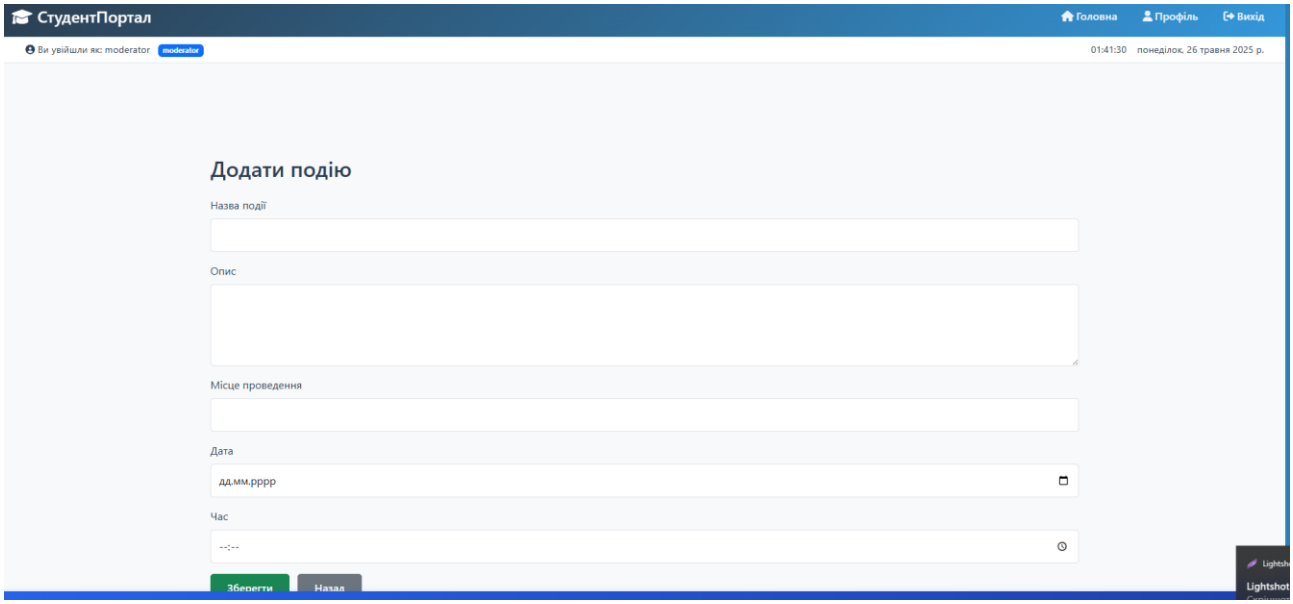

Рисунок 3.19 – Додавання нових подій до модулю

Для роботи з модулем загублених/знайдених речей користувач переглядає список оголошень та може додати річ в разі необхідності (Рис. 3.20).

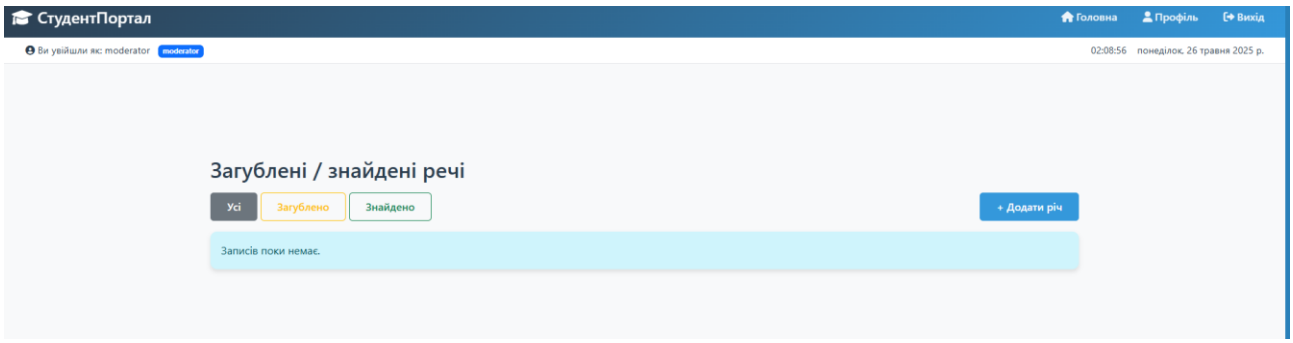

Рисунок 3.20 – Модуль загублених/знайдених речей

Для відправки анонімного повідомлення користувач заповнює форму з темою, текстом та категорією без необхідності авторизації, що забезпечує повну анонімність (Рис. 3.21).

