

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ГОРБАЛЬ Юлія-Марія Мар'янівна

Веб-орієнтована інформаційна система керування контентом міжнародного освітнього проєкту / Web-based Information System for Content Management of an International Educational Project

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконала студентка групи КН-41
Ю.-М.М. Горбаль

Науковий керівник:
д.т.н., професор М.П. Комар

Кваліфікаційну роботу допущено до
захисту

«___»_____ 2026 р.

Завідувач кафедри

_____ Н.В. Дзюбановська

Тернопіль – 2026

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
Н.В. Дзюбановська
«_____» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ
ГОРБАЛЬ Юлії-Марії Мар'янівні

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

Веб-орієнтована інформаційна система керування контентом міжнародного освітнього проєкту / Web-based Information System for Content Management of an International Educational Project

керівник роботи д.т.н., професор М. П. Комар

затверджені наказом по університету від 01 грудня 2025 року № 894.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2026 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити

- проаналізувати предметну область міжнародних освітніх проєктів та особливості їх інформаційного супроводу;

- дослідити сучасні підходи до побудови веб-орієнтованих систем керування контентом;

- визначити функціональні та нефункціональні вимоги до інформаційної системи;

- розробити структуру та архітектуру веб-орієнтованої інформаційної системи;

- сформулювати модель даних і логіку взаємодії основних модулів системи;

- розробити алгоритми роботи системи;

- реалізувати програмний прототип системи керування контентом міжнародного освітнього проєкту;

- виконати тестування розробленого рішення.

5. Перелік графічного матеріалу в роботі

- схема архітектури веб-орієнтованої інформаційної системи;

- схема взаємозв'язків інформаційних об'єктів веб-орієнтованої інформаційної системи;

- функціональна схема модулів веб-орієнтованої інформаційної системи;

- схема алгоритму відображення контенту в публічній частині сайту;

- схема алгоритму створення та публікації контенту;

- схема алгоритму перекладу контенту;

- схема алгоритму динамічного підвантаження даних;
- схема алгоритму оброблення повідомлень із форм зворотного зв'язку;
- узагальнена структурна схема програмної реалізації веб-орієнтованої інформаційної системи.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2026 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2026 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 15.03.2026 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2026 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2026 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2026 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2026 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2026 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06. 2026 р.	

Студентка _____ Ю.-М.М. Горбаль
підпис

Керівник роботи _____ д.т.н., професор М. П. Комар
підпис

АНОТАЦІЯ

Кваліфікаційна робота на тему «Веб-орієнтована інформаційна система керування контентом міжнародного освітнього проєкту» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 93 сторінки і містить 15 ілюстрацій, 20 таблиць, 2 додатки та 29 використаних джерел.

Метою роботи є розроблення веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту, яка забезпечує структуроване подання, зберігання, оновлення та адміністрування інформаційних матеріалів у багатомовному веб-середовищі.

Методи проєктування включають методи системного та структурного проєктування, методи проєктування баз даних, методи об'єктно-орієнтованого проєктування, методи веб-проєктування.

Розроблено веб-орієнтовану інформаційну систему керування контентом міжнародного освітнього проєкту, яка забезпечує подання інформації про проєкт, публікацію новин, представлення партнерів, розміщення ресурсів, підтримку галереї, навчальних розділів і сторінки залучення користувачів.

Результати дослідження можуть бути використані для створення та супроводу веб-платформ міжнародних освітніх проєктів, розроблення систем керування контентом для освітніх ініціатив, організації багатомовних інформаційних ресурсів, автоматизації адміністрування новин, матеріалів, партнерських сторінок і навчальних розділів, а також для подальшого розвитку цифрових платформ у сфері міжнародної освітньої співпраці.

Ключові слова: ВЕБ-ОРІЄНТОВАНА ІНФОРМАЦІЙНА СИСТЕМА, КЕРУВАННЯ КОНТЕНТОМ, МІЖНАРОДНИЙ ОСВІТНІЙ ПРОЄКТ, CMS, LARAVEL, PHP, BLADE, MYSQL, FILAMENT, LIVEWIRE, DOCKER, МУЛЬТИМОВНІСТЬ, DEEPL, ВЕБ-ПЛАТФОРМА.

ANNOTATION

Qualification work on the topic «Web-based Information System for Content Management of an International Educational Project» for Bachelor's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 93 pages and it contains 15 figures, 20 tables, 2 annexes and 29 sources.

The purpose of the work is to develop a web-oriented information system for content management of an international educational project, which ensures the structured presentation, storage, updating, and administration of information materials in a multilingual web environment.

The design methods include methods of system and structural design, database design methods, object-oriented design methods, and web design methods.

A web-oriented information system for content management of an international educational project has been developed. It provides the presentation of information about the project, publication of news, presentation of partners, placement of resources, support for the gallery, educational sections, and the user engagement page.

The research results can be used for the creation and support of web platforms for international educational projects, the development of content management systems for educational initiatives, the organization of multilingual information resources, the automation of administration of news, materials, partner pages, and educational sections, as well as for the further development of digital platforms in the field of international educational cooperation.

Keywords: WEB-ORIENTED INFORMATION SYSTEM, CONTENT MANAGEMENT, INTERNATIONAL EDUCATIONAL PROJECT, CMS, LARAVEL, PHP, BLADE, MYSQL, FILAMENT, LIVEWIRE, DOCKER, MULTILINGUALISM, DEEPL, WEB PLATFORM.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної області та постановка задачі	10
1.1 Опис предметної області	10
1.2 Аналіз існуючих підходів та методів	14
1.3 Постановка задачі дослідження.....	21
2 Проєктування веб-орієнтованої інформаційної системи керування контентом .	24
2.1 Формування вимог до системи та загальної архітектури	24
2.2 Проєктування структури даних і моделі контентних інформаційних об'єктів	26
2.3 Проєктування функціональної структури системи.....	29
2.4 Алгоритми роботи веб-орієнтованої інформаційної системи	33
3 Програмна реалізація та тестування веб-орієнтованої інформаційної системи..	38
3.1 Реалізація структури програмного проєкту та серверної частини системи...	38
3.2 Реалізація публічної частини сайту.....	45
3.3 Реалізація адміністративної панелі та керування контентом	51
3.4 Реалізація мультимовності та динамічного підвантаження даних	56
3.5 Тестування веб-орієнтованої інформаційної системи.....	59
Висновки	64
Список використаних джерел	67
Додаток А Лістинги програмного коду	70
Додаток Б Апробація результатів роботи.....	88

ВСТУП

У сучасних умовах цифрової трансформації освітнього середовища веб-орієнтовані інформаційні системи набули важливого значення для організації, супроводу та поширення матеріалів міжнародних освітніх ініціатив. Зростання кількості міжнародних програм, партнерських мереж, навчальних платформ, інформаційних кампаній і спільних цифрових ресурсів зумовило потребу у створенні зручних засобів централізованого керування контентом. Особливої актуальності це набуває для міжнародних освітніх проєктів, у межах яких необхідно забезпечити регулярне оновлення новин, публікацію матеріалів, підтримку багатомовності, представлення партнерів, висвітлення заходів, а також доступність інформації для різних категорій користувачів [8, 9, 22].

Традиційне розміщення матеріалів без єдиного підходу до їх структурування, оновлення та адміністрування ускладнює інформаційний супровід проєкту. За відсутності спеціалізованої веб-орієнтованої системи виникають труднощі з підтримкою актуальності контенту, його логічною організацією, повторним використанням, розмежуванням прав доступу та забезпеченням однакового стилю подання інформації. У таких умовах система керування контентом розглядається не лише як технічний інструмент публікації матеріалів, а як засіб упорядкування інформаційних потоків, підтримки комунікації та представлення результатів міжнародного освітнього проєкту в цифровому середовищі [12, 20, 22].

Актуальність теми визначається також тим, що сучасна веб-орієнтована інформаційна система має відповідати не лише функціональним вимогам, а й вимогам адаптивності, безпеки, масштабованості, зручності використання та доступності вебконтенту. Під час проєктування таких систем враховуються підходи до організації вебконтенту, принципи адаптивного дизайну, вимоги до вебформ, правила міжнародного та багатомовного представлення інформації, а також рекомендації щодо вебдоступності й захисту даних [3–7, 10, 14–17, 23–25]. Це зумовлює необхідність комплексного підходу, у межах якого поєднуються засоби веброзроблення, моделі керування контентом, архітектурні рішення та практичні методи реалізації.

Науково-практичний інтерес до проблеми створення систем керування контентом підтверджується низкою праць, присвячених еволюції систем управління веб-контентом (content management systems – CMS), архітектурі платформ керування інформацією, модельно-керованому проектуванню CMS та застосуванню інтелектуальних підходів до побудови контентних систем [1, 11, 12, 19–21, 26, 27]. У цих дослідженнях висвітлено як загальні принципи побудови систем керування контентом, так і окремі аспекти їх адаптації до різних предметних областей. Водночас специфіка міжнародного освітнього проєкту вимагає врахування додаткових особливостей: багатомовної структури ресурсу, ролей учасників, наявності навчальних та інформаційних розділів, а також постійної підтримки актуальності опублікованих матеріалів [8, 9, 23, 24].

Практична значущість роботи полягає у можливості створення веб-орієнтованої інформаційної системи, що забезпечує централізоване керування контентом міжнародного освітнього проєкту, спрощує процес наповнення ресурсу, підвищує зручність адміністрування та покращує доступ користувачів до структурованої інформації. Розроблення такого рішення дозволяє поєднати в межах єдиної системи представлення новин, інформації про партнерів, матеріалів проєкту, галереї, інформації про заходи та іншого контенту, що формують цифрову екосистему проєкту.

Метою роботи є розроблення веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту, яка забезпечує структуроване подання, зберігання, оновлення та адміністрування інформаційних матеріалів у багатомовному веб-середовищі.

Для досягнення поставленої мети сформовано такі завдання дослідження:

- проаналізувати предметну область міжнародних освітніх проєктів та особливості їх інформаційного супроводу;
- дослідити сучасні підходи до побудови веб-орієнтованих систем керування контентом;
- визначити функціональні та нефункціональні вимоги до інформаційної системи;
- розробити структуру та архітектуру веб-орієнтованої інформаційної

системи;

- сформувати модель даних і логіку взаємодії основних модулів системи;
- розробити алгоритми роботи системи;
- реалізувати програмний прототип системи керування контентом міжнародного освітнього проєкту;
- виконати тестування розробленого рішення.

Об'єктом дослідження є процеси інформаційного супроводу та керування контентом міжнародного освітнього проєкту.

Предметом дослідження є методи, моделі, архітектурні підходи та програмні засоби розроблення веб-орієнтованої інформаційної системи керування контентом.

Результати кваліфікаційної роботи апробовані та опубліковані у матеріалах Міжнародної наукової інтернет-конференції «Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення», (м. Тернопіль, Україна, м. Ополе, Польща, 7-8 травня 2026 р.).

Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної області

Предметна область даної роботи охоплює процеси створення, організації, оновлення та публікації цифрового контенту в межах міжнародного освітнього проєкту. У сучасному інформаційному середовищі такі проєкти функціонують як багатокomпонентні комунікаційні системи, у яких поєднуються інформаційна підтримка, представлення результатів діяльності, взаємодія між партнерами, поширення навчальних матеріалів, а також забезпечення відкритого доступу до ресурсів для різних категорій користувачів [8, 12].

Міжнародний освітній проєкт, як правило, передбачає участь кількох організацій із різних країн, спільне планування активностей, підготовку навчального та інформаційного контенту, публікацію новин, організацію подій, координацію комунікації з цільовими аудиторіями та звітування про результати. За таких умов вебресурс виконує не лише представницьку функцію, а стає центральним цифровим середовищем, у якому акумулюється основна інформація про проєкт, його учасників, матеріали, заходи та результати [8, 22]. Саме тому ефективність інформаційного супроводу значною мірою залежить від наявності веб-орієнтованої інформаційної системи, здатної забезпечити централізоване керування контентом.

У межах цієї предметної області ключовим поняттям є контент. Під контентом доцільно розуміти сукупність інформаційних об'єктів, що публікуються у вебсистемі: текстові матеріали, новини, зображення, документи, навчальні ресурси, відомості про партнерів, анонси подій, мультимедійні елементи, контактні дані та інші структуровані сутності. У системах керування контентом контент розглядається не як набір окремих сторінок, а як сукупність логічно пов'язаних об'єктів, що мають власні атрибути, категорії, дату створення, статус публікації, автора та інші параметри [1, 12, 20]. Такий підхід забезпечує повторне використання інформації, спрощує оновлення вмісту та підвищує узгодженість структури ресурсу.

Особливістю міжнародного освітнього проєкту є те, що його контент має

різнорідний характер. Одна частина матеріалів орієнтована на широке коло користувачів і виконує інформаційно-презентаційну функцію. До неї належать опис проєкту, новини, відомості про партнерські організації, фотоматеріали, анонси заходів і результати діяльності. Інша частина контенту має освітнє або методичне призначення й включає навчальні модулі, документи, презентації, довідкові матеріали, форми для зворотного зв'язку та ресурси для учасників проєкту. Відповідно, веб-орієнтована система має підтримувати різні типи контентних сутностей, різні режими доступу до них і різні сценарії адміністрування [8, 20, 22].

Основні компоненти веб-орієнтованої інформаційної системи міжнародного освітнього проєкту:

- загальна інформація про міжнародний освітній проєкт;
- партнери та координатори;
- інформаційні матеріали;
- навчальні ресурси;
- новини та події;
- користувачі й цільові аудиторії;
- адміністративне керування контентом;
- вебплатформа як середовище публікації та взаємодії.

Функціонування міжнародного освітнього проєкту у вебсередовищі пов'язане з кількома групами користувачів. До них належать адміністратори системи, контент-менеджери, редактори, партнери проєкту, зареєстровані учасники та зовнішні відвідувачі ресурсу. Кожна з цих груп має власні інформаційні потреби та власний рівень доступу до функцій системи. Адміністратори відповідають за загальне налаштування платформи, структуру розділів, керування обліковими записами та контроль доступу. Контент-менеджери й редактори здійснюють підготовку, редагування та публікацію матеріалів. Партнери можуть надавати інформацію про активності, результати й документи. Зовнішні користувачі переглядають опубліковані ресурси, новини та іншу загальнодоступну інформацію [7, 22].

З позиції інформаційних процесів предметна область включає кілька

взаємопов'язаних етапів:

- створення або отримання контенту;
- структурування матеріалів за категоріями та розділами;
- редагування і перевірку змісту;
- публікацію у вебсистемі;
- оновлення, архівування або видалення застарілих матеріалів;
- забезпечення доступу користувачів до опублікованого вмісту.

Наведена послідовність відображає життєвий цикл контенту в системі керування вмістом, який у наукових джерелах розглядається як одна з базових характеристик web content management systems [12]. Для міжнародного освітнього проєкту цей цикл ускладнюється тим, що матеріали часто створюються різними учасниками, потребують узгодження, перекладу кількома мовами та подальшого супроводу після публікації.

Однією з головних характеристик предметної області є багатомовність. Міжнародний освітній проєкт орієнтований на учасників із різних країн, тому вебсистема повинна забезпечувати представлення контенту кількома мовами, підтримку відповідних мовних версій сторінок, коректне структурування навігації та збереження смислової узгодженості між перекладами [9, 23, 24]. Багатомовність у цьому випадку не є додатковою опцією, а виступає однією з основних вимог до системи. Вона впливає і на модель даних, і на структуру інтерфейсу, і на логіку керування контентом.

Не менш важливою характеристикою предметної області є адаптивність вебресурсу. Користувачі міжнародних освітніх проєктів взаємодіють із платформою з різних пристроїв: персональних комп'ютерів, планшетів, смартфонів. Це вимагає використання адаптивного дизайну, гнучкої сітки сторінки, коректного відображення форм, меню, мультимедійних елементів та інформаційних блоків на різних екранах [3–6, 14, 15]. Відсутність адаптивності знижує зручність використання системи та обмежує доступність її функцій для частини аудиторії.

Ще одним важливим аспектом є вебдоступність. Оскільки міжнародний освітній проєкт орієнтований на широку аудиторію, інформаційна система повинна

враховувати базові принципи доступності контенту для користувачів з різними потребами. Йдеться про зрозумілу структуру сторінок, логічну навігацію, коректне використання заголовків, альтернативний текст для зображень, достатню контрастність і сумісність із допоміжними технологіями [25]. Дотримання таких вимог підвищує якість системи та розширює коло її потенційних користувачів.

Окрему роль у предметній області відіграють вебформи та механізми зворотного зв'язку. У міжнародних освітніх проєктах форми використовуються для реєстрації учасників, подання заявок, надсилання запитів, отримання відгуків, збору контактної інформації та комунікації з цільовими групами. Тому система повинна підтримувати створення та оброблення форм, перевірку введених даних, захист від помилок і зручну подачу результатів адміністратору [3, 6, 16]. У межах предметної області форми є не допоміжним, а функціонально значущим елементом, оскільки забезпечують двосторонню взаємодію між платформою та користувачами.

Веб-орієнтована інформаційна система міжнародного освітнього проєкту також функціонує в умовах підвищених вимог до інформаційної безпеки та захисту даних. У разі роботи з обліковими записами, контактною інформацією, матеріалами партнерів, формами реєстрації та адміністративною панеллю необхідно враховувати ризики несанкціонованого доступу, втрати даних, уразливостей вебзастосунків і порушення конфіденційності [10, 17]. Відповідно, предметна область включає не лише керування контентом як таким, а й організацію безпечного середовища його зберігання, редагування та публікації.

Для систем цього класу характерною є також наявність структурованої моделі даних. У ній зазвичай виділяються сутності типу «сторінка», «новина», «подія», «партнер», «ресурс», «категорія», «користувач», «мова», «форма зворотного зв'язку», між якими встановлюються логічні зв'язки. Таке структурування дозволяє розглядати вебресурс як інформаційну систему, а не як набір статичних сторінок [7, 18, 20]. У результаті забезпечується зручне керування контентом, підтримується його повторне використання, автоматизується частина адміністративних операцій і спрощується розширення функціональності системи.

Отже, предметна область даної роботи характеризується наявністю

багатокомпонентного контенту, множинністю користувачів, вимогами до багатомовності, адаптивності, доступності та безпеки, а також потребою в централізованому керуванні інформаційними ресурсами міжнародного освітнього проєкту. Проведений аналіз показав, що для ефективного вирішення зазначених задач необхідне застосування спеціалізованої веб-орієнтованої інформаційної системи керування контентом.

1.2 Аналіз існуючих підходів та методів

Розроблення веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту потребує вибору таких підходів і методів, які забезпечують зручне адміністрування, структуроване зберігання даних, підтримку багатомовності, адаптивність інтерфейсу, безпечне оброблення інформації та можливість подальшого розширення функціональності. Для досягнення цієї мети доцільно проаналізувати сучасні підходи до побудови контентних систем, методи серверної та клієнтської розробки, засоби організації адміністративного керування, а також підходи до реалізації інтерактивної взаємодії з користувачем [1, 2, 12].

Одним із базових напрямів у цій предметній області є підхід, заснований на використанні систем керування контентом. CMS-рішення призначені для створення, редагування, публікації та оновлення інформаційних матеріалів без прямого редагування програмного коду. У праці S. McKeever системи керування вебконтентом розглянуто як інструменти, що забезпечують повний життєвий цикл інформаційного наповнення вебресурсу: від створення до архівування та повторного використання [12]. Для міжнародного освітнього проєкту такий підхід є принципово важливим, оскільки контент ресурсу постійно змінюється, доповнюється новинами, матеріалами, подіями, інформацією про партнерів і результатами проєктної діяльності.