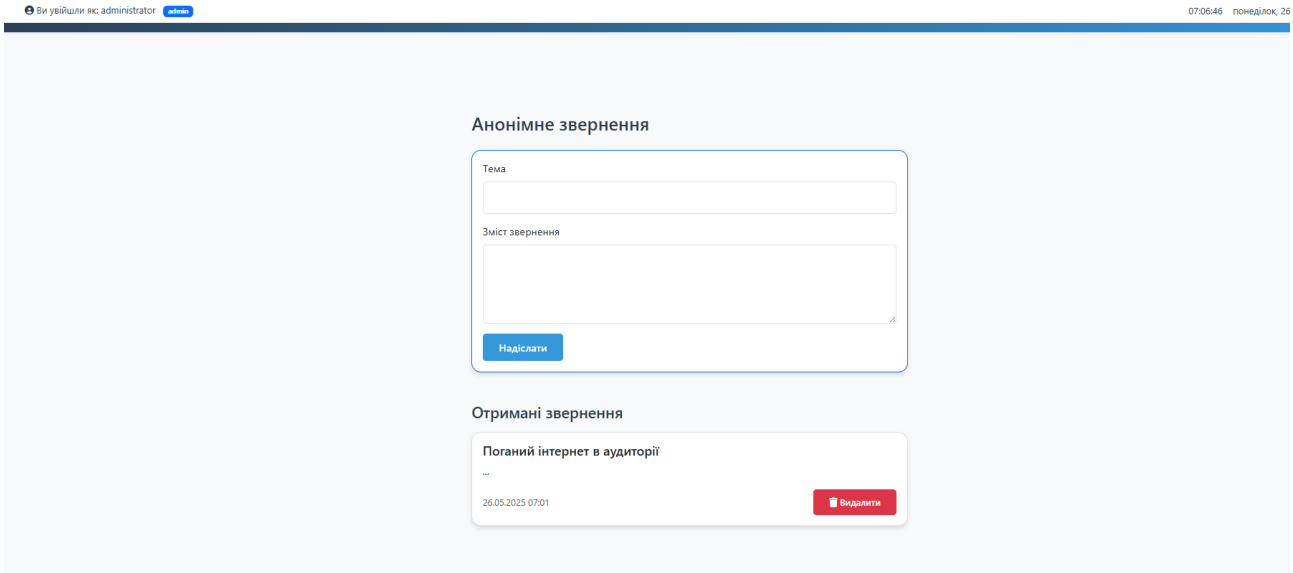

Рисунок 3.21 – Модуль анонімних звернень

В модулі спортивних подій та екскурсій користувач може переглядати актуальні події, переглядати деталі. Користувачі що мають права модераторів можуть редагувати, видаляти та додавати нові події (Рис. 3.22).

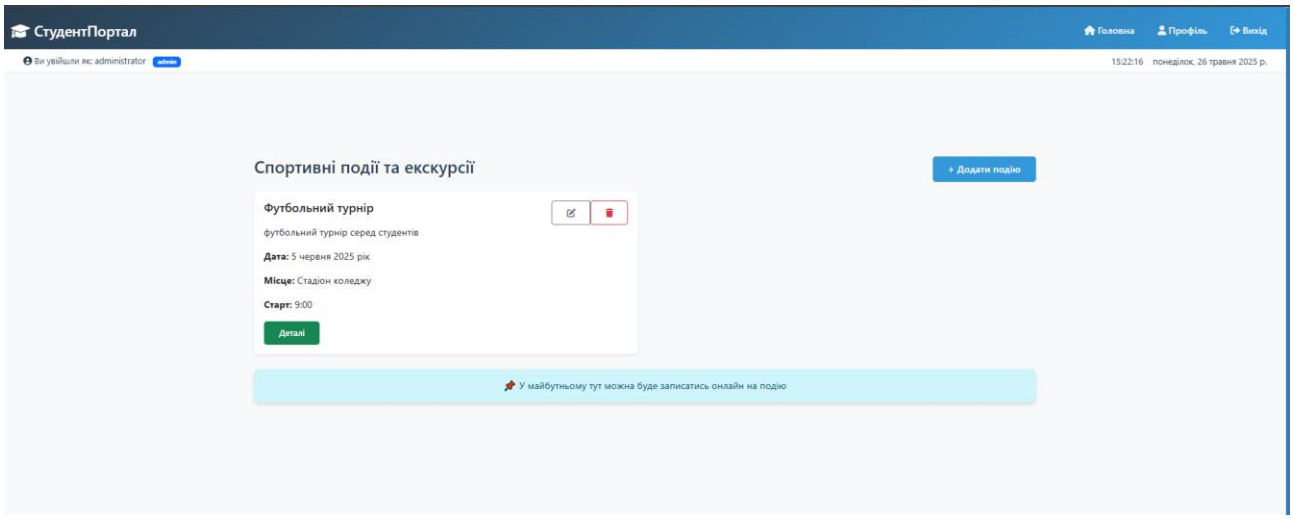

Рисунок 3.22 – Модуль спортивних подій та екскурсій

Керівництво для модераторів

Модератори веб-сайту студентської спільноти КІФК КНУБА мають розширений функціонал для управління контентом та користувачами системи.

Модерація контенту забезпечує підтримку якості інформації через можливість перегляду, редагування, приховування або видалення різних типів контенту.

Робота з анонімними повідомленнями дозволяє уповноваженим особам переглядати та обробляти повідомлення з можливістю зміни статусу та надання відповіді, зберігаючи анонімність відправника. Налаштування системи дають можливість конфігурувати різні аспекти роботи без необхідності внесення змін у код, включаючи параметри безпеки, сповіщень, модерації та інтерфейсу.

3.5. Тестування програмного продукту.

План тестування.

План тестування веб-сайту студентської спільноти КІФК КНУБА був розроблений для забезпечення всебічної перевірки функціональності, продуктивності, безпеки та зручності використання системи перед її введенням в експлуатацію. План включає різні рівні та типи тестування, розподілені за етапами розробки.