У сучасній практиці можна виділити кілька підходів до побудови веб-орієнтованих систем керування контентом. Перший підхід передбачає використання готових універсальних CMS-платформ. Їх перевага полягає у швидкому запуску та наявності типових механізмів адміністрування. Проте для

міжнародного освітнього проєкту такий варіант часто виявляється недостатньо гнучким, оскільки структура даних, логіка багатомовності, особливості представлення матеріалів і сценарії адміністрування можуть вимагати індивідуального налаштування або розроблення нестандартних модулів. У результаті система поступово втрачає цілісність, а її підтримка ускладнюється.

Другий підхід базується на індивідуальному розробленні вебзастосунку на основі серверного фреймворку. У цьому випадку система створюється під конкретну предметну область, а структура контенту, ролі користувачів, модулі, інтерфейси та логіка взаємодії визначаються відповідно до потреб проєкту. Саме цей підхід є більш придатним для теми даної роботи, оскільки дозволяє побудувати систему не як універсальний шаблонний сайт, а як спеціалізовану інформаційну платформу з чітко визначеними контентними сутностями та правилами їх опрацювання [20, 21, 26].

Окрему групу становлять підходи, орієнтовані на інтелектуалізацію та автоматизацію контентних процесів. У працях [1, 11] розглянуто застосування онтологічного моделювання, SEO-орієнтованого структурування та інтелектуального розподілу вебконтенту. Такі методи є корисними для складних систем із великою кількістю інформаційних об'єктів і розвиненими механізмами аналітики. Проте для веб-орієнтованої інформаційної системи міжнародного освітнього проєкту на першому плані перебувають не алгоритми інтелектуального аналізу, а стабільне керування сторінками, новинами, матеріалами, формами, мовними версіями та ролями доступу. Тому інтелектуалізовані підходи в даній роботі доцільно розглядати як перспективний напрям подальшого розвитку, але не як основу первинної реалізації [1, 11].

Значну увагу слід приділити методам серверної розробки, оскільки саме вони визначають логіку функціонування інформаційної системи. Для реалізації такої системи доцільним є використання фреймворкового підходу на базі Laravel 12. Цей підхід забезпечує структуровану організацію коду, підтримку маршрутизації, моделей даних, контролерів, шаблонів подання, автентифікації, валідації та інтеграції з реляційною базою даних. На відміну від створення вебзастосунку без фреймворку, використання Laravel дозволяє реалізувати систему з чітким

розмежуванням логічних компонентів і спрощує супровід програмного продукту в майбутньому. Для задачі керування контентом це особливо важливо, оскільки контентні сутності мають бути пов'язані між собою, а адміністрування повинно здійснюватися в уніфікованому середовищі.

В основі Laravel лежить архітектурний підхід MVC. У межах цього підходу модель відповідає за роботу з даними, контролер – за оброблення запитів і бізнес-логіку, а представлення – за відображення інформації користувачеві. Такий розподіл дозволяє підвищити зрозумілість програмної структури та полегшує масштабування системи. Для міжнародного освітнього проєкту MVC-підхід є виправданим, оскільки в системі одночасно співіснують різні типи контенту, окремі адміністративні сценарії, модулі публічної частини сайту та механізми взаємодії з базою даних.

Для формування користувацького інтерфейсу в роботі використовується підхід на основі Blade-шаблонів. Blade є серверним шаблонізатором Laravel, який дозволяє поєднати HTML-розмітку з логікою відображення даних без ускладнення клієнтської частини. Перевага цього підходу полягає в тому, що він забезпечує динамічне формування сторінок, повторне використання шаблонних компонентів, єдину структуру макетів і зручність підтримки багатосторінкового вебресурсу. Для інформаційної системи міжнародного освітнього проєкту це є важливим, оскільки сайт містить однотипні елементи навігації, блоки новин, сторінки партнерів, ресурси, галерею та інші повторювані компоненти, які доцільно реалізовувати через шаблонний механізм.

Загальні підходи до побудови вебінтерфейсу, вебформ і адаптивного дизайну описано в документації Bootstrap та MDN Web Docs [3–6, 13–16]. Ці джерела демонструють, що сучасний інтерфейс повинен підтримувати коректну роботу на різних типах пристроїв, забезпечувати зручне введення даних у формах і мати логічну структуру сторінок. У контексті даної роботи такі методи використовуються на рівні HTML, CSS і JavaScript у складі Blade-шаблонів. Отже, для реалізації фронтенду доцільно поєднати серверне шаблонування з класичними клієнтськими вебтехнологіями, що забезпечують адаптивність, візуальну цілісність і зручність користування системою [3–6, 14–16].

Важливим елементом аналізу є підхід до організації адміністративної панелі. У типовій системі керування контентом значна частина функціональності пов'язана саме з адмініструванням: створенням, редагуванням, модерацією, переглядом і видаленням контенту. Існує два основні підходи до реалізації такого функціоналу. Перший підхід полягає у повністю ручному створенні адміністративного модуля. Його перевагою є максимальна гнучкість, але недоліком – значні витрати часу на реалізацію типових операцій. Другий підхід передбачає використання спеціалізованих адміністративних пакетів. У межах даної роботи обрано саме цей варіант, оскільки для Laravel існує пакет Filament v5, який дозволяє швидко сформувати функціональну адміністративну панель для роботи з моделями, таблицями, формами, фільтрами та правами доступу. Для системи міжнародного освітнього проєкту це є ефективним рішенням, оскільки скорочує обсяг допоміжного програмування і дає змогу зосередитися на предметно-орієнтованій логіці системи.

Порівняння підходів до реалізації адміністративного керування подано в таблиці 1.1.

Таблиця 1.1 – Порівняння підходів до реалізації адміністративної панелі

Підхід	Переваги	Недоліки	Доцільність для теми роботи
Повністю ручна розробка адмін-панелі	Максимальна індивідуалізація, повний контроль над інтерфейсом	Висока трудомісткість, дублювання типових операцій	Середня
Використання готового адміністративного пакета	Швидка реалізація CRUD-операцій, уніфікований інтерфейс, зручне розширення	Певна залежність від архітектури пакета	Висока
Використання універсальної CMS	Швидкий старт, готове адміністрування	Обмежена гнучкість і складність адаптації до специфіки проєкту	Середня або низька

Як видно з таблиці 1.1, для веб-орієнтованої інформаційної системи

міжнародного освітнього проєкту найбільш доцільним є використання спеціалізованого адміністративного пакета в межах фреймворкової архітектури. Саме тому застосування Filament v5 є обґрунтованим як з позиції швидкості реалізації, так і з позиції подальшої підтримки системи.

Окремо слід розглянути методи реалізації інтерактивності та динамічного підвантаження даних. У традиційному багатосторінковому вебзастосунку кожна зміна стану сторінки супроводжується її повним перезавантаженням. Для контентної системи це не завжди є зручним, особливо у випадках перегляду великих списків новин, ресурсів або галерей. Існує кілька підходів до розв'язання цієї задачі: повністю клієнтська реалізація на JavaScript-фреймворках, асинхронні запити через API, а також серверно-керована інтерактивність. У даній роботі доцільним є третій варіант, реалізований за допомогою Livewire 4. Перевага такого підходу полягає в тому, що інтерактивні компоненти залишаються інтегрованими в Laravel-архітектуру, але при цьому забезпечується часткове оновлення сторінки без повного перезавантаження. Це є особливо корисним для сценаріїв, у яких новини або інші матеріали завантажуються порціями, наприклад по 10 елементів за один запит.

З методичної точки зору використання Livewire є компромісним рішенням між повністю статичним серверним відображенням і складною SPA-архітектурою. Для бакалаврської роботи та реального проєктного сайту такий підхід є виправданим, оскільки дозволяє підвищити зручність користування, не ускладнюючи систему надмірною клієнтською логікою.

Для зберігання структурованої інформації у веб-орієнтованій системі використовується реляційний підхід до організації даних. У даній роботі для цього застосовується MySQL. Реляційна модель є доцільною, оскільки контент системи має чітко визначені сутності та зв'язки між ними: новини належать до певних категорій, сторінки мають мовні версії, партнери містять атрибути опису, ресурси можуть бути пов'язані з файлами або зображеннями, а користувачі мають певні ролі та права доступу. Використання реляційної бази даних забезпечує цілісність зберігання, впорядкованість структури та можливість побудови запитів для вибірки, сортування й фільтрації інформації. Аналогічні підходи до використання

структурованих моделей даних у системах керування контентом підтверджуються в дослідженнях [19, 20, 26].

Суттєве значення для міжнародного освітнього проєкту має підтримка багатомовності. Документи European Commission та W3C підкреслюють, що міжнародні вебресурси повинні забезпечувати коректну мовну організацію контенту, зрозумілу навігацію між мовними версіями та цілісність інформаційного подання [8, 9, 23, 24]. Існує кілька підходів до реалізації багатомовності: зберігання перекладів у базі даних, використання окремих мовних файлів, побудова окремих версій сторінок або комбінований підхід. У межах даної роботи використовується комбінований варіант: статичні елементи інтерфейсу перекладаються за допомогою мовних файлів, а контент після створення перекладається із застосуванням DeepL. Такий метод є практичним, оскільки дозволяє відокремити системні текстові елементи від редакційного контенту та спрощує підтримку кількох мовних версій сайту.

Підхід, заснований на використанні мовних файлів, є придатним для меню, заголовків, службових написів, повідомлень інтерфейсу та системних підказок. Натомість застосування DeepL для перекладу створеного контенту дозволяє прискорити роботу редактора, особливо якщо сайт містить новини, статті, описи сторінок і матеріали, які регулярно оновлюються. Разом із тим такий метод потребує редакторського контролю, оскільки автоматизований переклад не завжди забезпечує повну термінологічну та стилістичну точність. Отже, підхід до мультимовності у даній роботі поєднує автоматизацію та контроль якості, що є обґрунтованим для міжнародного проєктного середовища [8, 9, 23, 24].

Не менш важливими є методи забезпечення безпеки та захисту даних. Веб-орієнтована система міжнародного освітнього проєкту працює з обліковими записами адміністраторів, формами зворотного зв'язку, контактною інформацією, файлами та опублікованим контентом. У зв'язку з цим необхідно враховувати ризики несанкціонованого доступу, міжсайтових атак, помилок валідації та витоку інформації. Підходи до захисту вебзастосунків викладено у рекомендаціях OWASP Top Ten та в матеріалах, що стосуються data protection by design and by default [10, 17]. Для даної роботи ці принципи означають необхідність валідації вхідних даних,

розмежування ролей доступу, захисту форм, контрольованої роботи з файлами та належної конфігурації серверного середовища.

Окремий інтерес становить контейнеризований підхід до розгортання системи. Традиційне ручне налаштування середовища виконання часто призводить до відмінностей між локальною розробкою, тестуванням і промисловим запуском. Для усунення цієї проблеми в роботі використовується Docker. Застосування контейнеризації дозволяє уніфікувати середовище для вебсервера, застосунку та бази даних, спрощує конфігурацію залежностей, підвищує відтворюваність розгортання та полегшує перенесення системи між різними серверами. Для бакалаврської роботи це є важливою перевагою, оскільки демонструє не лише програмну реалізацію, а й сучасний підхід до розгортання вебзастосунку.

На рисунку 1.1 подано узагальнену схему проаналізованих підходів.

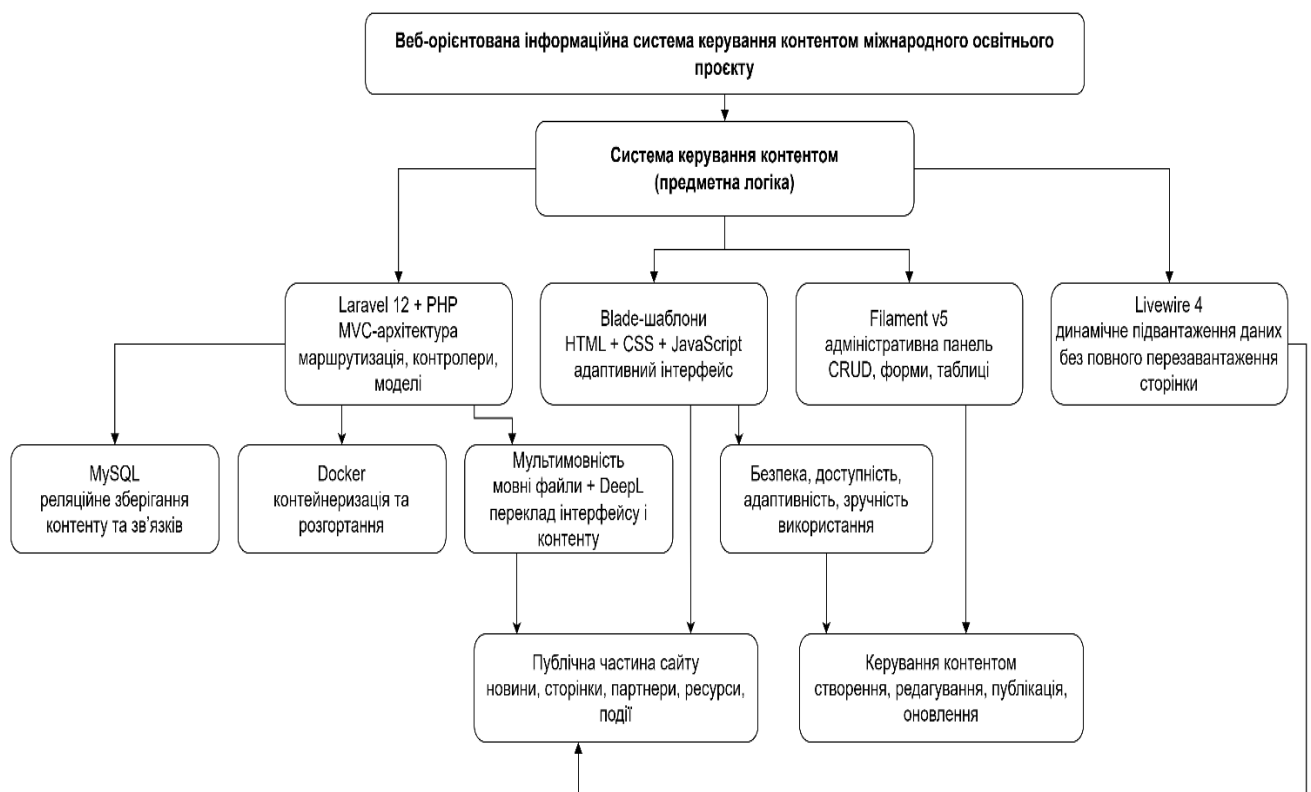


Рисунок 1.1 – Основні підходи та методи побудови веб-орієнтованої інформаційної системи керування контентом

Обґрунтування вибору основних технологій реалізації системи подано в таблиці 1.2.

Таблиця 1.2 – Обґрунтування вибору основних технологій реалізації системи

Компонент	Обрана технологія	Причина вибору
Backend	Laravel 12 + PHP	Чітка структура застосунку, підтримка MVC, маршрутизації, моделей і форм
Frontend	Blade + HTML + CSS + JS	Зручне шаблонування, інтеграція з backend, підтримка багатосторінкової структури
Адмін-панель	Filament v5	Швидка реалізація CRUD-функцій, таблиць, форм і керування контентом
Інтерактивність	Livewire 4	Динамічне підвантаження даних без повного перезавантаження сторінки
База даних	MySQL	Надійне реляційне зберігання контентних сутностей і зв'язків між ними
Розгортання	Docker	Уніфікація середовища, спрощення конфігурування і перенесення системи
Мультимовність	Мовні файли + DeepL	Поєднання статичного перекладу інтерфейсу та автоматизованого перекладу контенту

Отже, проведений аналіз існуючих підходів і методів показав, що для розроблення веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту найбільш доцільним є індивідуальний фреймворковий підхід із використанням Laravel 12. Такий підхід дає змогу реалізувати структуровану серверну логіку, інтегрувати адміністративне середовище на базі Filament v5, забезпечити динамічне підвантаження інформації через Livewire 4, організувати зручний фронтенд на основі Blade-шаблонів, зберігати дані в MySQL та виконувати розгортання системи через Docker. Додатковою перевагою є можливість реалізації мультимовності шляхом поєднання мовних файлів і перекладу контенту через DeepL, що відповідає специфіці міжнародного освітнього проєкту.

1.3 Постановка задачі дослідження

Проведений аналіз предметної області міжнародних освітніх проєктів і сучасних підходів до побудови систем керування контентом показав, що ефективний інформаційний супровід таких ініціатив потребує спеціалізованого веб-орієнтованого програмного рішення [8, 12, 20]. Звичайне розміщення

матеріалів у вигляді набору окремих сторінок не забезпечує належного рівня структурованості, зручного адміністрування, багатомовного подання та оперативного оновлення інформації. Унаслідок цього виникає потреба у створенні веб-орієнтованої інформаційної системи, яка дозволяє централізовано керувати контентом міжнародного освітнього проєкту, підтримувати його актуальність і забезпечувати доступність інформації для різних категорій користувачів.

Особливість міжнародного освітнього проєкту полягає в тому, що його інформаційне середовище є багатокomпонентним. У межах одного ресурсу необхідно поєднати відомості про проєкт, партнерів, новини, події, інформаційні та навчальні матеріали, галерею, форми взаємодії з користувачами, а також багатомовне представлення контенту. За відсутності спеціалізованої системи така інформація зазвичай оновлюється фрагментарно, вимагає значних ручних зусиль і не забезпечує належної узгодженості між окремими розділами. Це знижує ефективність комунікації проєкту та ускладнює підтримку вебресурсу.

У межах даної роботи ставиться задача розроблення веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту, яка забезпечує структуроване подання інформації, централізоване адміністрування матеріалів, підтримку багатомовності, зручну публікацію новин і ресурсів, а також можливість подальшого супроводу та розширення функціональності. На відміну від шаблонного вебсайту, така система повинна бути орієнтована саме на контентні процеси: створення, редагування, публікацію, оновлення та керування інформаційними об'єктами.

Постановка задачі дослідження полягає у створенні системи, що поєднує дві взаємопов'язані складові. Перша складова – це публічна частина вебресурсу, призначена для представлення інформації про міжнародний освітній проєкт. Друга складова – адміністративне середовище, у межах якого виконується керування контентом, його структуризація та оновлення. Отже, задача має не лише прикладний, а й системний характер, оскільки передбачає узгоджену реалізацію моделі даних, логіки адміністрування, інтерфейсу користувача та засобів підтримки вебфункціональності.

Основними проблемами, що зумовлюють необхідність розроблення такої

системи, є:

- відсутність єдиного середовища для централізованого керування контентом;
- складність підтримки актуальності великої кількості інформаційних матеріалів;
- необхідність багатомовного подання інформації для міжнародної аудиторії;
- потреба в зручному розмежуванні публічної та адміністративної частин ресурсу;
- необхідність спрощення роботи редакторів і адміністраторів під час оновлення вмісту;
- потреба в сучасному вебрішенні, придатному до подальшого розвитку й підтримки.

Основною задачею дослідження є розроблення такої веб-орієнтованої інформаційної системи, яка забезпечує ефективне керування контентом міжнародного освітнього проєкту та створює єдине цифрове середовище для публікації, оновлення і представлення інформації.

Очікуваним результатом виконання роботи є реалізована веб-орієнтована інформаційна система керування контентом міжнародного освітнього проєкту, яка забезпечує зручне адміністрування інформаційних матеріалів, багатомовне представлення контенту, структуровану організацію даних і сучасний формат взаємодії користувачів із вебресурсом. Практичний результат повинен полягати у створенні працездатного програмного рішення, придатного для використання як цифрова платформа інформаційного супроводу міжнародного освітнього проєкту.

Таким чином, у підрозділі сформульовано постановку задачі дослідження, визначено основну проблему, окреслено головну задачу та завдання, які необхідно розв'язати в межах кваліфікаційної роботи.

2 ПРОЄКТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ КЕРУВАННЯ КОНТЕНТОМ

2.1 Формування вимог до системи та загальної архітектури

На основі аналізу предметної області та постановки задачі сформовано основні вимоги до веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту. Система повинна забезпечувати зручне подання інформації у публічній частині сайту та централізоване керування контентом через адміністративне середовище. Основна увага приділяється простоті наповнення ресурсу, багатомовності, структурованому зберіганню даних і зручності подальшого супроводу.

До функціональних вимог системи належать:

- відображення основних розділів сайту;
- публікація новин, сторінок, матеріалів, даних про партнерів і галереї;
- створення, редагування та видалення контенту через адміністративну панель;
- підтримка багатомовного інтерфейсу;
- переклад контенту після створення;
- динамічне підвантаження списків даних;
- робота форм зворотного зв'язку;
- розмежування прав доступу для адміністраторів і редакторів.

До нефункціональних вимог належать:

- зрозумілий і адаптивний інтерфейс;
- зручність адміністрування;
- швидке завантаження сторінок;
- цілісність зберігання даних;
- безпечне оброблення введеної інформації;
- можливість розгортання системи в уніфікованому середовищі;
- придатність до подальшого розширення.

Узагальнення вимог подано в таблиці 2.1.

Таблиця 2.1 – Основні вимоги до системи

Група вимог	Зміст
Інформаційні	Подання сторінок, новин, ресурсів, партнерів, галереї
Адміністративні	Створення, редагування, видалення контенту
Мовні	Підтримка кількох мов інтерфейсу і контенту
Інтерактивні	Динамічне підвантаження даних на сторінки
Безпекові	Контроль доступу, валідація даних
Технічні	Стабільність, адаптивність, можливість супроводу

Загальна архітектура системи побудована за багаторівневим принципом. Основу серверної частини становить Laravel 12, який забезпечує маршрутизацію, оброблення запитів, роботу з моделями та бізнес-логікою. Формування сторінок публічної частини виконується за допомогою Blade-шаблонів із використанням HTML, CSS і JavaScript. Для зберігання структурованих даних використовується MySQL. Адміністративну панель реалізовано засобами Filament v5, а динамічне підвантаження даних без повного перезавантаження сторінки забезпечується через Livewire 4. Розгортання системи здійснюється за допомогою Docker.

На рисунку 2.1 подано загальну архітектуру системи [28].

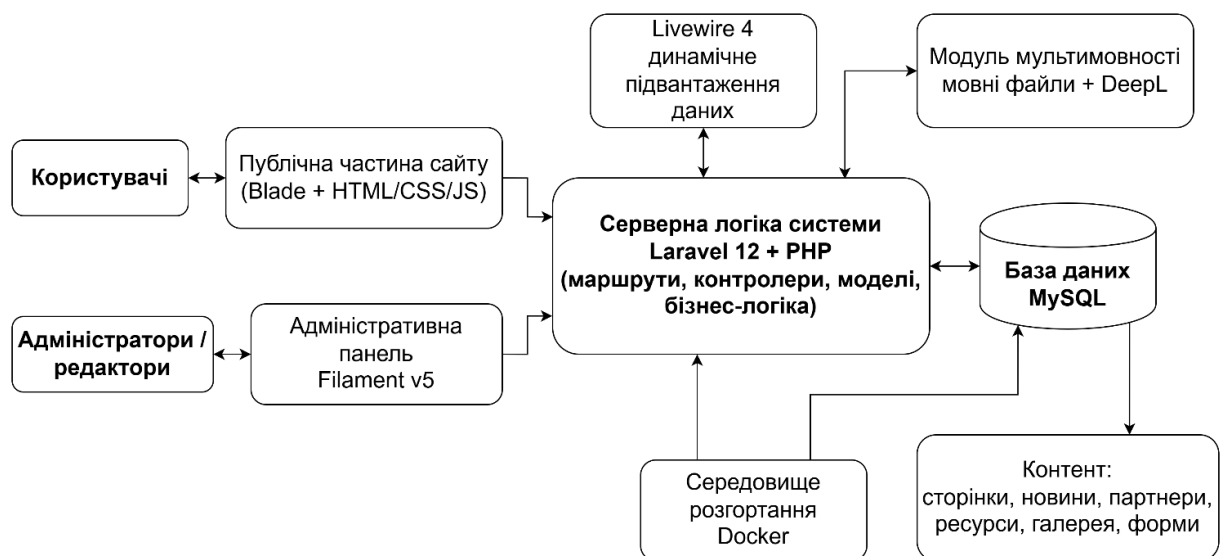


Рисунок 2.1 – Загальна архітектура веб-орієнтованої інформаційної системи

У структурі системи виділено такі основні компоненти [28]:

- публічна частина сайту – забезпечує доступ користувачів до інформаційних сторінок, новин, матеріалів і галереї;

- адміністративна панель – використовується для керування контентом;
- модуль мультимовності – відповідає за статичні переклади та переклад контенту;
- модуль динамічної взаємодії – забезпечує часткове оновлення сторінок;
- база даних – зберігає контентні сутності, користувачів і службову інформацію;
- середовище розгортання – забезпечує запуск системи в контейнеризованій інфраструктурі.

Отже, сформовані вимоги та загальна архітектура системи відображають основні потреби міжнародного освітнього проєкту й створюють основу для подальшого проєктування структури даних і функціональних модулів системи.

2.2 Проєктування структури даних і моделі контентних інформаційних об'єктів

Проєктування структури даних є одним із ключових етапів розроблення веб-орієнтованої інформаційної системи, оскільки саме на цьому рівні визначається, як зберігатиметься, пов'язуватиметься та відображатиметься контент. Для міжнародного освітнього проєкту структура даних повинна забезпечувати впорядковане подання інформації, підтримку багатомовності, зручне адміністрування та можливість подальшого розширення системи.

Основу моделі становлять інформаційні об'єкти, тобто структуровані типи контенту, з якими працює система. У межах розроблюваного рішення розроблено такі основні інформаційні об'єкти:

- Page – інформаційні сторінки;
- News – новини;
- Partner – партнери проєкту;
- Resource – матеріали та ресурси;
- GalleryItem – елементи галереї;
- Category – категорії;
- User – користувачі адміністративної панелі;

– ContactRequest – повідомлення із форм зворотного зв’язку.

Кожен із цих об’єктів має власне призначення. Page використовується для сталих сторінок сайту, News – для публікацій і новинних записів, Partner – для подання відомостей про організації-партнери, Resource – для документів і навчальних матеріалів, GalleryItem – для зображень і підписів до них. Такий поділ відповідає логіці реального контентного наповнення міжнародного освітнього проекту.

На рисунку 2.2 представлено ER-діаграму інформаційних об’єктів системи.

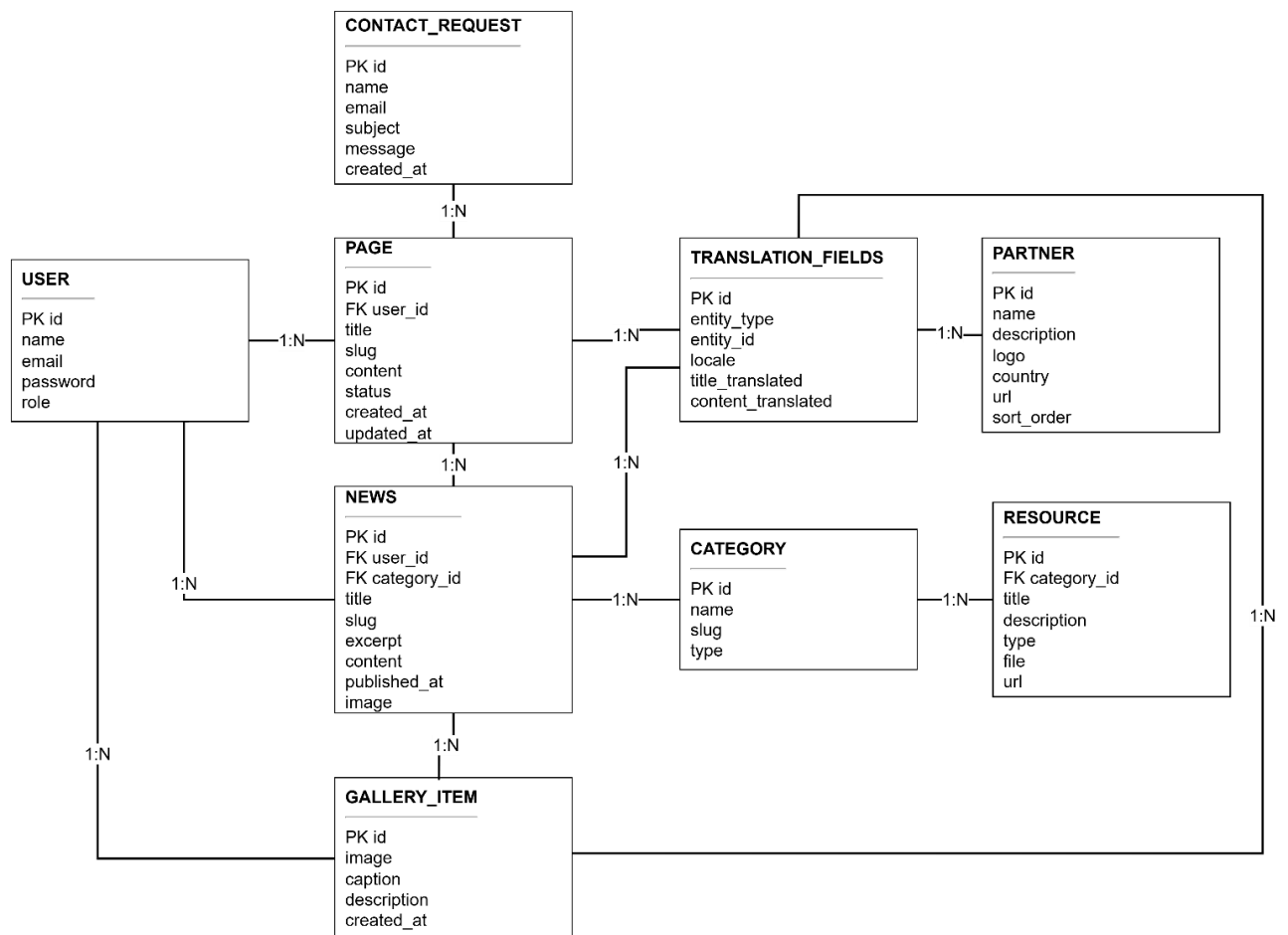


Рисунок 2.2 – Схема взаємозв’язків інформаційних об’єктів веб-орієнтованої інформаційної системи

Для кожного інформаційного об’єкта задається набір основних атрибутів. Наприклад, для сторінок і новин доцільно використовувати назву, slug, текстовий вміст, статус публікації, дату створення та дату оновлення. Для новин додатково використовуються дата публікації, короткий опис і зображення. Для партнерів

важливими є назва, опис, логотип, країна та вебпосилання. Для ресурсів – назва, тип, файл або URL, короткий опис. Для галереї – зображення, підпис і опис.

Узагальнену характеристику основних інформаційних об’єктів подано в таблиці 2.2.

Таблиця 2.2 – Основні інформаційні об’єкти системи

Інформаційний об’єкт	Призначення	Основні атрибути
Page	Збереження сталих сторінок	title, slug, content, status
News	Збереження новин	title, slug, excerpt, content, published_at, image
Partner	Збереження інформації про партнерів	name, description, logo, country, url
Resource	Збереження документів і матеріалів	title, description, type, file, url
GalleryItem	Збереження зображень галереї	image, caption, description
Category	Групування контенту	name, slug, type
User	Доступ до адміністративної панелі	name, email, password, role
ContactRequest	Збереження звернень користувачів	name, email, subject, message

Важливим елементом проєктування є визначення зв’язків між інформаційними об’єктами. У системі доцільно використовувати переважно зв’язки типу один до багатьох. Наприклад, один користувач може створювати багато новин або сторінок, а одна категорія може об’єднувати багато новин чи ресурсів. Такий підхід забезпечує логічну структуру моделі даних і не ускладнює реалізацію.

Основні зв’язки подано в таблиці 2.3.

Таблиця 2.3 – Основні зв’язки між інформаційними об’єктами системи

Базовий об’єкт	Пов’язаний об’єкт	Тип зв’язку	Призначення
User	News	один до багатьох	Визначення автора новини
User	Page	один до багатьох	Визначення автора сторінки
Category	News	один до багатьох	Групування новин
Category	Resource	один до багатьох	Групування ресурсів

Окрему увагу під час проєктування слід приділити багатомовності. У системі доцільно розмежувати статичні елементи інтерфейсу та контентні дані. Статичні написи меню, кнопок і службових повідомлень зберігаються у мовних файлах. Контент, створений в адміністративній панелі, перекладається після створення. Такий підхід є зручним, оскільки дозволяє окремо керувати інтерфейсом і змістовим наповненням сайту.

Для коректної роботи системи інформаційні об'єкти повинні містити також службові поля: статус публікації, slug, дату створення, дату оновлення, порядок сортування та ознаку активності. Саме ці поля забезпечують не лише зберігання контенту, а й його повноцінне адміністрування та відображення в публічній частині сайту.

Сформована структура даних відповідає обраному стеку технологій. У середовищі Laravel 12 кожен інформаційний об'єкт реалізується як окрема модель, пов'язана з таблицею в MySQL. Це спрощує створення CRUD-операцій в Filament v5, а також забезпечує зручне виведення даних у публічній частині сайту через Blade-шаблони. Крім того, така структура придатна для інтеграції з Livewire 4 під час динамічного підвантаження новин, матеріалів і галереї.

Отже, у підрозділі сформовано структуру даних і модель основних контентних інформаційних об'єктів системи. Визначено їх призначення, базові атрибути та основні зв'язки. Сформована модель є основою для подальшого проєктування функціональних модулів веб-орієнтованої інформаційної системи.

2.3 Проєктування функціональної структури системи

Після формування загальної архітектури та моделі даних наступним етапом є проєктування функціональних модулів веб-орієнтованої інформаційної системи. На цьому рівні система розглядається як сукупність взаємопов'язаних програмних компонентів, кожен із яких виконує окрему прикладну функцію. Такий підхід дозволяє забезпечити логічну структурованість розробки, спростити реалізацію й подальший супровід програмного продукту.

Для інформаційної системи керування контентом міжнародного освітнього

проекту доцільно виділити кілька основних функціональних модулів:

- модуль публічної частини сайту;
- модуль керування контентом;
- модуль адміністрування;
- модуль мультимовності;
- модуль динамічного підвантаження даних;
- модуль форм зворотного зв'язку;
- модуль доступу й безпеки.

На рисунку 2.3 подано функціональну схему модулів системи.



Рисунок 2.3 – Функціональна схема модулів веб-орієнтованої інформаційної системи

Центральне місце в системі займає модуль публічної частини сайту. Його призначення полягає у відображенні контенту для кінцевих користувачів. Через цей модуль забезпечується перегляд сторінок про проєкт, новин, відомостей про партнерів, галереї, ресурсів та інших розділів вебресурсу. Саме він формує зовнішній інтерфейс системи й забезпечує доступ до інформації без авторизації. Для реалізації цього модуля використовуються Blade-шаблони, маршрутизація Laravel і засоби відображення даних із бази MySQL.

Другим важливим компонентом є модуль керування контентом. Його функція полягає у створенні, редагуванні, оновленні, публікації та видаленні інформаційних матеріалів. На цьому рівні система працює з такими

інформаційними об'єктами, як сторінки, новини, партнери, ресурси, елементи галереї та категорії. Саме цей модуль забезпечує логіку життєвого циклу контенту. Його наявність відрізняє веб-орієнтовану інформаційну систему від звичайного статичного сайту.

Функції централізованого доступу до даних і керування контентом реалізуються через модуль адміністрування. У межах даної роботи його побудовано на основі Filament v5. Використання цього пакета дозволяє організувати зручне середовище для роботи адміністратора або редактора, реалізувати форми введення, таблиці перегляду записів, механізми фільтрації, сортування та редагування. Унаслідок цього значно спрощується керування великим обсягом інформації, а структура інтерфейсу залишається уніфікованою для різних типів контенту.

Окрему роль виконує модуль мультимовності. Для міжнародного освітнього проєкту цей модуль є обов'язковим, оскільки система повинна підтримувати кілька мов інтерфейсу та контенту. Його функції охоплюють відображення статичних перекладів у мовних файлах, перемикання мовних версій сторінок, а також переклад створеного контенту після його додавання. Таким чином забезпечується цілісне багатомовне середовище, у якому користувач отримує доступ до тієї самої інформації різними мовами.

Для покращення взаємодії користувача з контентом у системі виділено модуль динамічного підвантаження даних. Його призначення полягає в поступовому завантаженні інформації без повного перезавантаження сторінки. Це особливо важливо для новин, ресурсів або інших списків, що містять багато записів. У даній роботі така функціональність реалізується за допомогою Livewire 4. Завдяки цьому зменшується навантаження на інтерфейс, підвищується зручність користування сайтом і забезпечується сучасний формат взаємодії з контентом.

Не менш важливим є модуль форм зворотного зв'язку. Його завдання полягає в організації комунікації між користувачами та адміністрацією проєкту. Через цей модуль можуть передаватися запити, повідомлення, заявки або інші звернення. Його функціонування передбачає приймання даних із форми, перевірку коректності введення, збереження повідомлення в системі або передавання його на

електронну пошту. Такий модуль є важливим елементом комунікаційної підтримки міжнародного освітнього проєкту.

Для захисту адміністративного середовища та службових операцій до складу системи входить модуль доступу й безпеки. Його основними функціями є автентифікація користувачів, розмежування прав доступу, обмеження несанкціонованого втручання в систему та перевірка даних, що надходять із форм. Цей модуль забезпечує безпечну роботу адміністративної панелі та захищає критично важливі операції, пов'язані з керуванням контентом.

Проектування функціональних модулів доцільно виконувати з урахуванням їх взаємодії між собою. Наприклад, модуль публічної частини отримує дані з модуля керування контентом через рівень моделей і контролерів. Модуль адміністрування взаємодіє з тими самими інформаційними об'єктами, але в іншому режимі – через інтерфейс створення та редагування. Модуль мультимовності впливає як на публічну частину, так і на адміністративне середовище. Модуль динамічного підвантаження даних розширює можливості публічної частини сайту, а модуль безпеки охоплює адміністративну та комунікаційну частини системи.

З погляду програмної реалізації кожен функціональний модуль спирається на засоби обраного технологічного стеку. Laravel 12 забезпечує серверну логіку, маршрутизацію та роботу з моделями. Blade-шаблони відповідають за відображення даних у публічній частині. Filament v5 використовується для побудови адміністративного інтерфейсу. Livewire 4 забезпечує інтерактивність на рівні окремих сторінок. MySQL використовується для зберігання інформаційних об'єктів, а Docker – для організації середовища виконання системи.