Модульне тестування [10] (Unit Testing) проводилося для перевірки окремих компонентів системи в ізоляції від інших частин. Були написані автоматизовані тести для всіх основних моделей та контролерів з використанням фреймворку PHPUnit. Тести перевіряли коректність базових операцій (CRUD) для кожної моделі, валідацію вхідних даних, обробку граничних випадків та винятків. Модульне тестування дозволило виявити та виправити помилки на ранніх етапах розробки.

Інтеграційне тестування перевіряло взаємодію між різними компонентами системи, зокрема між фронтендом та бекендом, між різними модулями системи та з зовнішніми сервісами. Були протестовані сценарії реєстрації та авторизації, створення та редагування різних типів контенту, пошук та фільтрація, відправка сповіщень та інші основні функції. Особлива увага приділялася перевірці цілісності даних при виконанні складних операцій, що зачіпають кілька таблиць бази даних.

Функціональне тестування перевіряло відповідність системи функціональним вимогам, зазначеним у технічному завданні. Були розроблені тест-кейси для кожного модуля системи, які охоплювали всі основні сценарії використання. Тестування проводилося як розробниками, так і представниками замовника для забезпечення об'єктивної оцінки. Результати функціонального тестування документувалися у вигляді звітів з переліком виявлених помилок та невідповідностей вимогам.

Інші типи тестування включали тестування безпеки для виявлення вразливостей, навантажувальне тестування для перевірки продуктивності системи під навантаженням, тестування інтерфейсу користувача для перевірки зручності використання та адаптивності дизайну, а також кросбраузерне тестування для забезпечення коректної роботи системи в різних браузерах та на різних пристроях.

Результати тестування.

Результати тестування веб-сайту студентської спільноти КІФК КНУБА були систематизовані та проаналізовані для оцінки готовності системи до впровадження та визначення напрямків подальшого вдосконалення. Загалом тестування показало високий рівень якості розробки, але виявило ряд проблем, які потребували виправлення.

Модульне тестування виявило кілька помилок в логіці роботи моделей, зокрема проблеми з валідацією даних, неправильну обробку деяких граничних випадків та помилки в методах для роботи з пов'язаними моделями. Всі виявлені проблеми були оперативно виправлені та повторно протестовані для підтвердження коректності виправлень. Загальний рівень проходження модульних тестів після виправлень склав 98%, що є хорошим показником для проєкту такого масштабу.

Інтеграційне тестування виявило проблеми з взаємодією між різними модулями системи, зокрема неправильну передачу даних між формами та API-ендпоінтами, помилки в обробці AJAX-запитів та проблеми з кешуванням даних. Також були виявлені проблеми з інтеграцією з зовнішніми сервісами, зокрема з

системою відправки електронних листів та API для перевірки reCAPTCHA. Більшість цих проблем були вирішені шляхом доопрацювання коду та конфігурації системи.

Функціональне тестування показало високий рівень відповідності системи функціональним вимогам. Більшість тест-кейсів були успішно пройдені, але виявлено кілька невідповідностей очікуванням користувачів, зокрема у модулі розкладу заходів та системі фільтрації оголошень. Ці проблеми були обговорені з замовником та вирішені шляхом внесення змін у логіку роботи відповідних модулів.

Тестування безпеки виявило кілька вразливостей, включаючи можливість XSS-атак через недостатню фільтрацію користувацького контенту, вразливості CSRF в деяких формах та недостатній захист від атак методом підбору паролів. Навантажувальне тестування показало задовільну продуктивність системи при середньому навантаженні, але виявило проблеми з масштабуванням при значному збільшенні кількості користувачів та обсягу даних. Тестування інтерфейсу користувача виявило проблеми з адаптивністю дизайну на деяких мобільних пристроях та недостатню інтуїтивність деяких елементів управління.

Виправлення помилок та оптимізація.

На основі результатів тестування було розроблено план виправлення виявлених помилок та оптимізації системи. План включав пріоритизацію завдань за критичністю та складністю, розподіл ресурсів та встановлення термінів виконання. Найбільш критичні проблеми, особливо пов'язані з безпекою та цілісністю даних, були вирішені в першу чергу.

Для виправлення проблем безпеки було впроваджено додаткову фільтрацію користувацького контенту з використанням бібліотеки HTML Purifier, посилено механізми захисту від підбору паролів через впровадження експоненціального збільшення часу блокування після невдалих спроб входу. Також було проведено аудит коду для виявлення та виправлення потенційних вразливостей SQL-ін'єкцій та інших типів атак.

Для оптимізації продуктивності системи було впроваджено кешування результатів частих запитів до бази даних, оптимізовано SQL-запити для зменшення навантаження на сервер, додано індекси для полів, за якими часто виконується пошук та сортування. Впроваджено відкладене завантаження зображень для прискорення початкового рендерингу сторінок

Для покращення користувацького інтерфейсу були внесені зміни в адаптивну верстку для забезпечення коректного відображення на всіх типах пристроїв, покращено навігацію для мобільних користувачів, оптимізовано форми для більш зручного введення даних, додано більш інформативні повідомлення про помилки та підказки для користувачів. Також було проведено юзабіліті-тестування зі студентами та викладачами коледжу для виявлення та усунення проблем з інтуїтивністю інтерфейсу.