Отже, у підрозділі визначено основні функціональні модулі веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту. Виділені модулі охоплюють публічне представлення інформації, адміністрування, мультимовність, інтерактивну взаємодію, зворотний зв'язок і безпеку.

2.4 Алгоритми роботи веб-орієнтованої інформаційної системи

Після визначення архітектури, структури даних і функціональних модулів доцільно описати алгоритми роботи веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту. На цьому етапі увага зосереджується не на програмній реалізації окремих компонентів, а на послідовності дій, які забезпечують функціонування системи в типових сценаріях використання. Такий підхід дозволяє формалізувати логіку роботи системи та підготувати основу для її подальшої реалізації й тестування.

Для розроблюваної системи доцільно виділити кілька основних алгоритмів:

- алгоритм відображення контенту в публічній частині сайту;
- алгоритм створення та публікації контенту в адміністративній панелі;
- алгоритм перекладу контенту;
- алгоритм динамічного підвантаження даних;
- алгоритм оброблення повідомлень із форм зворотного зв'язку.

Алгоритм відображення контенту в публічній частині сайту (рисунок 2.4) виконується щоразу, коли користувач відкриває сторінку вебресурсу. Його завдання полягає в отриманні актуальних даних із бази, їх опрацюванні та формуванні сторінки для відображення у браузері. Такий алгоритм є базовим для сторінок новин, інформаційних розділів, партнерів, ресурсів і галереї.

Алгоритм створення та публікації контенту (рисунок 2.5) реалізує роботу адміністратора або редактора в адміністративній панелі. Його основне призначення полягає у створенні нового запису, перевірці введених даних, збереженні їх у базі та подальшій публікації на сайті.

У разі редагування контенту, алгоритм є аналогічним, але передбачає завантаження вже наявного запису та його оновлення.

Алгоритм перекладу контенту подано на рисунку 2.6. Оскільки система орієнтована на міжнародний освітній проєкт, одним із важливих процесів є підтримка багатомовності. Для цього використовується комбінований підхід: статичні елементи інтерфейсу зберігаються у мовних файлах, а створений контент перекладається після додавання. Цей алгоритм дозволяє скоротити час на

підготовку багатомовного контенту та спростити супровід вебресурсу.

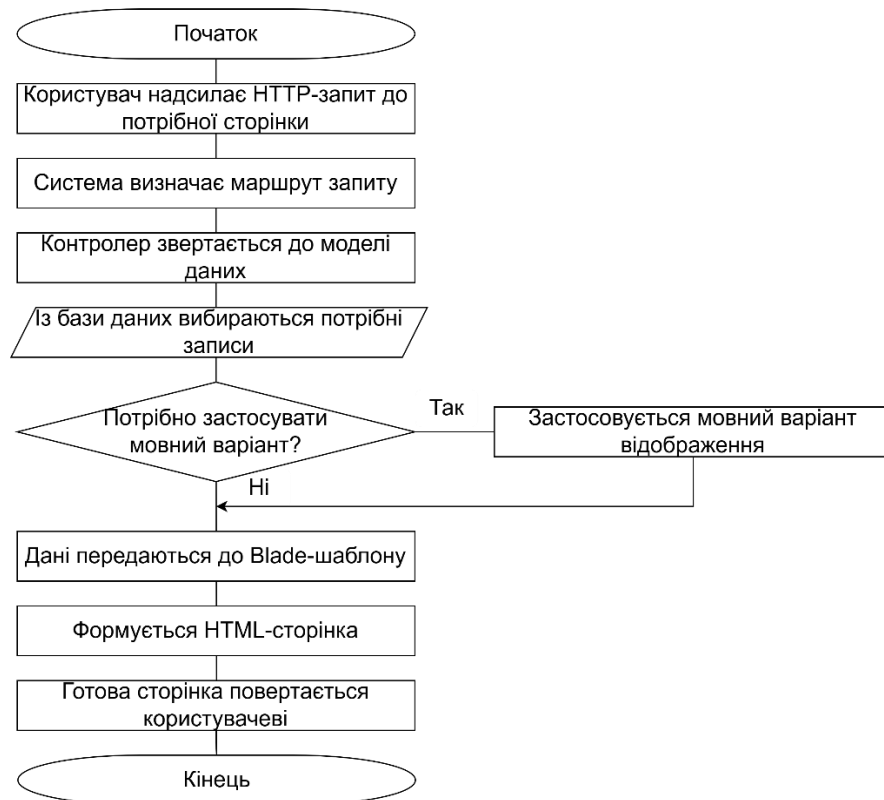


Рисунок 2.4 – Схема алгоритму відображення контенту в публічній частині сайту

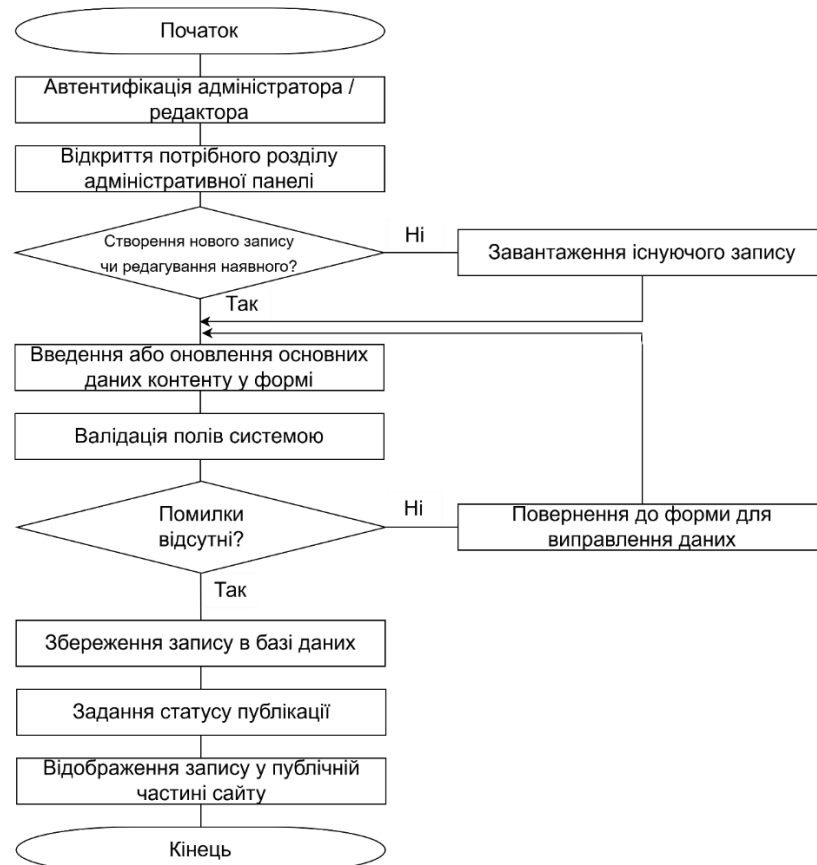


Рисунок 2.5 – Схема алгоритму створення та публікації контенту

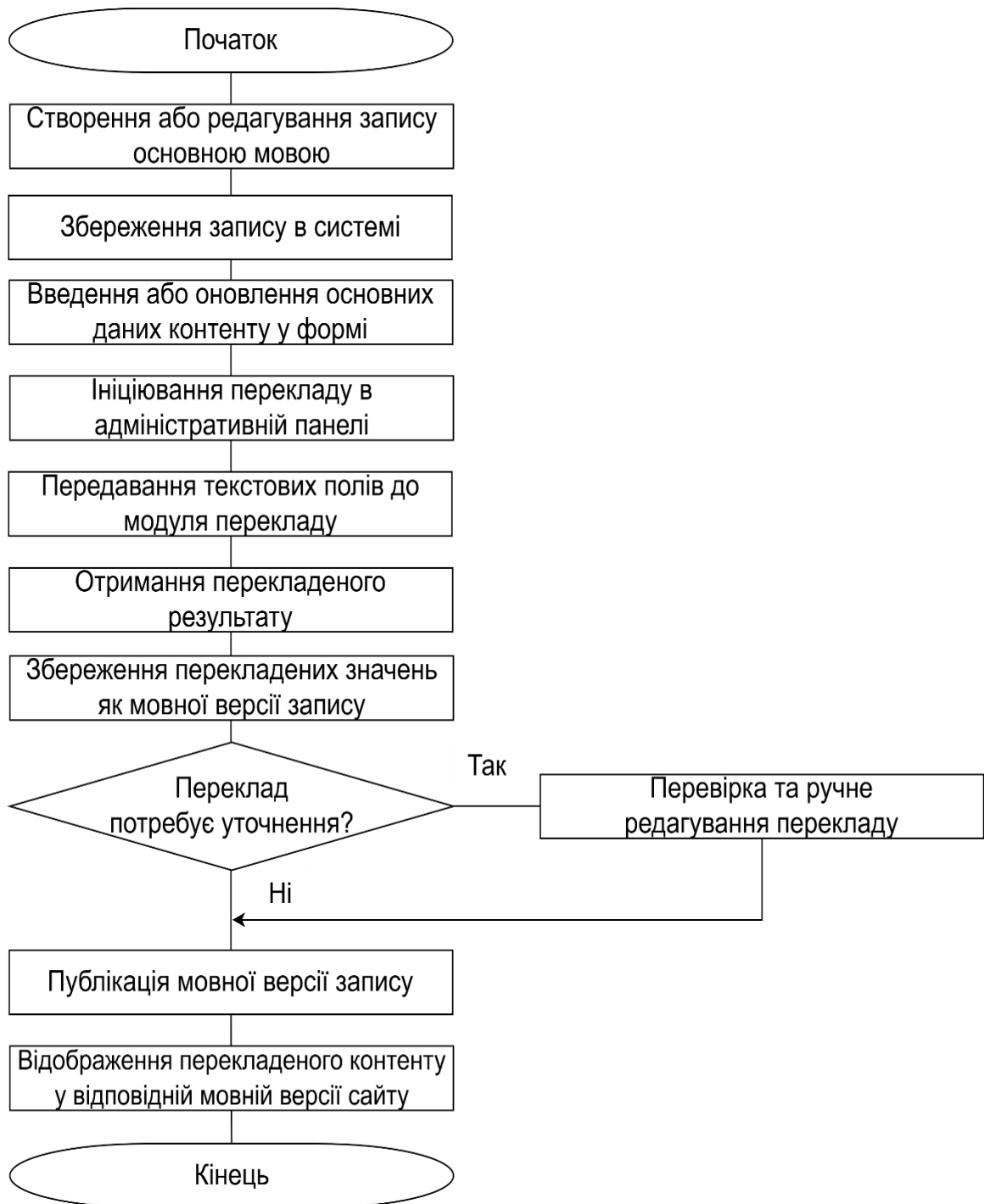


Рисунок 2.6 – Схема алгоритму перекладу контенту

Алгоритм динамічного підвантаження даних (рисунок 2.7) використовується для сторінок, на яких необхідно відображати великі списки записів, наприклад новини або ресурси. Його завдання полягає в тому, щоб не завантажувати всі записи одночасно, а передавати їх частинами. Такий алгоритм підвищує зручність користування сайтом і зменшує перевантаження сторінки великим обсягом контенту.



Рисунок 2.7 – Схема алгоритму динамічного підвантаження даних

Алгоритм оброблення повідомлень із форм зворотного зв'язку представлено на рисунку 2.8. Форми зворотного зв'язку забезпечують комунікацію між користувачами та адміністрацією проєкту. Алгоритм їх роботи передбачає приймання даних, перевірку коректності введення та подальше збереження або надсилання повідомлення.



Рисунок 2.8 – Схема алгоритму оброблення повідомлень із форм зворотного зв'язку

Отже, у підрозділі описано основні алгоритми роботи веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту. Виділено базові сценарії функціонування системи, пов'язані з відображенням контенту, його створенням, перекладом, динамічним підвантаженням і зворотним зв'язком.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Реалізація структури програмного проєкту та серверної частини системи

Програмну реалізацію веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту виконано з використанням фреймворку Laravel 12 та мови програмування PHP. Обраний підхід дозволив побудувати систему як структурований вебзастосунок із розділенням серверної логіки, представлення даних, адміністративного керування та роботи з базою даних. Така організація відповідає загальним принципам побудови сучасних вебсистем, у яких контент зберігається у структурованій формі, а його відображення формується динамічно залежно від запиту користувача [12, 20].

Реалізована система є основою сайту міжнародного освітнього проєкту YouthSOFTPOWER. Її публічна частина забезпечує доступ до інформаційних сторінок, новин, партнерів, ресурсів, галереї та навчальних розділів. Серверна частина відповідає за маршрутизацію запитів, отримання даних із бази, передавання їх у шаблони, оброблення форм, перевірку введених даних і взаємодію з адміністративною панеллю.

Загальну структуру програмної реалізації подано на рисунку 3.1.

Структура програмного проєкту сформована відповідно до логіки Laravel-застосунку. У ній виділено каталоги для моделей, контролерів, шаблонів, маршрутів, конфігураційних файлів, міграцій бази даних, ресурсів інтерфейсу та службових компонентів. Такий поділ спрощує супровід системи, оскільки кожен тип програмної логіки розміщується у відповідній частині проєкту.

У таблиці 3.1 наведено основні каталоги та файли програмної реалізації.

Як видно з таблиці 3.1, структура проєкту поділена на логічні рівні. Моделі відповідають за роботу з даними, контролери – за оброблення запитів, шаблони – за формування сторінок, а окремі каталоги Filament і Livewire забезпечують адміністративне керування та інтерактивну взаємодію з користувачем.

Серверна частина системи реалізована за принципом розділення відповідальності між маршрутом, контролером, моделлю та представленням.

Користувач надсилає HTTP-запит до певної сторінки. Далі система визначає відповідний маршрут, викликає контролер, отримує дані з моделі та передає їх у Blade-шаблон. Після цього формується HTML-сторінка, яка повертається користувачеві. Такий підхід забезпечує керованість логіки вебзастосунку та відповідає загальній моделі функціонування HTTP-взаємодії [13].

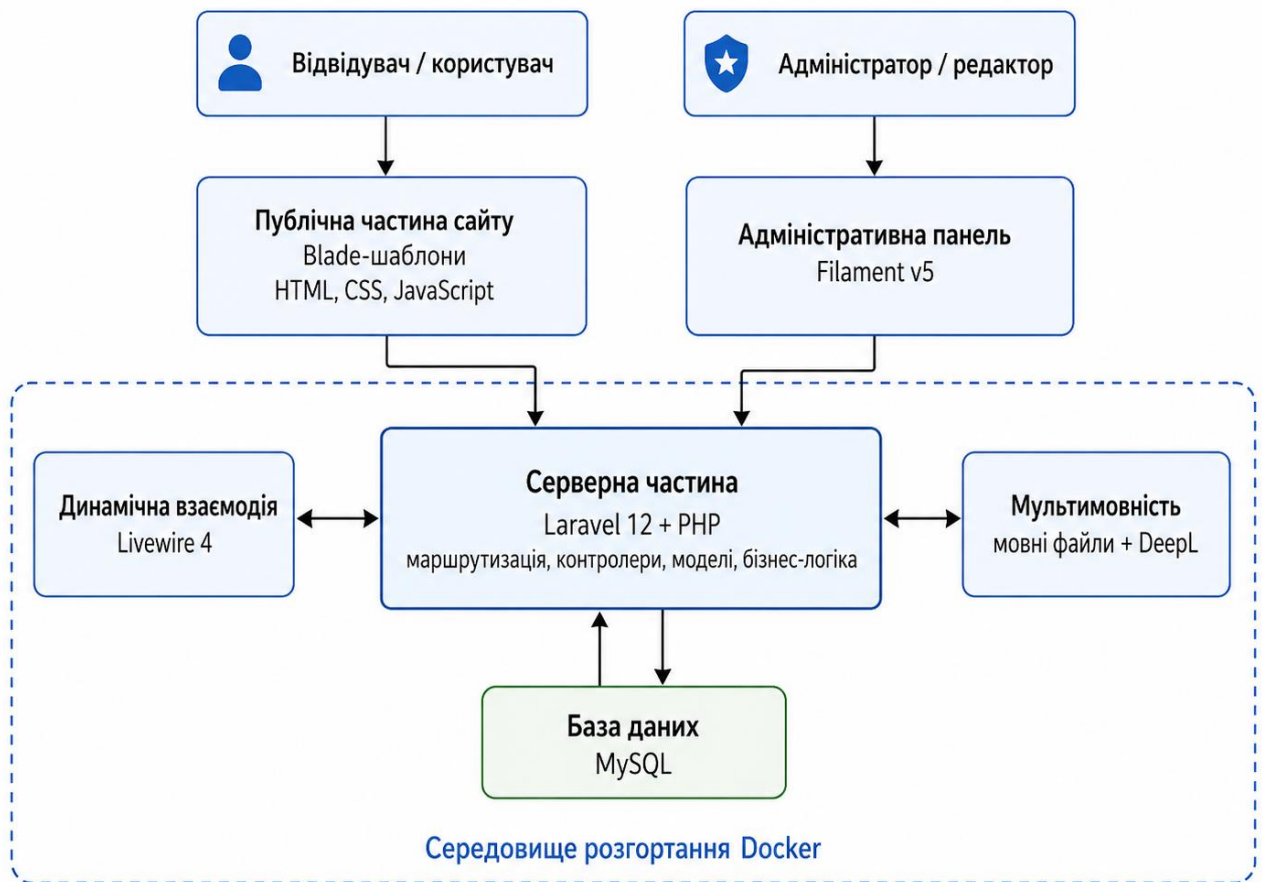


Рисунок 3.1 – Узагальнена структурна схема програмної реалізації веб-орієнтованої інформаційної системи

Таблиця 3.1 – Основні каталоги та файли програмної реалізації

Каталог або файл	Призначення
app/Models	Збереження моделей інформаційних об’єктів системи
app/Http/Controllers	Оброблення HTTP-запитів і передавання даних у представлення
app/Filament	Реалізація адміністративної панелі на основі Filament v5
app/Livewire	Компоненти для динамічного підвантаження даних

Продовження таблиці 3.1

Каталог або файл	Призначення
resources/views	Blade-шаблони публічної частини сайту
resources/lang	Мовні файли для статичних елементів інтерфейсу
routes/web.php	Опис маршрутів публічної частини системи
database/migrations	Структура таблиць бази даних
public	Публічні ресурси: зображення, стилі, скрипти
docker-compose.yml	Конфігурація контейнерного середовища
.env	Параметри середовища, зокрема підключення до бази даних

Приклад узагальненого маршруту для сторінки новин наведено в лістингу 3.1.

Лістинг 3.1 – Приклад маршруту для відображення новин

```
Route::get('/news', [NewsController::class, 'index'])
    ->name('news.index');

Route::get('/news/{slug}', [NewsController::class, 'show'])
    ->name('news.show');
```

У наведеному фрагменті перший маршрут відповідає за відкриття списку новин, а другий – за перегляд окремої новини за її slug. Повний програмний код маршрутів, контролерів і моделей подано в додатку А.

Для роботи з контентом у системі створено набір моделей, які відповідають основним інформаційним об'єктам: сторінкам, новинам, партнерам, ресурсам, галереї, категоріям і повідомленням із форм зворотного зв'язку. Кожна модель пов'язана з відповідною таблицею бази даних MySQL. На рівні моделей визначено атрибути, які можуть заповнюватися через адміністративну панель, а також зв'язки між об'єктами даних.