Після завершення всіх виправлень та оптимізацій було проведено повторне тестування системи для підтвердження ефективності внесених змін. Результати показали значне покращення в усіх аспектах роботи системи: підвищення рівня безпеки, покращення продуктивності при високих навантаженнях, більш зручний та інтуїтивний інтерфейс користувача. Всі критичні та більшість некритичних проблем були успішно вирішені, що дозволило перейти до етапу впровадження системи в експлуатацію з високим рівнем впевненості в її надійності та відповідності вимогам.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було створено повноцінний веб-сайт для студентської спільноти Київського індустріального фахового коледжу КНУБА, який вирішує виявлені проблеми комунікації та забезпечує ефективний інформаційний обмін між усіма учасниками освітнього процесу. Проєкт пройшов усі етапи розробки – від аналізу предметної області до тестування та оптимізації системи, і готовий до впровадження в експлуатацію.

Аналіз існуючого стану інформаційного обміну в коледжі виявив суттєві проблеми, зокрема розрізненість інформаційних каналів, несвоєчасність поширення інформації, відсутність системи зворотного зв'язку та неефективність обліку загублених та знайдених речей. Проведене дослідження аналогічних систем дозволило визначити оптимальні підходи до вирішення цих проблем та розробити ефективну архітектуру веб-сайту.

Розроблена система складається з кількох ключових модулів, які забезпечують комплексне вирішення поставлених задач:

1. Модуль аутентифікації та авторизації забезпечує безпечну ідентифікацію користувачів та розмежування доступу до функцій системи залежно від ролі користувача.
2. Модуль оголошень надає зручний інструмент для централізованого розміщення та управління інформаційними повідомленнями різних категорій.
3. Модуль розкладу заходів забезпечує ефективне планування, анонсування та координацію культурно-масових, спортивних та інших подій.
4. Модуль коментування дозволяє організувати інтерактивне обговорення опублікованої інформації.
5. Модуль анонімних повідомлень та скарг створює безпечний канал зворотного зв'язку між студентами та адміністрацією.
6. Модуль загублених/знайдених речей вирішує практичну проблему обліку та повернення загублених предметів.

Всі модулі інтегровані в єдину систему з уніфікованим інтерфейсом користувача, що забезпечує зручність та інтуїтивність використання. Інтерфейс розроблено з використанням адаптивного дизайну, що дозволяє ефективно використовувати систему на різних пристроях – від настільних комп'ютерів до мобільних телефонів.

Проведене тестування системи виявило та дозволило усунути ряд проблем, пов'язаних з безпекою, продуктивністю та зручністю використання. Внесені оптимізації забезпечили високу надійність, швидкодію та масштабованість системи, що дозволяє їй ефективно функціонувати при зростанні кількості користувачів та обсягу даних.

Розроблений веб-сайт має значний потенціал для подальшого розвитку та вдосконалення. Перспективними напрямками є інтеграція з системою електронного навчання коледжу, розробка мобільного додатку для ще більш зручного доступу до інформації, розширення функціоналу модуля розкладу заходів з додаванням можливості синхронізації з популярними календарями, а також впровадження елементів гейміфікації для підвищення залученості студентів.

Економічний аналіз проєкту показав його високу рентабельність з прийнятним періодом окупності інвестицій. Впровадження розробленої системи дозволить коледжу не лише підвищити ефективність комунікації, але й отримати суттєві економічні вигоди через оптимізацію адміністративних процесів, зменшення витрат на друк інформаційних матеріалів та підвищення ефективності використання ресурсів.

Таким чином, розроблений веб-сайт для студентської спільноти КІФК КНУБА є повноцінним, сучасним та ефективним інструментом для вирішення комунікаційних проблем в коледжі. Його впровадження дозволить створити єдиний інформаційний простір, підвищити рівень інформованості студентів про події та можливості самореалізації, забезпечити зворотний зв'язок між студентами та адміністрацією, та загалом підвищити якість освітнього середовища в коледжі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Беррі, П. Head First. Python : пер. з англ. / П. Беррі. – 2-ге вид. – Харків : Фабула, 2023. – 624 с.
2. Київський індустріальний фаховий коледж КНУБА. URL: https://www.kifk.org (дата звернення: 18.05.2025).
3. Маргітич А. ОРГАНІЗАЦІЙНА КУЛЬТУРА КОЛЕДЖУ: ЦІННОСТІ ТА НОРМИ ДЛЯ СТУДЕНТІВ І ВИКЛАДАЧІВ. 13.05.2021. URL: http://dspaces.msu.edu.ua:8080/bitstream/123456789/8474/1/Organizational_culture_of_the_college_values_and_norms_for_students_and_teachers.pdf (дата звернення: 01.06.2025).
4. Сичева, Світлана Володимирівна. Формування цифрової компетентності студентів педагогічних коледжів у процесі фахової підготовки. 2021 URL: https://dspace.znu.edu.ua/xmlui/handle/12345/5663 (дата звернення: 18.05.2025).
5. Крепич, С. Я. Якість програмного забезпечення та тестування : базовий курс [Електронний ресурс] : навч. посіб. / С. Я. Крепич, І. Я. Співак. - Тернопіль : ФОП Паляниця В. А., 2020. - 479 с.
6. Flask’s documentation. URL: https://flask.palletsprojects.com/en/stable/ (дата звернення: 18.05.2025).
7. *SQLite documentation*. URL: https://www.sqlite.org/docs.html (дата звернення: 18.05.2025).
8. Flask’s documentation. URL: https://flask.palletsprojects.com/en/stable/ (дата звернення: 18.05.2025).
9. Логічна модель БД. URL: https://data-life-ua.com/db/lohichna-model-danykh/ (дата звернення: 18.05.2025).
10. *Модульне тестування*. URL: https://qalight.ua/baza-znaniy/modulne-testuvannya/ (дата звернення: 18.05.2025).

11. Нікольський, Ю. В. Системи штучного інтелекту : навч. посібник / Ю. В. Нікольський, В. В. Пасічник, Ю. М. Щербина ; за наук. ред. В. В. Пасічника. – 3-тє вид., випр. та доп. – Львів : Магнолія 2006, 2024. – 279 с.

12. Пасічник, В. В. Веб-дизайн : підручник / В. В. Пасічник, О. В. Пасічник ; за заг. ред. В. В. Пасічника. – Львів : Магнолія 2006, 2024. – 520 с.

13. Пасічник, В. В. Веб-технології : підручник / В. В. Пасічник, О. В. Пасічник, Д. І. Угрин. – Львів : Магнолія 2006, 2024. – 336 с

14. Пастернек, І. І. Інструментальні засоби веб-технологій : навч. посібник / І. І. Пастернек, А. Т. Костик ; Національний університет "Львівська політехніка". – Львів : Магнолія 2006, 2024. – 198 с.

15. Хомуляк, М. О. Адміністрування комп'ютерних систем і мереж : навч. посібник / М. О. Хомуляк ; Національний університет "Львівська політехніка". – Львів : Магнолія 2006, 2024. – 153 с.

16. Шаховська, Н. Б. Алгоритми та структури даних : навч. посібник / Н. Б. Шаховська, Р. О. Голощук ; Національний університет "Львівська політехніка". – Львів : Магнолія 2006, 2024. – 215 с.

17. Затковський В., Іванчук Я. Система автоматичного тестування веб-додатків. Система автоматичного тестування веб-додатків. 2021. URL: https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/40194/15680.pdf?sequence=3&isAllowed=y

18. Галісеєв, Г. В. Системне програмування : навч. посібник / Г. В. Галісеєв. – Київ : Ун-т "Україна", 2019. – 113 с

19. Буйницька, О. П. Інформаційні технології та технічні засоби навчання : навч. посібник / О. П. Буйницька ; Київ. ун-т ім. Б. Грінченка. – Київ : ЦУЛ, 2022. – 240 с.

20. Теоретичні та практичні аспекти розвитку Інтернет-економіки : навч. посібник / О. О. Шульга, О. І. Вівчар, М. Ф. Зайлик [та ін.] ; за наук. ред. : І. Л. Татомир, Л. Г. Квасній ; Прикарпат. ін-т ім. М. Грушевського ПрАТ ВНЗ "МАУП". – Дрогобич-Трускавець : Просвіт, 2021. – 386 с

21. ОПП ШІ: Методичні рекомендації до виконання кваліфікаційної роботи на здобуття освітнього ступеня «бакалавр» спеціальності 122 «Комп’ютерні науки» освітньо-професійної програми «Інформаційні системи та штучний інтелект» денної форми здобуття освіти [Електрон. ресурс] / уклад. С. В. Грибков, Н. В. Ліманська, М. П. Костіков. – К.: НУХТ, 2025. – 43 с.

22. ОПП КН: Методичні рекомендації до виконання кваліфікаційної роботи на здобуття освітнього ступеня «бакалавр» спеціальності 122 «Комп’ютерні науки» освітньо-професійної програми «Комп’ютерні науки» денної та заочної форми здобуття освіти [Електрон. ресурс] / уклад. С. В. Грибков, Н. В. Ліманська, М. П. Костіков. – К.: НУХТ, 2025. – 43 с.

23. Сукало, М. Л. Інформаційна технологія керування автоматизованим адаптивним моніторингом обмежених екосистем із застосуванням хмарних технологій та Інтернету речей : дис. ... д-ра філос. : 151 "Автоматизація та комп'ютерно-інтегровані технології" / Сукало Максим Леонідович ; Нац. ун-т харч. технол. – Київ, 2024. – 229 с

24. Інформаційні технології та моделювання бізнес-процесів : навч. посібник / О. М. Томашевський, Г. Г. Цегелик, М. Б. Вітер, В. І. Дубук. – Київ : ЦУЛ, 2023. – 296 с.

25. Світличний, О. О. Основи геоінформатики : навч. посібник / О. О. Світличний, С. В. Плоницький. – Суми : Університетська книга, 2023. – 304 с.

26. Лісовська, Ю. П. Кібербезпека: ризики та заходи : навч. посібник / Ю. П. Лісовська. – Київ : Кондор, 2021. – 272 с.

27. Засорнов, О. С. Програмування мікроконтролерних та робототехнічних систем : навч. посібник / О. С. Засорнов, І. О. Засорнова. – Київ : Кондор, 2023. – 280 с.

28. Величко, О. М. Інтелектуальні інформаційні системи: структура і застосування : підручник / О. М. Величко, Т. Б. Гордієнко ; Держ. ун-т телекомунікацій. – Херсон : Олді-плюс, 2022. – 728 с.

29. Крамаренко С. П. "Проєктування інформаційних систем". Харків: ХНУРЕ, 2020. 172 с.