У таблиці 3.2 подано відповідність основних моделей системи їх призначенню.

Таблиця 3.2 – Основні моделі серверної частини системи

Модель	Призначення
Page	Збереження сталих інформаційних сторінок
News	Збереження новин і подій проєкту
Partner	Збереження інформації про партнерські організації
Resource	Збереження матеріалів і документів
GalleryItem	Збереження елементів галереї
Category	Групування новин і ресурсів
User	Користувачі адміністративної панелі
ContactRequest	Збереження повідомлень із форм зворотного зв'язку

Використання таких моделей дозволило уникнути хаотичного зберігання контенту та представити кожен тип інформації як окремий структурований об'єкт. Це важливо для системи керування контентом, оскільки адміністратор працює не з окремими HTML-сторінками, а з логічними записами, які надалі автоматично відображаються у відповідних розділах сайту [12, 20].

Для зберігання даних використано реляційну базу даних MySQL. У ній зберігаються записи сторінок, новин, партнерів, ресурсів, категорій, користувачів та службова інформація. Реляційна структура є зручною для цієї системи, оскільки контент має чіткі зв'язки: новини можуть належати до категорій, ресурси можуть групуватися за типами, а записи мають авторів, статуси публікації та мовні варіанти. Такий підхід забезпечує цілісність даних і спрощує подальше розширення системи.

У серверній частині також реалізовано базові службові поля, необхідні для керування контентом. До них належать:

- id – унікальний ідентифікатор запису;
- title або name – назва інформаційного об'єкта;
- slug – людиночитабельний ідентифікатор для URL;
- content або description – основний текстовий вміст;
- status або is_active – ознака доступності запису;
- created_at і updated_at – дати створення та оновлення;

– published_at – дата публікації для новин.

Приклад спрощеної структури моделі новини наведено в лістингу 3.2.

Лістинг 3.2 – Приклад структури моделі новини:

```
class News extends Model
{
    protected $fillable = [
        'title',
        'slug',
        'excerpt',
        'content',
        'image',
        'published_at',
        'is_active'
    ];
}
```

У лістингу 3.2 показано тільки фрагмент моделі. Він ілюструє набір полів, які можуть заповнюватися під час створення або редагування новини. Повну реалізацію моделі, включно зі зв'язками, методами вибірки та додатковими налаштуваннями подано в додаток А.

Формування сторінок публічної частини виконано за допомогою Blade-шаблонів. Такий підхід дозволив створити спільні макети для всіх сторінок сайту, повторно використовувати блоки меню, футера, мовного перемикача, карток новин і партнерів. У результаті забезпечено єдність структури інтерфейсу та спрощено підтримку публічної частини системи.

У таблиці 3.3 наведено основні типи Blade-шаблонів, використаних у системі.

Таблиця 3.3 – Основні типи Blade-шаблонів публічної частини

Тип шаблону	Призначення
Основний макет	Формування загальної структури сторінки
Шаблон головної сторінки	Відображення ключової інформації про проєкт
Шаблон списку новин	Відображення новин у вигляді карток
Шаблон детальної новини	Відображення повного тексту новини
Шаблон партнерів	Відображення переліку партнерських організацій
Шаблон сторінки партнера	Відображення детальної інформації про партнера
Шаблон ресурсів і галереї	Відображення відповідних розділів сайту

Blade-шаблони взаємодіють із даними, отриманими через контролери. Наприклад, контролер новин вибирає активні записи з бази даних, сортує їх за датою публікації та передає у шаблон списку. У шаблоні ці записи відображаються як набір карток із заголовком, датою, зображенням і посиланням на повний текст.

Короткий приклад виведення списку новин у Blade-шаблоні наведено в лістингу 3.3.

Лістинг 3.3 – Приклад виведення новин у Blade-шаблоні

```
@foreach($news as $item)
    <article class="news-card">
        title
    }}">
        <h3>{{ $item->title }}</h3>
        <p>{{ $item->excerpt }}</p>
        <a href="{{ route('news.show', $item->slug) }}">Read
More</a>
    </article>
@endforeach
```

Наведений фрагмент демонструє принцип динамічного формування списку новин. Публічна сторінка не створюється вручну для кожної новини, а формується автоматично на основі записів із бази даних. Це підтверджує, що розроблена система виконує функції керування контентом, а не лише відображає статичні HTML-сторінки.

Для адміністративного керування системою використано Filament v5. Його застосування дозволило реалізувати форми створення та редагування записів, таблиці перегляду контенту, фільтрацію, сортування та базові CRUD-операції. У підрозділі 3.3 адміністративну панель буде розглянуто детальніше, однак уже на рівні структури проєкту важливо зазначити, що Filament інтегровано у Laravel-застосунок як окремий адміністративний рівень.

Для інтерактивного підвантаження даних використано Livewire 4. Його компоненти дають змогу частково оновлювати сторінки без повного перезавантаження. У межах системи такий підхід доцільний для сторінок зі списками новин, матеріалів або галереї. Це покращує зручність користування та зменшує обсяг даних, які завантажуються під час першого відкриття сторінки.

Розгортання програмної системи виконано за допомогою Docker.

Контейнеризація дозволила уніфікувати середовище виконання, зокрема роботу вебзастосунку, бази даних і супровідних сервісів. Для кваліфікаційної роботи це важливо, оскільки така організація спрощує перенесення системи між середовищами розробки, тестування та розгортання.

У таблиці 3.4 подано основні компоненти технологічного стеку, використані у програмній реалізації.

Таблиця 3.4 – Компоненти технологічного стеку програмної реалізації

Компонент	Технологія	Роль у системі
Backend	Laravel 12 + PHP	Серверна логіка, маршрутизація, моделі, контролери
Frontend	Blade, HTML, CSS, JavaScript	Формування сторінок публічної частини
База даних	MySQL	Зберігання структурованого контенту
Адміністрування	Filament v5	Керування записами через адміністративну панель
Інтерактивність	Livewire 4	Динамічне підвантаження даних
Мультимовність	Мовні файли, DeepL	Переклад інтерфейсу та контенту
Розгортання	Docker	Контейнеризоване середовище виконання

Як видно з таблиці 3.4, програмна реалізація побудована як поєднання кількох взаємопов'язаних технологічних компонентів. Laravel забезпечує основу серверної логіки, Blade відповідає за формування інтерфейсу, MySQL зберігає контент, Filament забезпечує адміністрування, Livewire реалізує інтерактивність, а Docker підтримує стабільне середовище розгортання.

Отже, у межах підрозділу описано структуру програмного проєкту та серверну частину веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту. Реалізацію побудовано на основі Laravel 12, Blade-шаблонів, MySQL, Filament v5, Livewire 4 і Docker.

3.2 Реалізація публічної частини сайту

Публічну частину веб-орієнтованої інформаційної системи реалізовано як відкритий інтерфейс взаємодії користувачів із матеріалами міжнародного освітнього проєкту YouthSOFTPOWER. Її основне призначення полягає у представленні загальної інформації про проєкт, публікації новин, поданні відомостей про партнерів, розміщенні ресурсів, галереї, навчальних розділів і засобів залучення користувачів до активностей проєкту.

На головній сторінці сайту реалізовано навігаційне меню, перемикач мов, презентаційний блок, опис мети проєкту, блок цілей, галерею, останні новини, перелік партнерів, контактну інформацію та посилання на соціальні мережі. У меню передбачено розділи Learning Platform, About, News, Partners, Resources, Gallery, Get Involved, а в підменю навчальної платформи – Courses, Materials, Quizzes, Pilots & Events. Також реалізовано мовні варіанти EN, ES, UK, EL, TR, FI, що відповідає міжнародному характеру проєкту.

На рисунку 3.2 подано головну сторінку веб-орієнтованої інформаційної системи.



Рисунок 3.2 – Головна сторінка веб-орієнтованої інформаційної системи YouthSOFTPOWER

Головна сторінка виконує функцію центральної точки входу до системи. Вона не лише інформує користувача про призначення проєкту, а й забезпечує

швидкий перехід до основних модулів сайту. Зокрема, на ній розміщено заклик “Empower Youth. Shape Global Impact.”, короткий опис ідеї проєкту, кнопку переходу до навчальної платформи, блок цілей, останні новини та перелік партнерських організацій. Це підтверджує, що головна сторінка побудована як інтегрований шаблон, у якому поєднано кілька типів контенту: текстові блоки, зображення, новини, партнерські записи й навігаційні елементи.

У таблиці 3.5 наведено основні сторінки публічної частини системи та їх призначення.

Таблиця 3.5 – Реалізовані сторінки публічної частини сайту

Сторінка	Призначення
Home	Загальне представлення проєкту, новини, партнери, галерея
About	Опис проєкту, місії, цільових груп і завдань
News	Перелік новин і подій проєкту
News detail	Детальний перегляд окремої новини
Partners	Перелік партнерських організацій
Partner detail	Детальний опис партнера
Resources	Розділ для матеріалів і результатів проєкту
Gallery	Розділ для візуальних матеріалів
Courses	Розділ навчальних курсів
Materials	Розділ навчальних матеріалів
Quizzes	Розділ тестових завдань
Pilots & Events	Розділ пілотних заходів і подій
Get Involved	Сторінка залучення користувачів і комунікації

Публічна частина сайту охоплює як уже наповнені інформаційні розділи, так і підготовлені сторінки для подальшого розширення навчального та ресурсного контенту. Такий підхід є доцільним для міжнародного освітнього проєкту, оскільки контент платформи формується поступово відповідно до етапів реалізації проєктних активностей.

Окреме місце в публічній частині займає новинний модуль. Його реалізовано

у вигляді списку новин із датами публікації, зображеннями, заголовками та посиланнями Read More. На головній сторінці відображаються останні новини, зокрема записи з датами 01.02.2026, 24.11.2025 і 29.10.2025. Це свідчить про те, що новини зберігаються як окремі інформаційні об'єкти та можуть автоматично виводитися в потрібних частинах сайту.

На рисунку 3.3 подано сторінку списку новин.

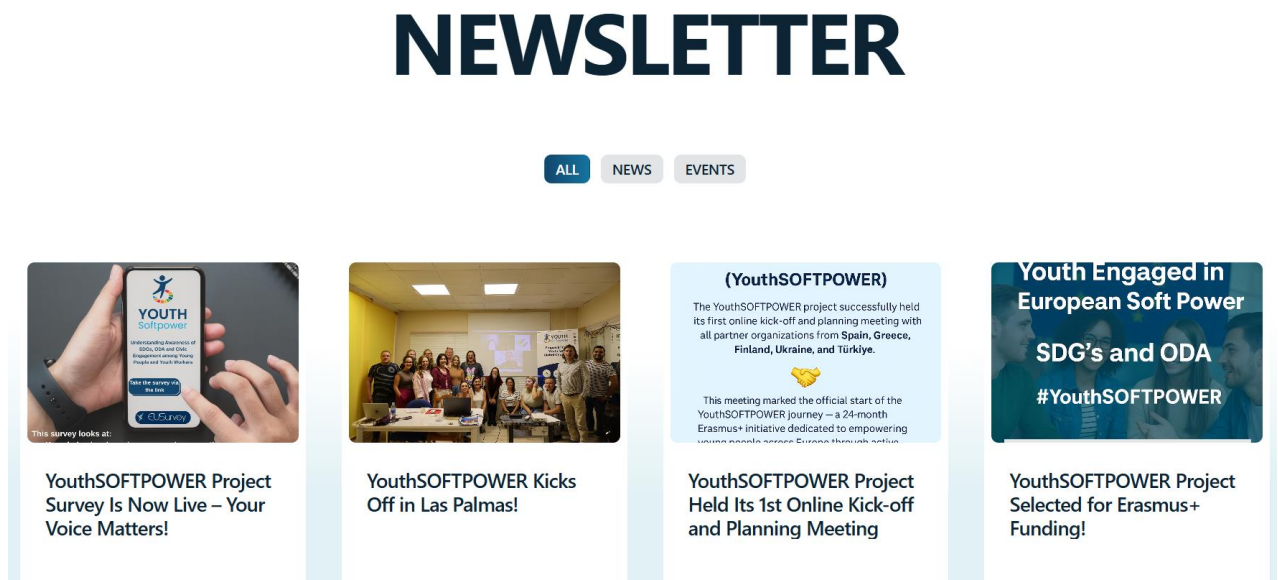


Рисунок 3.3 – Сторінка новин веб-орієнтованої інформаційної системи

Реалізація новинного модуля передбачає два сценарії використання. Перший сценарій полягає у перегляді списку новин, де користувач бачить коротку інформацію про публікації. Другий сценарій передбачає перехід до детальної сторінки новини. Такий підхід є зручним, оскільки дозволяє не перевантажувати сторінку списку великим обсягом тексту, а повний матеріал відкривати лише за потреби.

Для виведення новин у публічній частині використано Blade-шаблони. Дані новин передаються з контролера до шаблону, після чого формуються картки записів. Спрощений приклад такого виведення наведено в лістингу 3.4.

Лістинг 3.4 – Фрагмент Blade-шаблону для виведення карток новин

```
@foreach($news as $item)
    <article class="news-card">
        title
    }}">
```

```

<span>{{ $item->published_at->format('d.m.Y') }}</span>
<h3>{{ $item->title }}</h3>
<a href="{ route('news.show', $item->slug) }}">Read
More</a>
</article>
@endforeach

```

У лістингу 3.4 показано тільки короткий фрагмент коду, який ілюструє принцип динамічного формування новинної стрічки. Повний код шаблонів, контролерів і моделей подано в додатку А.

Наступним важливим елементом публічної частини є модуль партнерів. На головній сторінці відображено перелік партнерських організацій, серед яких Consulta Europa Projects and Innovation S.L., University of La Laguna, West Ukrainian National University, Liofyllo Social Cooperative Enterprise, ECO-IN, Keuropa Ry. Кожен партнер представлений як окремий інформаційний об'єкт, що може містити назву, опис, зображення, логотип, країну та зовнішнє посилання.

На рисунку 3.4 подано сторінку партнерів міжнародного освітнього проєкту.

PARTNERS OF THE PROJECT



Рисунок 3.4 – Сторінка партнерів міжнародного освітнього проєкту

Модуль партнерів є важливим для міжнародного освітнього проєкту, оскільки забезпечує прозоре представлення організацій, які беруть участь у реалізації проєкту. У межах системи цей модуль пов'язаний із моделлю Partner, таблицею бази даних і Blade-шаблонами для списку та детального перегляду. Такий підхід дозволяє адміністратору змінювати інформацію про партнерів без

редагування HTML-коду сторінок.

Публічна частина також містить навчальну платформу, що включає сторінки Courses, Materials, Quizzes, Pilots & Events. У поточній реалізації ці розділи мають підготовлену навігаційну структуру та можуть використовуватися для подальшого наповнення навчальним контентом. Їх наявність у меню підтверджує, що система спроектована не лише як інформаційний сайт, а як основа для розширення до освітньої платформи.

У таблиці 3.6 наведено відповідність публічних модулів основним інформаційним об'єктам системи.

Таблиця 3.6 – Відповідність публічних модулів інформаційним об'єктам системи

Публічний модуль	Інформаційний об'єкт	Спосіб відображення
Home	Page, News, Partner, GalleryItem	Змішане представлення блоками
About	Page	Стала інформаційна сторінка
News	News, Category	Список записів і детальна сторінка
Partners	Partner	Картки партнерів і детальний перегляд
Resources	Resource, Category	Сторінка майбутніх матеріалів
Gallery	GalleryItem	Галерейне представлення
Learning Platform	Course, Material, Quiz, Event	Підготовлена структура навчальних розділів
Get Involved	ContactRequest	Комунікаційна сторінка і форма

Як показано в таблиці 3.6, публічна частина сайту не є набором ізольованих статичних сторінок. Вона побудована на основі інформаційних об'єктів, які зберігаються в базі даних і відображаються через відповідні шаблони. Це забезпечує гнучкість оновлення контенту та можливість розширення структури

сайту.

Окрему роль у публічній частині виконує сторінка Get Involved. Вона призначена для залучення користувачів до активностей проєкту, комунікації з командою та переходу до каналів зв'язку. На головній сторінці також розміщено контактну адресу youthsoftpowerproject@gmail.com і посилання на соціальні мережі, що підтримує комунікаційну функцію системи.

На рисунку 3.5 доцільно подати сторінку залучення користувачів.

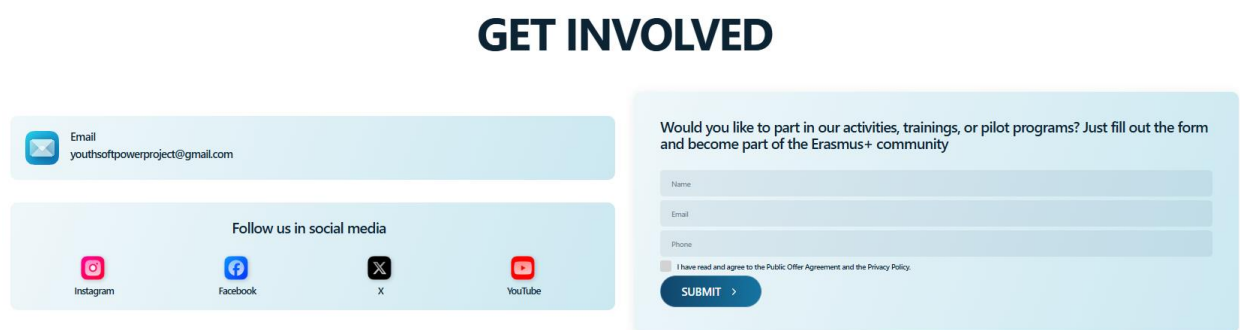


Рисунок 3.5 – Сторінка Get Involved як засіб комунікації з користувачами

Для публічної частини важливою є також багатомовність. На сайті реалізовано перемикач мов із варіантами EN, ES, UK, EL, TR, FI. Це дозволяє адаптувати інтерфейс до потреб різних груп користувачів і відповідає специфіці міжнародного освітнього проєкту. Мовна навігація розміщена в інтерфейсі сайту й забезпечує доступ до різних мовних версій публічних матеріалів.

Реалізована публічна частина відповідає основним вимогам, сформованим у другому розділі. Вона забезпечує:

- представлення загальної інформації про проєкт;
- перегляд новин і подій;
- подання партнерських організацій;
- підготовлену структуру ресурсів і навчальної платформи;
- комунікацію з користувачами;
- багатомовну навігацію;
- повторне використання контентних блоків у різних частинах сайту.

Узагальнену характеристику реалізації публічної частини подано в таблиці 3.7.

Таблиця 3.7 – Оцінка реалізації публічної частини системи

Критерій	Реалізація
Навігація	Реалізовано головне меню та підменю Learning Platform
Головна сторінка	Реалізовано презентаційний, новинний, партнерський і контактний блоки
Новини	Реалізовано список і перехід до повної новини
Партнери	Реалізовано перелік партнерів і можливість деталізації
Ресурси й навчальні розділи	Підготовлено структуру для подальшого наповнення
Багатомовність	Реалізовано перемикач мов EN, ES, UK, EL, TR, FI
Комунікація	Реалізовано сторінку Get Involved, email і соціальні посилання

Отже, у межах підрозділу описано реалізацію публічної частини веб-орієнтованої інформаційної системи YouthSOFTPOWER. Встановлено, що система забезпечує структуроване представлення міжнародного освітнього проєкту, підтримує основні інформаційні розділи, новини, партнерів, навчальну навігацію, комунікаційні елементи та багатомовний інтерфейс.