30. Теоретичні та практичні аспекти розвитку Інтернет-економіки : навч. посібник / О. О. Шульга, О. І. Вівчар, М. Ф. Зайлик [та ін.] ; за наук. ред. : І. Л. Татомир, Л. Г. Квасній ; Прикарпат. ін-т ім. М. Грушевського ПрАТ ВНЗ "МАУП". – Дрогобич-Трускавець : Просвіт, 2021. – 386 с.

ДОДАТКИ

Додаток А. Функціональні моделі

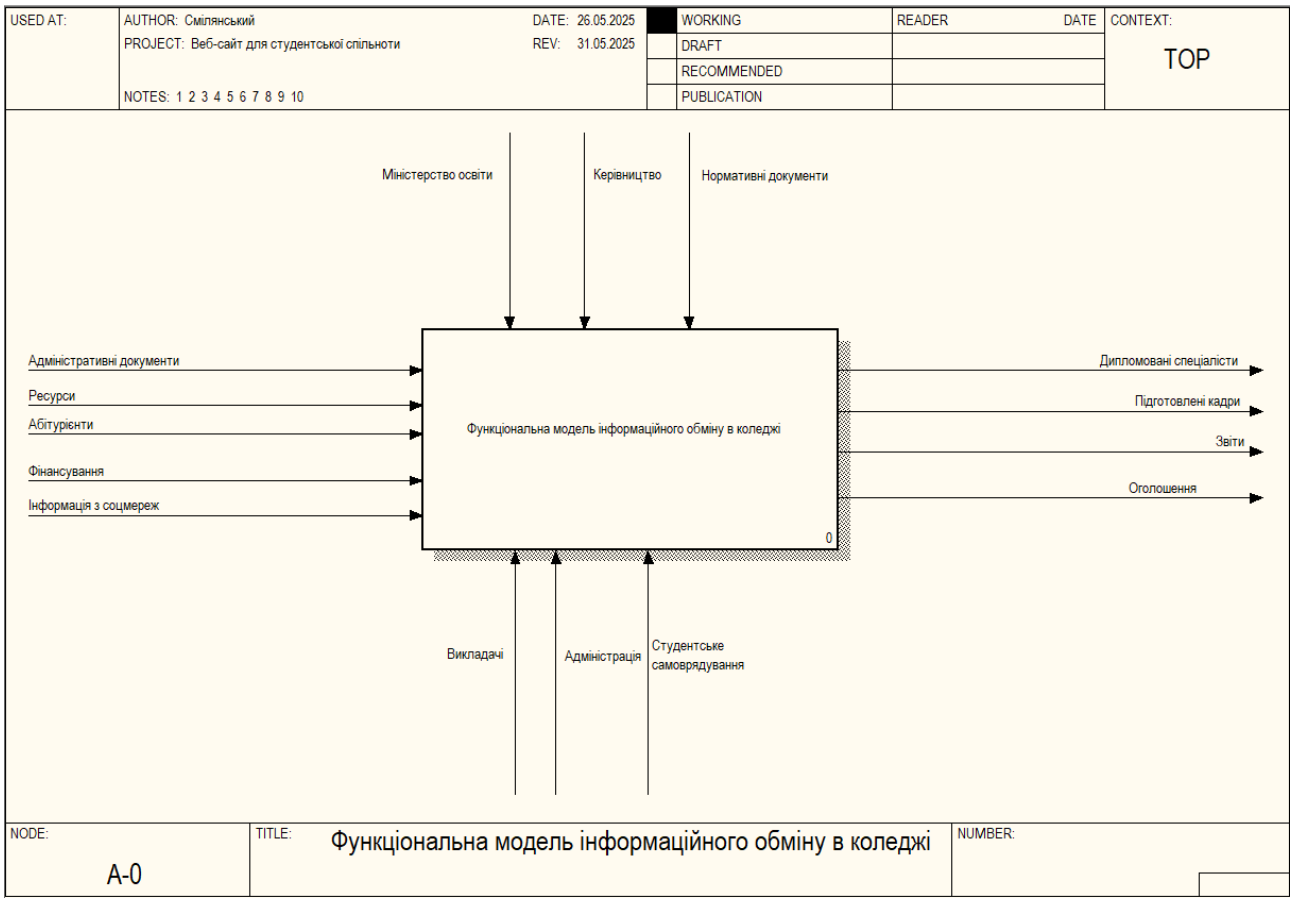

Рисунок А.1 – Перший рівень діаграми IDEF0

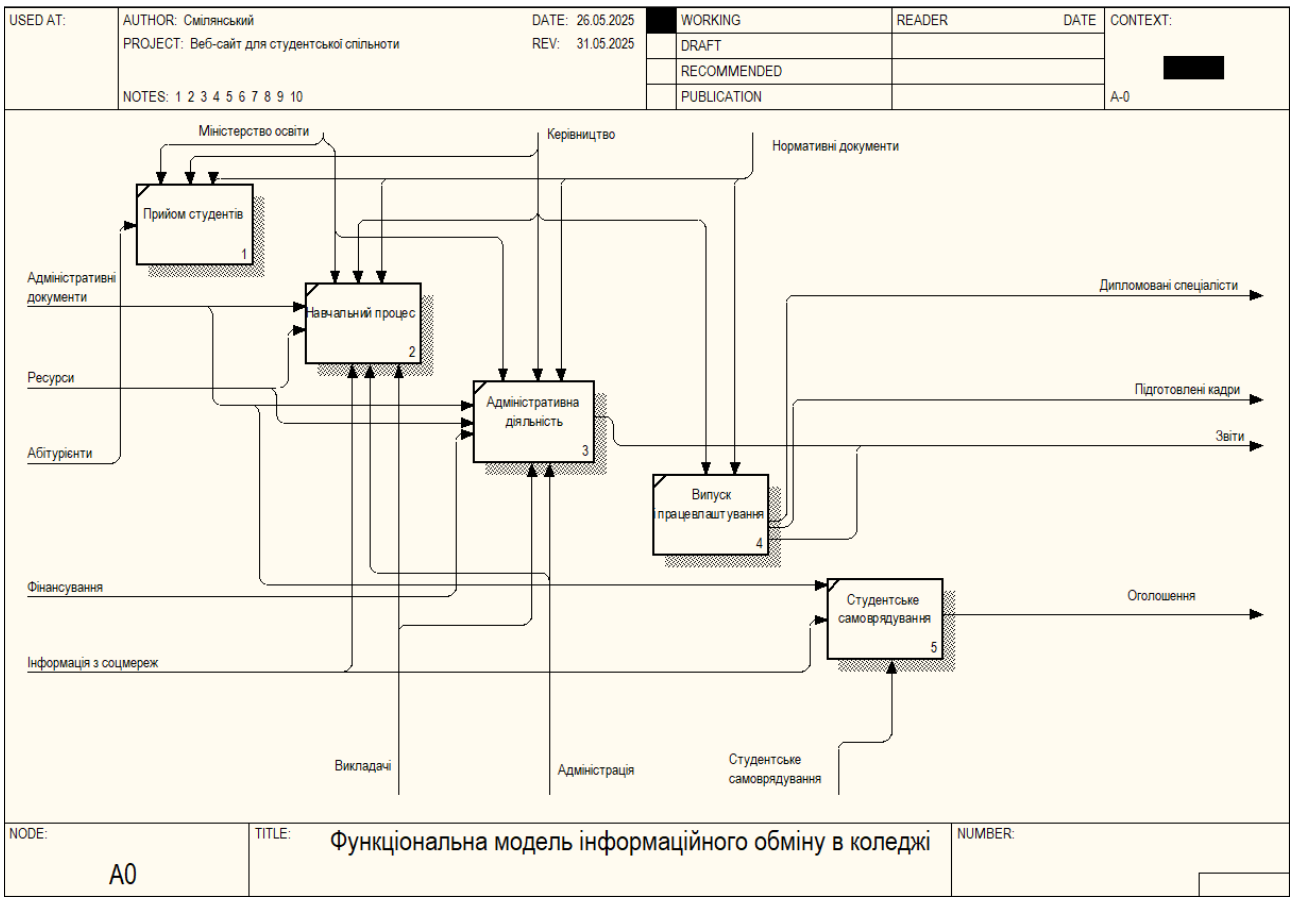

Рисунок А.2 - Другий рівень декомпозиції

```python
@auth_bp.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        remember = True if request.form.get('remember') else False
        user = User.query.filter_by(email=email).first()
        if user and check_password_hash(user.password, password):
            login_user(user, remember=remember)
            return redirect(url_for('home'))  # ← після входу — на захищену головну
        else:
            flash('Невірний логін або пароль')
    return render_template('login.html')

@auth_bp.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = generate_password_hash(request.form['password'])
        role = 'student'  # Автоматично призначаємо роль студента
        new_user = User(username=username, email=email, password=password, role=role)
        db.session.add(new_user)
        db.session.commit()
        flash('Успішна реєстрація! Увійдіть.')
        return redirect(url_for('auth.login'))
    return render_template('register.html')

@auth_bp.route('/logout')