3.3 Реалізація адміністративної панелі та керування контентом

Адміністративну частину веб-орієнтованої інформаційної системи реалізовано як окреме середовище для керування контентом міжнародного освітнього проєкту. Її призначення полягає у створенні, редагуванні, публікації, оновленні та видаленні інформаційних матеріалів без прямого редагування програмного коду. Це дає змогу адміністратору або редактору наповнювати сайт через зручний інтерфейс.

Для реалізації адміністративної панелі використано Filament v5, що інтегрується з Laravel-застосунком і забезпечує швидке створення інтерфейсів для роботи з моделями даних. У системі через адміністративну панель передбачено керування сторінками, новинами, партнерами, ресурсами, елементами галереї, категоріями та повідомленнями, отриманими через форми зворотного зв'язку.

На рисунку 3.6 подано загальну схему керування контентом через адміністративну панель.

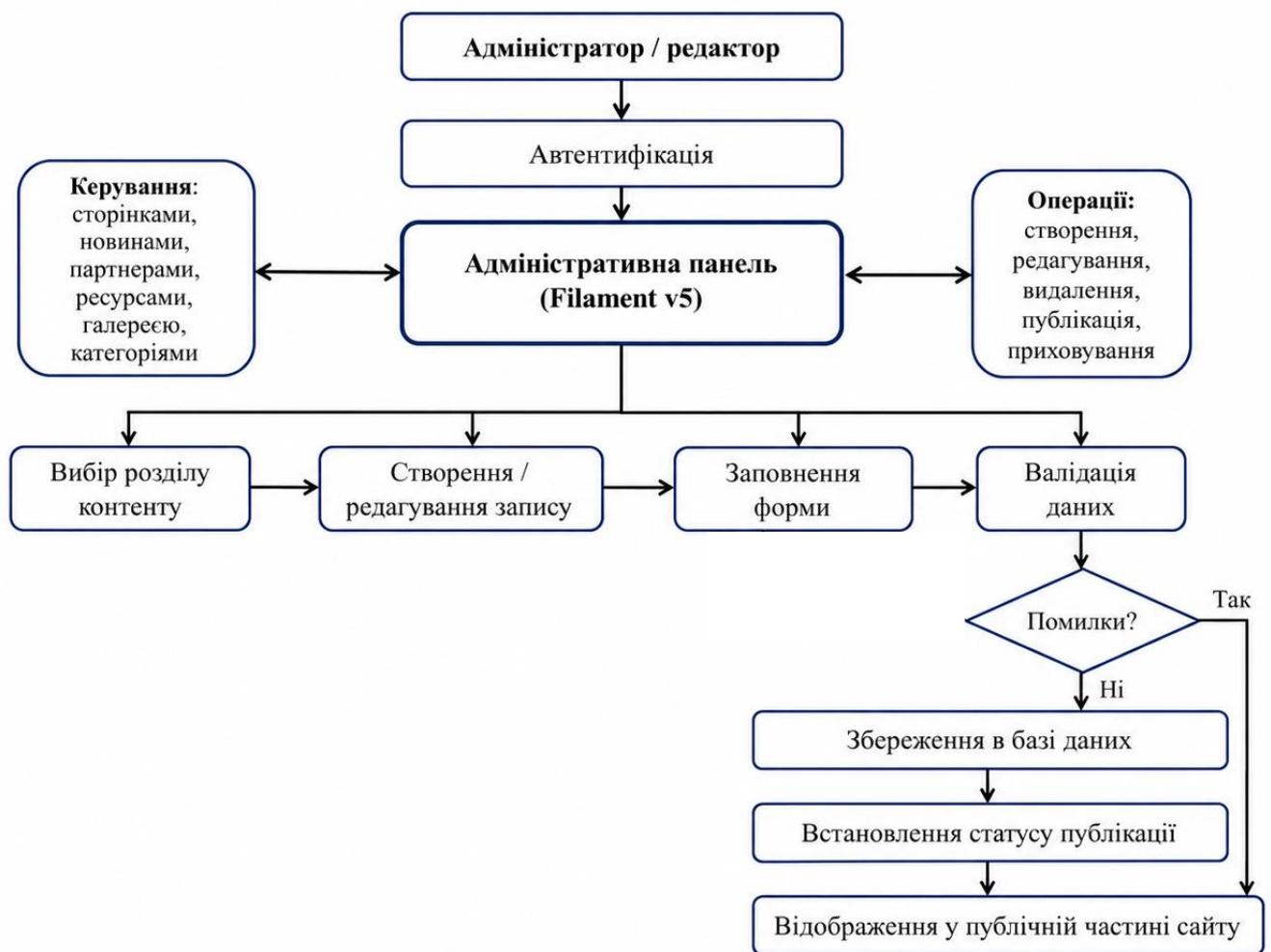


Рисунок 3.6 – Схема керування контентом через адміністративну панель

У процесі адміністрування користувач проходить автентифікацію, відкриває потрібний розділ панелі, створює або редагує запис, заповнює форму, перевіряє введені дані та задає статус публікації. Після збереження запис потрапляє до бази даних і, залежно від статусу, відображається у публічній частині сайту.

Основні операції адміністративної панелі наведено в таблиці 3.8.

Як видно з таблиці 3.8, адміністративна панель охоплює базові CRUD-операції, необхідні для роботи із системою керування контентом. Це дозволяє відокремити процес наповнення сайту від програмної реалізації та зробити оновлення інформації доступним для користувачів із відповідними правами.

Таблиця 3.8 – Основні операції адміністративної панелі

Операція	Призначення	Результат
Створення запису	Додавання нового матеріалу	Запис збережено в базі даних
Редагування	Оновлення наявного контенту	Дані змінено
Видалення	Прибирання неактуального матеріалу	Запис вилучено або приховано
Публікація	Надання доступу до матеріалу	Контент доступний користувачам
Приховування	Тимчасове вимкнення відображення	Контент не показується на сайті
Фільтрація і сортування	Зручний пошук та упорядкування	Адміністратор швидше знаходить потрібні записи

Для кожного типу контенту створено окремий адміністративний розділ. Наприклад, розділ новин забезпечує додавання заголовка, короткого опису, основного тексту, зображення, дати публікації, категорії та статусу активності. Розділ партнерів використовується для збереження назви організації, опису, логотипу, країни та зовнішнього посилання.

У таблиці 3.9 подано відповідність адміністративних розділів основним інформаційним об'єктам системи.

На основі таблиці 3.9 видно, що адміністративне середовище побудовано відповідно до моделі даних, описаної в розділі 2. Кожен основний інформаційний об'єкт має власний розділ керування, що спрощує роботу адміністратора та забезпечує логічну організацію контенту.

Важливою складовою адміністративної панелі є форми створення та редагування записів. Вони містять лише ті поля, які потрібні для конкретного типу

контенту. Для новини це заголовок, slug, короткий опис, основний текст, зображення, дата публікації та статус активності. Для партнера – назва, опис, логотип, країна та зовнішнє посилання.

Таблиця 3.9 – Адміністративні розділи для керування контентом

Адміністративний розділ	Інформаційний об’єкт	Основні дії
Pages	Інформаційні сторінки	Створення, редагування, публікація
News	Новини та події	Додавання, редагування, вибір статусу
Partners	Партнерські організації	Додавання опису, логотипу, посилання
Resources	Матеріали та документи	Завантаження файлів, додавання опису
Gallery	Зображення галереї	Додавання зображень і підписів
Categories	Категорії	Групування новин і ресурсів
Contact Requests	Звернення користувачів	Перегляд отриманих повідомлень
Users	Користувачі панелі	Керування адміністраторами та редакторами

Короткий приклад опису форми для адміністративного ресурсу новин наведено в лістингу 3.5.

Лістинг 3.5 – Фрагмент опису форми адміністративного ресурсу новин

```

TextInput::make('title')
    ->label('Title')
    ->required()
    ->maxLength(255);

Textarea::make('excerpt')
    ->label('Short description')
    ->rows(3);

RichEditor::make('content')
    ->label('Content')
    ->required();

```

```
Toggle::make('is_active')  
->label('Published');
```

У лістингу 3.5 наведено лише скорочений приклад опису полів форми. Повний програмний код адміністративних ресурсів доцільно винести в додаток А.

Для забезпечення коректності введених даних у системі застосовано валідацію. Вона використовується для перевірки обов'язкових полів, довжини тексту, правильності введення URL, наявності зображення або файлу, а також статусу публікації.

У системі також передбачено керування статусами публікації. Запис може бути чернеткою, активним, неактивним або запланованим. Активні записи відображаються у публічній частині сайту, тоді як чернетки та неактивні матеріали залишаються прихованими.

Окремо реалізовано роботу з медіафайлами. Зображення використовуються для новин, партнерів і галереї, а файли – для ресурсів або документів. Під час завантаження система перевіряє тип, розмір і коректність збереження файлу.

Короткий приклад поля для завантаження зображення наведено в лістингу 3.6.

Лістинг 3.6 – Фрагмент поля завантаження зображення

```
FileUpload::make('image')  
->label('Image')  
->image()  
->directory('news')  
->maxSize(2048);
```

У межах адміністративної панелі передбачено також розмежування доступу. Адміністратор має повний доступ до керування контентом, користувачами та системними налаштуваннями. Редактор працює переважно з матеріалами, новинами, сторінками й іншими записами, але не має доступу до критичних параметрів системи.

Отже, у межах підрозділу описано реалізацію адміністративної панелі та механізмів керування контентом. Використання Filament v5 дозволило створити зручне середовище для роботи з основними інформаційними об'єктами системи. Реалізовані форми, таблиці, статуси публікації, валідація, завантаження

медіафайлів і розмежування доступу забезпечують повноцінне адміністрування сайту міжнародного освітнього проєкту.

3.4 Реалізація мультимовності та динамічного підвантаження даних

У веб-орієнтованій інформаційній системі міжнародного освітнього проєкту важливе значення має підтримка мультимовності, оскільки сайт орієнтований на користувачів із різних країн. У межах реалізованої системи передбачено мовні версії інтерфейсу для EN, ES, UK, EL, TR, FI. Це дозволяє адаптувати основні елементи навігації, службові написи, кнопки та повідомлення до мовних потреб цільової аудиторії. Такий підхід відповідає загальним вимогам до міжнародних вебресурсів, де важливо забезпечити зрозумілу навігацію та доступність інформації різними мовами [9, 23, 24].

Мультимовність у системі реалізовано за комбінованим підходом. Статичні елементи інтерфейсу зберігаються у мовних файлах Laravel, а контент, створений через адміністративну панель, може перекладатися після додавання із застосуванням сервісу DeepL. Такий поділ є зручним, оскільки службові написи інтерфейсу змінюються рідко, тоді як новини, сторінки, описи партнерів і матеріали можуть регулярно оновлюватися.

У загальному вигляді процес роботи з мультимовним контентом передбачає такі етапи:

- адміністратор створює запис основною мовою;
- система зберігає запис у базі даних;
- за потреби ініціюється переклад контенту;
- текстові поля передаються до DeepL;
- отриманий переклад зберігається як мовна версія запису;
- користувач обирає потрібну мову в публічній частині сайту;
- система відображає відповідну мовну версію інтерфейсу та контенту.

У таблиці 3.10 наведено основні компоненти реалізації мультимовності.

Таблиця 3.10 – Компоненти реалізації мультимовності

Компонент	Призначення
Мовні файли Laravel	Збереження статичних написів інтерфейсу
Перемикач мов	Вибір мовної версії сайту користувачем
DeepL	Переклад створеного контенту
База даних	Збереження мовних версій інформаційних об'єктів
Blade-шаблони	Відображення перекладених елементів у публічній частині

Короткий приклад використання мовного файлу у Blade-шаблоні наведено в лістингу 3.7.

Лістинг 3.7 – Приклад виведення перекладеного напису в Blade-шаблоні

```
<a href="{{ route('news.index') }}">
    {{ __('navigation.news') }}
</a>
```

У наведеному фрагменті замість фіксованого тексту використовується ключ `navigation.news`, значення якого береться з відповідного мовного файлу. Це дозволяє змінювати мову інтерфейсу без редагування HTML-шаблонів. Повний код мовних файлів подано в додатку А.

Крім мультимовності, у системі реалізовано динамічне підвантаження даних. Такий механізм використовується для сторінок, де може бути велика кількість записів, наприклад новин, ресурсів або елементів галереї. Його призначення полягає в тому, щоб не завантажувати весь контент одразу, а поступово додавати записи на сторінку. У реалізованій системі для цього використовується Livewire 4, який забезпечує оновлення окремих частин сторінки без повного перезавантаження.

Алгоритм динамічного підвантаження даних має таку послідовність:

- користувач відкриває сторінку зі списком записів;
- система завантажує початкову частину даних;
- користувач натискає кнопку підвантаження;
- Livewire-компонент надсилає запит до серверної частини;

- система вибирає наступну групу записів із бази даних;
- нові записи додаються до вже відображеного списку;
- сторінка оновлюється частково, без повного перезавантаження.

У таблиці 3.11 подано переваги використання Livewire для динамічного підвантаження.

Таблиця 3.11 – Переваги використання Livewire 4 для підвантаження даних

Перевага	Зміст
Часткове оновлення сторінки	Оновлюється лише потрібний блок контенту
Зменшення початкового навантаження	Не потрібно завантажувати всі записи одразу
Інтеграція з Laravel	Компонент працює в межах серверної логіки системи
Зручність для користувача	Нові записи додаються без переходу на іншу сторінку
Придатність до розширення	Механізм можна застосувати до новин, ресурсів і галереї

Короткий приклад логіки підвантаження записів наведено в лістингу 3.8.

Лістинг 3.8 – Приклад логіки динамічного підвантаження записів

```
public int $limit = 10;

public function loadMore(): void
{
    $this->limit += 10;
}

public function render()
{
    return view('livewire.news-list', [
        'news' => News::latest()
            ->take($this->limit)
            ->get()
    ]);
}
```

У лістингу 3.8 показано спрощений принцип роботи Livewire-компонента.

Спочатку відображається певна кількість записів, а після виклику методу `loadMore()` кількість завантажених елементів збільшується. Повну реалізацію Livewire-компонентів подано в додатку А.

Отже, у системі реалізовано мультимовність і динамічне підвантаження даних як важливі функціональні можливості публічної частини сайту. Мультимовність забезпечує доступність інформації для міжнародної аудиторії, а використання Livewire 4 покращує взаємодію користувача зі списками контенту. Поєднання мовних файлів, DeepL, Blade-шаблонів і Livewire-компонентів дозволило зробити систему гнучкою, зручною для супроводу та придатною до подальшого розширення.

3.5 Тестування веб-орієнтованої інформаційної системи

Після реалізації основних модулів веб-орієнтованої інформаційної системи виконано перевірку її працездатності. Метою тестування було встановлення відповідності системи сформованим вимогам, перевірка коректності роботи публічної частини сайту, адміністративної панелі, мультимовності, динамічного підвантаження даних і базових механізмів безпеки.

Тестування виконано за функціональним підходом. Перевірялися типові сценарії використання системи: відкриття сторінок, перегляд новин, перехід до детального опису партнера, зміна мовної версії, робота з адміністративною панеллю, створення запису, валідація форми та відображення опублікованого контенту в публічній частині сайту.

У таблиці 3.12 наведено результати функціонального тестування публічної частини системи.

Як видно з таблиці 3.12, основні сторінки публічної частини сайту відкриваються коректно та забезпечують доступ користувачів до ключових інформаційних розділів. Окремі розділи, зокрема Resources, Gallery і навчальні сторінки, мають підготовлену структуру для подальшого наповнення матеріалами.

Окремо виконано перевірку адміністративної частини системи. Основну увагу приділено сценаріям створення, редагування, збереження та публікації

контенту. Результати тестування адміністративної панелі наведено в таблиці 3.13.

Таблиця 3.12 – Результати тестування публічної частини сайту

Тестовий сценарій	Очікуваний результат	Фактичний результат
Відкриття головної сторінки	Відображення основної інформації про проєкт	Головна сторінка відображається коректно
Перехід до сторінки About	Відображення опису проєкту, місії та цілей	Дані відображаються коректно
Перехід до розділу News	Відображення списку новин	Список новин доступний
Відкриття окремої новини	Відображення повного тексту новини	Детальна сторінка відкривається
Перехід до сторінки Partners	Відображення переліку партнерів	Партнери відображаються картками
Відкриття сторінки партнера	Відображення опису організації	Детальна сторінка партнера доступна
Перехід до Resources	Відображення сторінки ресурсів	Структура сторінки реалізована
Перехід до Gallery	Відображення сторінки галереї	Структура сторінки реалізована
Перехід до Get Involved	Відображення комунікаційного блоку	Сторінка відкривається коректно

Таблиця 3.13 – Результати тестування адміністративної панелі

Вхід адміністратора	Доступ після введення правильних даних	Авторизація виконується
Відкриття розділу News	Відображення списку новин	Розділ доступний
Створення новини	Збереження нового запису	Запис створюється
Редагування новини	Оновлення наявного запису	Дані оновлюються
Встановлення статусу публікації	Керування видимістю запису	Статус застосовується

Продовження таблиці 3.13

Вхід адміністратора	Доступ після введення правильних даних	Авторизація виконується
Завантаження зображення	Збереження медіафайлу	Зображення додається до запису
Видалення запису	Вилучення або приховування матеріалу	Операція виконується
Валідація обов'язкових полів	Виведення повідомлення про помилку	Некоректні дані не зберігаються
Вхід адміністратора	Доступ після введення правильних даних	Авторизація виконується

Дані таблиці 3.13 підтверджують, що адміністративна панель забезпечує базовий набір операцій, необхідних для керування контентом. Реалізовано створення, редагування, публікацію, приховування та видалення записів, а також перевірку введених даних.

Для перевірки мультимовності протестовано перемикання мовних версій інтерфейсу та відображення перекладених елементів. У системі передбачено мовні варіанти EN, ES, UK, EL, TR, FI. Перевірку виконано для навігаційних елементів, заголовків, службових написів і сторінок публічної частини. Узагальнені результати наведено в таблиці 3.14.

Таблиця 3.14 – Результати тестування мультимовності

Тестовий сценарій	Очікуваний результат	Фактичний результат
Перемикання на EN	Відображення англійської версії	Мова змінюється
Перемикання на UK	Відображення української версії	Мова змінюється
Перемикання на ES	Відображення іспанської версії	Мова змінюється
Перемикання на EL	Відображення грецької версії	Мова змінюється
Перемикання на TR	Відображення турецької версії	Мова змінюється
Перемикання на FI	Відображення фінської версії	Мова змінюється
Відображення службових написів	Підстановка значень із мовних файлів	Написи відображаються відповідною мовою

Результати таблиці 3.14 показують, що механізм мультимовності працює відповідно до поставленої задачі. Це важливо для міжнародного освітнього проєкту, оскільки користувачі системи можуть належати до різних мовних груп.

Також виконано перевірку динамічного підвантаження даних. Такий механізм потрібний для сторінок, де кількість записів може поступово збільшуватися, зокрема для новин, ресурсів або галереї. Результати перевірки наведено в таблиці 3.15.