@login_required
def logout():
    logout_user()
    return redirect(url_for('auth.login'))
```

Рисунок Б.1 – Реєстрація та вхід

```python
#  Авторизація
login_manager = LoginManager()
login_manager.login_view = 'auth.login'  # важливо: 'auth.login', бо це назва blueprint + функція
login_manager.init_app(app)

@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))
```

Рисунок Б.2 – Авторизація

```python
def create_announcement():
    if current_user.role != 'admin':
        flash("У вас немає прав для створення оголошення.")
        return redirect(url_for('announcements'))

    if request.method == 'POST':
        title = request.form['title']
        content = request.form['content']
        new_announcement = Announcement(title=title, content=content)
        db.session.add(new_announcement)
        db.session.commit()
        flash("Оголошення опубліковано.")
        return redirect(url_for('announcements'))

    return render_template('create_announcement.html')

@app.route('/announcements/delete/<int:announcement_id>', methods=['POST'])
@login_required
def delete_announcement(announcement_id):
    if current_user.role != 'admin':
        flash("У вас немає прав для видалення оголошень.")
        return redirect(url_for('announcements'))

    announcement = Announcement.query.get_or_404(announcement_id)
    db.session.delete(announcement)
    db.session.commit()
    flash("Оголошення видалено.")
    return redirect(url_for('announcements'))
```