Таблиця 3.15 – Результати тестування динамічного підвантаження даних

Тестовий сценарій	Очікуваний результат	Фактичний результат
Початкове відкриття списку	Завантаження першої групи записів	Початкові записи відображаються
Натискання кнопки підвантаження	Завантаження наступної групи записів	Нові записи додаються до списку
Часткове оновлення сторінки	Оновлення без повного перезавантаження	Сторінка не перезавантажується повністю
Завершення списку	Приховування або вимкнення кнопки	Подальше підвантаження не виконується

Перевірка показала, що використання Livewire 4 забезпечує зручний сценарій роботи зі списками контенту. Користувач може поступово переглядати нові записи без переходу на іншу сторінку та без повторного завантаження всієї сторінки.

Додатково виконано оцінювання відповідності реалізованої системи вимогам, визначеним у другому розділі. Результати такого оцінювання подано в таблиці 3.16.

Як видно з таблиці 3.16, більшість поставлених вимог реалізовано. Частково виконаною є вимога щодо наповнення окремих освітніх і ресурсних розділів, оскільки їх структура вже підготовлена, однак кінцеве контентне наповнення залежить від подальших етапів реалізації міжнародного проєкту.

Під час тестування також перевірено базову зручність використання сайту. Навігація між основними розділами є зрозумілою, сторінки мають єдину

структуру, а публічна частина забезпечує швидкий доступ до ключової інформації про проєкт. Наявність перемикача мов, сторінки партнерів, новинного модуля та сторінки залучення користувачів підвищує практичну придатність системи.

Таблиця 3.16 – Відповідність реалізованої системи поставленим вимогам

Вимога	Реалізація
Публічне подання інформації	Реалізовано головну сторінку, About, News, Partners, Get Involved
Керування контентом	Реалізовано адміністративну панель на основі Filament v5
Зберігання даних	Використано реляційну базу даних MySQL
Мультимовність	Реалізовано мовні файли та переклад контенту
Динамічне підвантаження	Реалізовано через Livewire 4
Адаптивність	Інтерфейс пристосовано до різних екранів
Безпека доступу	Передбачено автентифікацію адміністратора
Розгортання	Систему піднято за допомогою Docker
Розширення функціональності	Підготовлено сторінки Resources, Gallery, Courses, Materials, Quizzes, Pilots & Events

Отже, у підрозділі виконано тестування та оцінювання працездатності веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту. Перевірено роботу публічної частини сайту, адміністративної панелі, механізмів мультимовності, динамічного підвантаження даних, валідації форм і базових засобів доступу. Результати тестування підтвердили працездатність розробленої системи та її відповідність основним функціональним вимогам. Виявлено, що система є придатною для інформаційного супроводу міжнародного освітнього проєкту та може бути розширена шляхом подальшого наповнення навчальних і ресурсних розділів.

ВИСНОВКИ

1. Виконано аналіз предметної області веб-орієнтованих інформаційних систем керування контентом міжнародного освітнього проєкту. Визначено, що специфіка таких систем зумовлюється поєднанням інформаційної, комунікаційної та організаційної функцій, оскільки вебресурс повинен забезпечувати не лише представлення проєкту, а й централізоване керування новинами, сторінками, матеріалами, даними про партнерів, галереєю та іншими видами контенту. Встановлено, що важливими вимогами до таких систем є багатомовність, адаптивність інтерфейсу, зручність адміністрування, безпечне оброблення даних і можливість подальшого супроводу.

2. Проаналізовано існуючі підходи та методи побудови систем керування контентом. Визначено, що для поставленої задачі найбільш доцільним є індивідуальний фреймворковий підхід, який забезпечує достатню гнучкість, структурованість програмної логіки та можливість адаптації до потреб міжнародного освітнього проєкту. Обґрунтовано використання технологічного стеку, що включає Laravel 12, Blade-шаблони, MySQL, Docker, Filament v5, Livewire 4 та механізми багатомовності на основі мовних файлів і DeepL.

3. На основі проведеного аналізу сформульовано постановку задачі дослідження, визначено основну мету та перелік завдань, спрямованих на розроблення веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту.

4. Виконано проєктування веб-орієнтованої інформаційної системи. Сформовано функціональні та нефункціональні вимоги до системи, визначено її загальну архітектуру та основні компоненти. Встановлено, що система повинна поєднувати публічну частину сайту, адміністративне середовище, модуль мультимовності, механізми динамічного підвантаження даних, форми зворотного зв'язку та засоби контролю доступу. Це дозволило побудувати цілісне уявлення про структуру програмного рішення та його місце в предметній області.

5. Спроектовано структуру даних і модель основних контентних

інформаційних об'єктів системи. Виділено базові інформаційні об'єкти, зокрема сторінки, новини, партнерів, ресурси, елементи галереї, категорії, користувачів адміністративної панелі та повідомлення форм зворотного зв'язку. Визначено їх основні атрибути та логічні зв'язки, що створило основу для реалізації моделі даних у середовищі Laravel і MySQL.

6. Спроектовано функціональні модулі системи та розроблено основні алгоритми її роботи, зокрема алгоритми відображення контенту, створення і публікації записів, перекладу контенту, динамічного підвантаження даних та оброблення повідомлень користувачів.

7. Виконано опис програмної реалізації та тестування веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту. Реалізацію системи здійснено на основі технологічного стеку Laravel 12 + PHP, Blade-шаблонів, MySQL, Filament v5, Livewire 4, Docker і засобів підтримки мультимовності. Такий підхід забезпечив логічне розділення серверної частини, публічного інтерфейсу, адміністративного середовища, бази даних і механізмів динамічної взаємодії.

8. Розроблено структуру програмного проєкту, у якій виділено моделі, контролери, маршрути, Blade-шаблони, адміністративні ресурси Filament, Livewire-компоненти, мовні файли та конфігурацію контейнерного середовища. Це дозволило організувати програмний код відповідно до принципу розділення відповідальності та забезпечити зручність подальшого супроводу системи.

9. Реалізовано публічну частину сайту YouthSOFTPOWER, яка містить головну сторінку, сторінку опису проєкту, модуль новин, сторінки партнерів, розділи ресурсів, галереї, навчальної платформи та сторінку залучення користувачів. Публічний інтерфейс забезпечує структуроване подання інформації про міжнародний освітній проєкт, його цілі, партнерські організації, новини та комунікаційні можливості. Окремі розділи, зокрема навчальні матеріали, ресурси й галерея, підготовлено для подальшого контентного наповнення.

10. Реалізовано адміністративну панель на основі Filament v5, яка забезпечує створення, редагування, публікацію, приховування та видалення інформаційних матеріалів. Через адміністративне середовище передбачено

керування сторінками, новинами, партнерами, ресурсами, галереєю, категоріями та повідомленнями користувачів. Застосування форм, таблиць, статусів публікації, валідації введених даних і завантаження медіафайлів забезпечило повний цикл керування контентом без прямого редагування програмного коду.

11. У системі реалізовано мультимовність, що є важливою вимогою для міжнародного освітнього проєкту. Статичні елементи інтерфейсу організовано через мовні файли Laravel, а переклад створеного контенту передбачено із застосуванням DeepL. Крім того, для покращення взаємодії користувача зі списками записів використано Livewire 4, що дозволяє виконувати динамічне підвантаження даних без повного перезавантаження сторінки.

12. Проведене тестування підтвердило працездатність основних модулів системи. Перевірено відкриття публічних сторінок, перегляд списку новин і детальних записів, відображення партнерів, роботу мовних версій, функції адміністративної панелі, валідацію форм, збереження записів у базі даних і динамічне підвантаження контенту. Результати тестування показали, що система відповідає основним функціональним вимогам і може використовуватися для інформаційного супроводу міжнародного освітнього проєкту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bibichkov I., Sokol V., Shevchenko O. Ontological approach to development of web-content generation method. *Technology Audit and Production Reserves*. 2019. Vol. 5 No. 2(49). P. 4–10.
2. Biilmann M., Hawksworth P. *Modern Web Development on the JAMstack: Modern Techniques for Ultra Fast Sites and Web Applications*. Sebastopol : O'Reilly Media, 2019. URL: <https://www.netlify.com/oreilly-jamstack/> (дата звернення: 13.01.2026).
3. Bootstrap Authors. *Bootstrap 5.3: Forms overview*. URL: <https://getbootstrap.com/docs/5.3/forms/overview/> (дата звернення: 13.01.2026).
4. Bootstrap Authors. *Bootstrap 5.3: Get started with Bootstrap*. URL: <https://getbootstrap.com/docs/5.3/getting-started/introduction/> (дата звернення: 13.01.2026).
5. Bootstrap Authors. *Bootstrap 5.3: Grid system*. URL: <https://getbootstrap.com/docs/5.3/layout/grid/> (дата звернення: 13.01.2026).
6. Bootstrap Authors. *Bootstrap 5.3: Validation*. URL: <https://getbootstrap.com/docs/5.3/forms/validation/> (дата звернення: 13.01.2026).
7. Django Software Foundation. *Django documentation*. URL: <https://docs.djangoproject.com/en/6.0/> (дата звернення: 13.01.2026).
8. European Commission. *Erasmus+ Programme Guide 2026*. URL: https://erasmus-plus.ec.europa.eu/sites/default/files/2025-11/programme-guide-2026_en.pdf (дата звернення: 13.01.2026).
9. European Commission. *Languages on our websites*. URL: https://commission.europa.eu/languages-our-websites_en (дата звернення: 13.01.2026).
10. Information Commissioner's Office. *Data protection by design and by default*. URL: <https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/accountability-and-governance/guide-to-accountability-and-governance/data-protection-by-design-and-by-default/> (дата звернення: 13.01.2026).
11. Lytvyn V., Vysotska V., Demchuk A., Demkiv I., Ukhanska O., Hladun V., Kovalchuk R., Petruchenko O., Dzyubyk L., Sokulska N. Design of the architecture of an

intelligent system for distributing commercial content in the internet space based on SEO-technologies, neural networks, and Machine Learning. *Eastern-European Journal of Enterprise Technologies*. 2019. Vol. 2, no. 2(98). P. 15–34.

12. McKeever S. Understanding Web Content Management Systems: Evolution, Lifecycle and Market. *Industrial Management & Data Systems*. 2003. Vol. 103, no. 8/9. P. 686–692.

13. MDN Web Docs. *Overview of HTTP*. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/Overview> (дата звернення: 13.01.2026).

14. MDN Web Docs. *Responsive web design*. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/CSS_layout/Responsive_Design (дата звернення: 13.01.2026).

15. MDN Web Docs. *Using media queries*. URL: https://developer.mozilla.org/en-US/docs/Web/CSS/Guides/Media_queries/Using (дата звернення: 13.01.2026).

16. MDN Web Docs. *Web forms*. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Forms (дата звернення: 13.01.2026).

17. OWASP Foundation. *OWASP Top Ten Web Application Security Risks*. URL: <https://owasp.org/www-project-top-ten/> (дата звернення: 13.01.2026).

18. PostgreSQL Global Development Group. *PostgreSQL Documentation*. URL: <https://www.postgresql.org/docs/> (дата звернення: 13.01.2026).

19. Priefer D., Karol S., Völter M. Model-driven Development of Content Management Systems. In: *Proceedings of MODELS 2014*. 2014. P. 1–6.

20. Rola P., Kuchta D. A Content Management System as an Information Management System in Interdisciplinary Research. *Acta Universitatis Lodzianae. Folia Oeconomica*. 2020. Vol. 5(350). P. 65–81.

21. Tsamis G., Kartakoullis N., Pallis G., Dikaiakos M. D. Cost-Effective Design, Content Management System, and Deployment of a Digital Platform for Public Financial Data. *Algorithms*. 2025. Vol. 18, no. 6. Art. 339.

22. Vlasenko K.V., Volkov S.V., Lovianova I.V., Chumak O.O., Sitak I.V. and

Bobyliiev D.Y. Management of online platform development and support process. *CTE Workshop Proceedings*. 2021. Vol. 8. P.333–345.

23. W3C. *International & multilingual web sites*. URL: <https://www.w3.org/International/questions/qa-international-multilingual> (дата звернення: 13.01.2026).

24. W3C. *Internationalization techniques: Authoring web pages*. URL: <https://www.w3.org/International/techniques/authoring-html> (дата звернення: 13.01.2026).

25. W3C. *Web Content Accessibility Guidelines (WCAG) 2.2*. URL: <https://www.w3.org/TR/WCAG22/> (дата звернення: 13.01.2026).

26. Xie W., Li X. Design and Implementation of Content Management System. In: *Proceedings of the 2023 8th International Conference on Intelligent Information Technology*. 2023. P. 213–217.

27. Романишин Ю., Лаба О., Лесюк О. Формування професійного контенту інтернет-ресурсів у маркетинговому середовищі організації. *Цифрова платформа: інформаційні технології в соціокультурній сфері*. 2023. № 6(1). С. 217–230.

28. Комар М.П., Горбаль Ю.-М.М. Проектування веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту. Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення: матеріали Міжнародної наукової інтернет-конференції, м. Тернопіль, Україна, м. Ополе, Польща, 7-8 травня 2026 р. 2026. С. 35-37.

29. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

Додаток А

Лістинги програмного коду

А.1 Структура програмного проекту

```
youthsoftpower/  
├── app/  
│   ├── Filament/  
│   │   └── Resources/  
│   │       ├── NewsResource.php  
│   │       ├── PageResource.php  
│   │       ├── PartnerResource.php  
│   │       ├── ResourceResource.php  
│   │       └── GalleryItemResource.php  
│   ├── Http/  
│   │   └── Controllers/  
│   │       ├── HomeController.php  
│   │       ├── NewsController.php  
│   │       ├── PartnerController.php  
│   │       ├── PageController.php  
│   │       └── ContactController.php  
│   ├── Livewire/  
│   │   ├── NewsList.php  
│   │   └── ResourceList.php  
│   └── Models/  
│       ├── News.php  
│       ├── Page.php  
│       ├── Partner.php  
│       ├── Resource.php  
│       ├── GalleryItem.php  
│       ├── Category.php  
│       └── ContactRequest.php  
├── database/  
│   └── migrations/  
├── resources/  
│   ├── lang/  
│   │   ├── en/  
│   │   ├── uk/  
│   │   ├── es/  
│   │   ├── el/  
│   │   ├── tr/  
│   │   └── fi/  
│   └── views/  
│       ├── layouts/  
│       ├── pages/  
│       ├── news/  
│       ├── partners/  
│       ├── resources/  
│       └── livewire/  
├── routes/  
│   └── web.php  
├── docker-compose.yml  
├── Dockerfile  
└── .env
```

А.2 Маршрути публічної частини системи

```
<?php  
  
use Illuminate\Support\Facades\Route;  
use App\Http\Controllers\HomeController;  
use App\Http\Controllers\NewsController;
```

```

use App\Http\Controllers\PartnerController;
use App\Http\Controllers\PageController;
use App\Http\Controllers\ResourceController;
use App\Http\Controllers>ContactController;

Route::get('/', [HomeController::class, 'index'])
    ->name('home');

Route::get('/about', [PageController::class, 'about'])
    ->name('about');

Route::get('/news', [NewsController::class, 'index'])
    ->name('news.index');

Route::get('/news/{slug}', [NewsController::class, 'show'])
    ->name('news.show');

Route::get('/partners', [PartnerController::class, 'index'])
    ->name('partners.index');

Route::get('/partners/{slug}', [PartnerController::class, 'show'])
    ->name('partners.show');

Route::get('/resources', [ResourceController::class, 'index'])
    ->name('resources.index');

Route::get('/get-involved', [ContactController::class, 'index'])
    ->name('contact.index');

Route::post('/get-involved', [ContactController::class, 'store'])
    ->name('contact.store');

```

A.3 Контролер головної сторінки

```

<?php

namespace App\Http\Controllers;

use App\Models\News;
use App\Models\Partner;
use App\Models\GalleryItem;

class HomeController extends Controller
{
    public function index()
    {
        $latestNews = News::query()
            ->where('is_active', true)
            ->latest('published_at')
            ->take(3)
            ->get();

        $partners = Partner::query()
            ->where('is_active', true)
            ->orderBy('sort_order')
            ->get();

        $galleryItems = GalleryItem::query()
            ->where('is_active', true)
            ->latest()
            ->take(6)
            ->get();

        return view('pages.home', compact(
            'latestNews',

```

```

        'partners',
        'galleryItems'
    ));
}
}

```

A.4 Контролер новин

```

<?php

namespace App\Http\Controllers;

use App\Models\News;

class NewsController extends Controller
{
    public function index()
    {
        $news = News::query()
            ->where('is_active', true)
            ->latest('published_at')
            ->paginate(9);

        return view('news.index', compact('news'));
    }

    public function show(string $slug)
    {
        $newsItem = News::query()
            ->where('slug', $slug)
            ->where('is_active', true)
            ->firstOrFail();

        return view('news.show', compact('newsItem'));
    }
}

```

A.5 Контролер партнерів

```

<?php

namespace App\Http\Controllers;

use App\Models\Partner;

class PartnerController extends Controller
{
    public function index()
    {
        $partners = Partner::query()
            ->where('is_active', true)
            ->orderBy('sort_order')
            ->get();

        return view('partners.index', compact('partners'));
    }

    public function show(string $slug)
    {
        $partner = Partner::query()
            ->where('slug', $slug)
            ->where('is_active', true)
            ->firstOrFail();
    }
}

```

```

        return view('partners.show', compact('partner'));
    }
}

```

A.6 Модель новини

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Relations\BelongsToMany;

class News extends Model
{
    use HasFactory;

    protected $fillable = [
        'title',
        'slug',
        'excerpt',
        'content',
        'image',
        'published_at',
        'is_active',
    ];

    protected $casts = [
        'published_at' => 'datetime',
        'is_active' => 'boolean',
    ];

    public function categories(): BelongsToMany
    {
        return $this->belongsToMany(Category::class, 'news_category');
    }
}

```

A.7 Модель партнера

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;

class Partner extends Model
{
    use HasFactory;

    protected $fillable = [
        'name',
        'slug',
        'description',
        'country',
        'logo',
        'image',
        'url',
        'sort_order',
        'is_active',
    ];
}

```

```

        protected $casts = [
            'is_active' => 'boolean',
            'sort_order' => 'integer',
        ];
    }

```

A.8 Модель інформаційної сторінки

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;

class Page extends Model
{
    use HasFactory;

    protected $fillable = [
        'title',
        'slug',
        'content',
        'meta_title',
        'meta_description',
        'is_active',
    ];

    protected $casts = [
        'is_active' => 'boolean',
    ];
}

```

A.9 Міграція таблиці новин

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('news', function (Blueprint $table) {
            $table->id();
            $table->string('title');
            $table->string('slug')->unique();
            $table->text('excerpt')->nullable();
            $table->longText('content');
            $table->string('image')->nullable();
            $table->dateTime('published_at')->nullable();
            $table->boolean('is_active')->default(false);
            $table->timestamps();
        });
    }

    public function down(): void
    {
        Schema::dropIfExists('news');
    }
};

```

A.10 Міграція таблиці партнерів

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('partners', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('slug')->unique();
            $table->longText('description')->nullable();
            $table->string('country')->nullable();
            $table->string('logo')->nullable();
            $table->string('image')->nullable();
            $table->string('url')->nullable();
            $table->integer('sort_order')->default(0);
            $table->boolean('is_active')->default(true);
            $table->timestamps();
        });
    }

    public function down(): void
    {
        Schema::dropIfExists('partners');
    }
};
```

A.11 Проміжна таблиця для зв'язку новин і категорій

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('news_category', function (Blueprint $table) {
            $table->id();

            $table->foreignId('news_id')
                ->constrained('news')
                ->cascadeOnDelete();

            $table->foreignId('category_id')
                ->constrained('categories')
                ->cascadeOnDelete();

            $table->timestamps();
        });
    }

    public function down(): void
    {
        Schema::dropIfExists('news_category');
    }
};
```