Рисунок Б.3 – Валідація для створення оголошень

```python
@app.route('/schedule/create', methods=['GET', 'POST'])
@login_required
def create_lesson():
    if current_user.role != 'admin':
        flash("У вас немає прав для створення занять.")
        return redirect(url_for('schedule'))

    if request.method == 'POST':
        subject = request.form['subject']
        teacher = request.form['teacher']
        classroom = request.form['classroom']
        group_name = request.form.get('group_name')
        date_str = request.form['date']
        date_end_str = request.form.get('date_end')
        time_str = request.form['time']

        try:
            lesson_date = datetime.strptime(date_str, '%Y-%m-%d').date()
            lesson_time = datetime.strptime(time_str, '%H:%M').time()
            lesson_date_end = datetime.strptime(date_end_str, '%Y-%m-%d').date() if date_end_str else None

            new_lesson = Lesson(
                subject=subject,
                teacher=teacher,
                classroom=classroom,
                group_name=group_name,
                date=lesson_date,
                date_end=lesson_date_end,
                time=lesson_time
            )
            db.session.add(new_lesson)
            db.session.commit()
            flash("Заняття додано.")
```

Рисунок Б.4 – Валідація для створення занять

```python
@app.route('/schedule/delete/<int:lesson_id>', methods=['POST'])
@login_required
def delete_lesson(lesson_id):
    if current_user.role != 'admin':
        flash("У вас немає прав для видалення занять.")
        return redirect(url_for('schedule'))

    lesson = Lesson.query.get_or_404(lesson_id)
    db.session.delete(lesson)
    db.session.commit()
    flash("Заняття видалено.")
    return redirect(url_for('schedule'))
```

Рисунок Б.5 – Валідація для видалення занять

```python
        try:
            event_date = datetime.strptime(date_str, '%Y-%m-%d').date()
            event_time = datetime.strptime(time_str, '%H:%M').time()

            new_event = Event(title=title, description=description, location=location, date=e
            db.session.add(new_event)
            db.session.commit()
            flash("Подію створено.")
            return redirect(url_for('events'))
        except Exception as e:
            flash("Помилка під час створення події: " + str(e))

    return render_template('create_event.html')
```

Рисунок Б.6 – Валідація для створення події

```python
@app.route('/events/<int:event_id>', methods=['GET', 'POST'])
@login_required
def view_event(event_id):
    event = Event.query.get_or_404(event_id)
    comments = EventComment.query.filter_by(event_id=event.id).order_by(EventComment.created_at.desc()).all()

    if request.method == 'POST':
        content = request.form.get('comment')
        if content:
            comment = EventComment(content=content, event_id=event.id, user_id=current_user.id)
            db.session.add(comment)
            db.session.commit()
            flash("Коментар додано.")
            return redirect(url_for('view_event', event_id=event.id))

    return render_template('view_event.html', event=event, comments=comments)
```

Рисунок Б.7 – Функція додавання коментарів

```python
@app.route('/lost-found/create', methods=['GET', 'POST'])
@login_required
def create_item():
    if request.method == 'POST':
        title = request.form['title']
        description = request.form['description']
        contact = request.form['contact']
        item_type = request.form['type']  # 'lost' або 'found'
        new_item = LostFoundItem(title=title, description=description, contact=contact, type=item_type)
        db.session.add(new_item)
        db.session.commit()
        flash('Річ успішно додано.')
        return redirect(url_for('lost_found'))

    return render_template('create_item.html')


@app.route('/lost-found/delete/<int:item_id>', methods=['POST'])
@login_required
def delete_item(item_id):
    item = LostFoundItem.query.get_or_404(item_id)
    db.session.delete(item)
    db.session.commit()
    flash('Запис успішно видалено.')
    return redirect(url_for('lost_found'))
```

Рисунок Б.8 – Модуль загублені/знайдені речі

```python
@app.route('/profile', methods=['GET', 'POST'])
@login_required
def profile():
    if request.method == 'POST':
        name = request.form.get('name')
        email = request.form.get('email')
        old_password = request.form.get('old_password')
        new_password = request.form.get('new_password')
        confirm_password = request.form.get('confirm_password')

        # Update name and email
        if name:
            current_user.username = name
        if email:
            current_user.email = email

        # Change password if fields are filled
        if old_password and new_password and confirm_password:
            if not check_password_hash(current_user.password, old_password):
                flash('Старий пароль невірний.', 'danger')
            elif new_password != confirm_password:
                flash('Новий пароль не співпадає з підтвердженням.', 'danger')
            elif len(new_password) < 6:
                flash('Новий пароль має бути не менше 6 символів.', 'danger')
            else:
                current_user.password = generate_password_hash(new_password)
                flash('Пароль успішно змінено.', 'success')
        db.session.commit()
        flash('Профіль оновлено.', 'success')
        return redirect(url_for('profile'))

    return render_template('profile.html')
```

Рисунок Б.9 – Функція зміни паролю

```python
@app.route('/sports/delete/<int:event_id>', methods=['POST'])
@login_required
def delete_sport_event(event_id):
    if current_user.role != 'admin':
        flash('У вас немає прав для видалення спортивних подій')
        return redirect(url_for('sports'))

    event = SportEvent.query.get_or_404(event_id)
    db.session.delete(event)
    db.session.commit()
    flash('Спортивну подію видалено')
    return redirect(url_for('sports'))

@app.route('/announcements/edit/<int:announcement_id>', methods=['GET', 'POST'])
@login_required
def edit_announcement(announcement_id):
    if current_user.role != 'admin':
        flash('У вас немає прав для редагування оголошень')
        return redirect(url_for('announcements'))
```

Рисунок Б.10 – Валідація для видалення спортивних подій