A.12 Blade-шаблон головної сторінки

```
@extends('layouts.app')

@section('content')
    <section class="hero">
        <div class="container">
            <h1>{{ __('home.hero_title') }}</h1>
            <p>{{ __('home.hero_description') }}</p>

            <a href="{{ route('about') }}" class="btn btn-primary">
                {{ __('home.learn_more') }}
            </a>
        </div>
    </section>

    <section class="latest-news">
        <div class="container">
            <h2>{{ __('navigation.news') }}</h2>

            <div class="news-grid">
                @foreach($latestNews as $item)
                    <article class="news-card">
                        @if($item->image)
                            title }}">
                        @endif

                        <span class="date">
                            {{ $item->published_at?->format('d.m.Y') }}
                        </span>

                        <h3>{{ $item->title }}</h3>
                        <p>{{ $item->excerpt }}</p>

                        <a href="{{ route('news.show', $item->slug) }}">
                            {{ __('buttons.read_more') }}
                        </a>
                    </article>
                @endforeach
            </div>
        </div>
    </section>
@endsection
```

A.13 Blade-шаблон списку новин

```
@extends('layouts.app')

@section('content')
    <section class="page-header">
        <div class="container">
            <h1>{{ __('navigation.news') }}</h1>
        </div>
    </section>

    <section class="news-list">
        <div class="container">
            <div class="news-grid">
                @foreach($news as $item)
                    <article class="news-card">
                        @if($item->image)
                            title }}">
                        @endif
                    </article>
                @endforeach
            </div>
        </div>
    </section>
@endsection
```

```

        <span>{{ $item->published_at?->format('d.m.Y') }}</span>

        <h3>{{ $item->title }}</h3>

        <p>{{ $item->excerpt }}</p>

        <a href="{{ route('news.show', $item->slug) }}">
            {{ __('buttons.read_more') }}
        </a>
    </article>
@endforeach
</div>

    {{ $news->links() }}
</div>
</section>
@endsection

```

A.14 Blade-шаблон детальної сторінки новини

```

@extends('layouts.app')

@section('content')
    <article class="news-detail">
        <div class="container">
            <nav class="breadcrumbs">
                <a href="{{ route('home') }}">{{ __('navigation.home') }}</a>
                <span></span>
                <a href="{{ route('news.index') }}">{{ __('navigation.news') }}</a>
            </nav>

            <h1>{{ $newsItem->title }}</h1>

            <span class="date">
                {{ $newsItem->published_at?->format('d.m.Y') }}
            </span>

            @if($newsItem->image)
                title }}"
                    class="news-main-image">
            @endif

            <div class="content">
                {!! $newsItem->content !!}
            </div>
        </div>
    </article>
@endsection

```

A.15 Blade-шаблон сторінки партнерів

```

@extends('layouts.app')

@section('content')
    <section class="partners">
        <div class="container">
            <h1>{{ __('navigation.partners') }}</h1>

            <div class="partners-grid">
                @foreach($partners as $partner)
                    <article class="partner-card">

```

```

        @if($partner->logo)
            name }}">
        @endif

        <h3>{{ $partner->name }}</h3>

        <p>{{ Str::limit(strip_tags($partner->description), 150)
    }}</p>

        <a href="{{ route('partners.show', $partner->slug) }}">
            {{ __('buttons.more_details') }}
        </a>
    </article>
@endforeach
</div>
</div>
</section>
@endsection

```

A.16 Адміністративний ресурс новин Filament

```

<?php

namespace App\Filament\Resources;

use App\Models\News;
use Filament\Forms;
use Filament\Tables;
use Filament\Resources\Resource;
use Filament\Forms\Form;
use Filament\Tables\Table;
use Filament\Forms\Components\TextInput;
use Filament\Forms\Components\Textarea;
use Filament\Forms\Components\RichEditor;
use Filament\Forms\Components\FileUpload;
use Filament\Forms\Components\DateTimePicker;
use Filament\Forms\Components\Toggle;

class NewsResource extends Resource
{
    protected static ?string $model = News::class;

    protected static ?string $navigationIcon = 'heroicon-o-newspaper';

    protected static ?string $navigationLabel = 'News';

    public static function form(Form $form): Form
    {
        return $form->schema([
            TextInput::make('title')
                ->label('Title')
                ->required()
                ->maxLength(255),

            TextInput::make('slug')
                ->label('Slug')
                ->required()
                ->unique(ignoreRecord: true)
                ->maxLength(255),

            Textarea::make('excerpt')
                ->label('Short description')
                ->rows(3),

```

```

        RichEditor::make('content')
            ->label('Content')
            ->required()
            ->columnSpanFull(),

        FileUpload::make('image')
            ->label('Image')
            ->image()
            ->directory('news')
            ->maxSize(2048),

        DateTimePicker::make('published_at')
            ->label('Publication date'),

        Toggle::make('is_active')
            ->label('Published')
            ->default(false),
    ]);
}

public static function table(Table $table): Table
{
    return $table
        ->columns([
            Tables\Columns\ImageColumn::make('image')
                ->label('Image'),

            Tables\Columns\TextColumn::make('title')
                ->label('Title')
                ->searchable()
                ->sortable(),

            Tables\Columns\TextColumn::make('published_at')
                ->label('Published at')
                ->dateTime()
                ->sortable(),

            Tables\Columns\IconColumn::make('is_active')
                ->label('Published')
                ->boolean(),
        ])
        ->actions([
            Tables\Actions\EditAction::make(),
            Tables\Actions\DeleteAction::make(),
        ])
        ->bulkActions([
            Tables\Actions\DeleteBulkAction::make(),
        ]);
}
}

```

A.17 Адміністративний ресурс партнерів Filament

<?php

```

namespace App\Filament\Resources;

use App\Models\Partner;
use Filament\Forms\Form;
use Filament\Tables\Table;
use Filament\Resources\Resource;
use Filament\Forms\Components\TextInput;
use Filament\Forms\Components\Textarea;
use Filament\Forms\Components\FileUpload;
use Filament\Forms\Components\Toggle;

```

```

use Filament\Tables;

class PartnerResource extends Resource
{
    protected static ?string $model = Partner::class;

    protected static ?string $navigationIcon = 'heroicon-o-building-office';

    protected static ?string $navigationLabel = 'Partners';

    public static function form(Form $form): Form
    {
        return $form->schema([
            TextInput::make('name')
                ->label('Partner name')
                ->required()
                ->maxLength(255),

            TextInput::make('slug')
                ->label('Slug')
                ->required()
                ->unique(ignoreRecord: true),

            TextInput::make('country')
                ->label('Country')
                ->maxLength(255),

            TextInput::make('url')
                ->label('Website')
                ->url()
                ->maxLength(255),

            FileUpload::make('logo')
                ->label('Logo')
                ->image()
                ->directory('partners/logos'),

            FileUpload::make('image')
                ->label('Image')
                ->image()
                ->directory('partners/images'),

            Textarea::make('description')
                ->label('Description')
                ->rows(6)
                ->columnSpanFull(),

            TextInput::make('sort_order')
                ->label('Sort order')
                ->numeric()
                ->default(0),

            Toggle::make('is_active')
                ->label('Active')
                ->default(true),
        ]);
    }

    public static function table(Table $table): Table
    {
        return $table
            ->columns([
                Tables\Columns\ImageColumn::make('logo')
                    ->label('Logo'),

                Tables\Columns\TextColumn::make('name')

```

```

        ->label('Name')
        ->searchable()
        ->sortable(),

        Tables\Columns\TextColumn::make('country')
        ->label('Country'),

        Tables\Columns\TextColumn::make('sort_order')
        ->label('Order')
        ->sortable(),

        Tables\Columns\IconColumn::make('is_active')
        ->label('Active')
        ->boolean(),
    ])
    ->actions([
        Tables\Actions\EditAction::make(),
        Tables\Actions\DeleteAction::make(),
    ]);
}
}

```

A.18 Livewire-компонент динамічного підвантаження новин

```

<?php

namespace App\Livewire;

use App\Models\News;
use Livewire\Component;

class NewsList extends Component
{
    public int $limit = 10;

    public function loadMore(): void
    {
        $this->limit += 10;
    }

    public function render()
    {
        $news = News::query()
            ->where('is_active', true)
            ->latest('published_at')
            ->take($this->limit)
            ->get();

        return view('livewire.news-list', [
            'news' => $news,
        ]);
    }
}

```

A.19 Blade-шаблон Livewire-компонента новин

```

<div>
    <div class="news-grid">
        @foreach($news as $item)
            <article class="news-card">
                @if($item->image)
                    title }}">
                @endif
            </article>
        @endforeach
    </div>
</div>

```

```

        <span class="date">
            {{ $item->published_at?->format('d.m.Y') }}
        </span>

        <h3>{{ $item->title }}</h3>

        <p>{{ $item->excerpt }}</p>

        <a href="{{ route('news.show', $item->slug) }}">
            {{ __('buttons.read_more') }}
        </a>
    </article>
@endforeach
</div>

<button type="button"
        wire:click="loadMore"
        class="btn btn-outline-primary">
    {{ __('buttons.load_more') }}
</button>
</div>

```

A.20 Мовний файл навігації

```

<?php

return [
    'home' => 'Home',
    'about' => 'About',
    'news' => 'News',
    'partners' => 'Partners',
    'resources' => 'Resources',
    'gallery' => 'Gallery',
    'get_involved' => 'Get Involved',
    'learning_platform' => 'Learning Platform',
    'courses' => 'Courses',
    'materials' => 'Materials',
    'quizzes' => 'Quizzes',
    'pilots_events' => 'Pilots & Events',
];

```

A.21 Україномовний мовний файл навігації

```

<?php

return [
    'home' => 'Головна',
    'about' => 'Про проект',
    'news' => 'Новини',
    'partners' => 'Партнери',
    'resources' => 'Ресурси',
    'gallery' => 'Галерея',
    'get_involved' => 'Долучитися',
    'learning_platform' => 'Навчальна платформа',
    'courses' => 'Курси',
    'materials' => 'Матеріали',
    'quizzes' => 'Тести',
    'pilots_events' => 'Пілоти та події',
];

```

A.22 Фрагмент перемикача мов у Blade-шаблоні

```
<nav class="language-switcher">
    <a href="{{ route('locale.switch', 'en') }}">EN</a>
    <a href="{{ route('locale.switch', 'es') }}">ES</a>
    <a href="{{ route('locale.switch', 'uk') }}">UK</a>
    <a href="{{ route('locale.switch', 'el') }}">EL</a>
    <a href="{{ route('locale.switch', 'tr') }}">TR</a>
    <a href="{{ route('locale.switch', 'fi') }}">FI</a>
</nav>
```

A.23 Оброблення форми зворотного зв'язку

```
<?php

namespace App\Http\Controllers;

use App\Models\ContactRequest;
use Illuminate\Http\Request;

class ContactController extends Controller
{
    public function index()
    {
        return view('pages.get-involved');
    }

    public function store(Request $request)
    {
        $data = $request->validate([
            'name' => ['required', 'string', 'max:255'],
            'email' => ['required', 'email', 'max:255'],
            'subject' => ['nullable', 'string', 'max:255'],
            'message' => ['required', 'string', 'max:5000'],
            'privacy' => ['accepted'],
        ]);

        ContactRequest::create($data);

        return back()->with(
            'success',
            __('messages.contact_request_sent')
        );
    }
}
```

A.24 Модель повідомлення з форми зворотного зв'язку

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;

class ContactRequest extends Model
{
    use HasFactory;

    protected $fillable = [
        'name',
        'email',
        'subject',
        'message',
        'privacy',
    ];
}
```

```

];

protected $casts = [
    'privacy' => 'boolean',
];
}

```

A.25 Фрагмент Dockerfile

```

FROM php:8.3-fpm

WORKDIR /var/www/html

RUN apt-get update && apt-get install -y \
    git \
    unzip \
    zip \
    curl \
    libpng-dev \
    libonig-dev \
    libxml2-dev \
    && docker-php-ext-install pdo_mysql mbstring exif pcntl bcmath gd

COPY . .

RUN chown -R www-data:www-data /var/www/html/storage \
    /var/www/html/bootstrap/cache

CMD ["php-fpm"]

```

A.26 Фрагмент docker-compose.yml

```

version: '3.9'

services:
  app:
    build:
      context: .
      dockerfile: Dockerfile
    container_name: youthsoftpower_app
    volumes:
      - ../var/www/html
    depends_on:
      - mysql
    networks:
      - youthsoftpower

  mysql:
    image: mysql:8.0
    container_name: youthsoftpower_mysql
    restart: unless-stopped
    environment:
      MYSQL_DATABASE: youthsoftpower
      MYSQL_USER: youthsoftpower_user
      MYSQL_PASSWORD: secret
      MYSQL_ROOT_PASSWORD: root
    ports:
      - "3306:3306"
    volumes:
      - mysql_data:/var/lib/mysql
    networks:
      - youthsoftpower

  nginx:

```

```

    image: nginx:latest
    container_name: youthsoftpower_nginx
    ports:
      - "80:80"
    volumes:
      - ../var/www/html
      - ../docker/nginx/default.conf:/etc/nginx/conf.d/default.conf
    depends_on:
      - app
    networks:
      - youthsoftpower

volumes:
  mysql_data:

networks:
  youthsoftpower:
    driver: bridge

```

A.27 Фрагмент конфігурації підключення до бази даних

```

APP_NAME=YouthSOFTPOWER
APP_ENV=production
APP_DEBUG=false
APP_URL=https://youthsoftpower.wunu.edu.ua

DB_CONNECTION=mysql
DB_HOST=mysql
DB_PORT=3306
DB_DATABASE=youthsoftpower
DB_USERNAME=youthsoftpower_user
DB_PASSWORD=secret

DEEPL_API_KEY=your_deepl_api_key
DEEPL_API_URL=https://api-free.deepl.com/v2/translate

```

A.28 Сервіс перекладу контенту через DeepL

```

<?php

namespace App\Services;

use Illuminate\Support\Facades\Http;

class DeepLTranslationService
{
    public function translate(
        string $text,
        string $targetLanguage
    ): ?string {
        $response = Http::asForm()->post(config('services.deepl.url'), [
            'auth_key' => config('services.deepl.key'),
            'text' => $text,
            'target_lang' => strtoupper($targetLanguage),
        ]);

        if ($response->failed()) {
            return null;
        }

        return $response->json('translations.0.text');
    }
}

```

A.29 Приклад використання сервісу перекладу

```
<?php

namespace App\Actions;

use App\Models\News;
use App\Services\DeepLTranslationService;

class TranslateNewsAction
{
    public function __construct(
        private readonly DeepLTranslationService $translator
    ) {
    }

    public function execute(News $news, string $language): void
    {
        $translatedTitle = $this->translator->translate(
            $news->title,
            $language
        );

        $translatedContent = $this->translator->translate(
            $news->content,
            $language
        );

        $news->translations()->updateOrCreate(
            ['language' => $language],
            [
                'title' => $translatedTitle,
                'content' => $translatedContent,
            ]
        );
    }
}
```

A.30 Загальний CSS-фрагмент картки новини

```
.news-grid {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));
    gap: 24px;
}

.news-card {
    border: 1px solid #e5e7eb;
    border-radius: 12px;
    overflow: hidden;
    background: #ffffff;
}

.news-card img {
    width: 100%;
    height: 220px;
    object-fit: cover;
}

.news-card h3 {
    font-size: 20px;
    margin: 16px;
}

.news-card p {
```

```
    margin: 0 16px 16px;
    color: #4b5563;
}

.news-card a {
    display: inline-block;
    margin: 0 16px 16px;
    font-weight: 600;
}
```

A.31 CSS-фрагмент адаптивного меню

```
.header {
    display: flex;
    align-items: center;
    justify-content: space-between;
    padding: 20px 0;
}

.navigation {
    display: flex;
    gap: 24px;
}

.language-switcher {
    display: flex;
    gap: 8px;
}

@media (max-width: 768px) {
    .navigation {
        flex-direction: column;
        gap: 12px;
    }

    .header {
        align-items: flex-start;
    }
}
```

www.konferenciaonline.org.ua

Міжнародна наукова
інтернет-конференція

**Інформаційне суспільство:
технологічні, економічні
та технічні аспекти становлення**

Випуск 110

ISSN 2522-932X

Google Scholar



AKADEMIA NAUK STOSOWANYCH
WYŻSZA SZKOŁA ZARZĄDZANIA I ADMINISTRACJI
W OPOLE

7-8 травня 2026 р.

м. Тернопіль, Україна – м. Ополе, Польща
2026

УДК 001 (063)

Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 110): матеріали Міжнародної наукової інтернет-конференції, (м. Тернопіль, Україна, м. Ополе, Польща, 7-8 травня 2026 р.) / редкол. : О. Патряк та ін. ГО "Наукова спільнота", WSZIA w Opolu. Тернопіль : ФО-П Шпак В.Б. 2026. 195 с. – ISSN 2522-932X

Збірник доповідей підготовлено за матеріалами Міжнародної наукової інтернет-конференції (випуск 110) 7-8 травня 2026 р. на сайті www.konferenciaonline.org.ua

Оргкомітет ГО Наукова спільнота:

Патряк Олександра Тарасівна, кандидат економічних наук, ЗУНУ;

Шевченко (Огінська) Анастасія Юріївна, кандидат економічних наук, директор ТОВ «Школа майбутнього»;

Назарчук Оксана Михайлівна, доктор філософії (Ph.D.), ННІ «Юридичний інститут КНЕУ імені Вадима Гетьмана»;

Гомотюк Оксана Євгенівна, доктор історичних наук, професор, ЗУНУ;

Біловус Леся Іванівна, доктор історичних наук, кандидат філологічних наук, професор, ЗУНУ;

Рибуха Лілія Зіновіївна, доктор педагогічних наук, кандидат психологічних наук, професор, ЗУНУ;

Недошитко Ірина Романівна, кандидат історичних наук, доцент, ЗУНУ;

Стефанишин Олена Василівна, кандидат історичних наук, доцент, ЗУНУ;

Яблонська Наталія Мирославівна, кандидат філологічних наук, старший викладач, ЗУНУ;

Рудакевич Оксана Мирославівна, кандидат філософських наук, ЗУНУ;

Русенко Святослав Ярославович, викладач ВСІП ФКЕПТ ЗУНУ.

Тексти матеріалів конференції подаються в авторській редакції. Відповідальність за точність, достовірність і зміст поданих матеріалів несуть автори. Всі роботи ліцензуються відповідно до Creative Commons Attribution 4.0 International License.

Автори зберігають авторське право, а також надають збірнику право першого опублікування оригінальних наукових статей на умовах ліцензії Creative Commons Attribution 4.0 International License, що дозволяє іншим розповсюджувати роботу з визнанням авторства твору та першої публікації в цьому збірнику.

Наша адреса: Оргкомітет МНІК "Конференція онлайн"

а/с 797, м. Тернопіль 46005

тел. моб. 068 366 0 525

e-mail: inetkonf@ukr.net

URL Інтернет-конференції: <http://www.konferenciaonline.org.ua/>

ISSN 2522-932X

© ГО "Наукова спільнота" 2026

© Автори статей 2026



| | |
|---|-----------|
| Віпшовський Юрій Андрійович, Грицюк Юрій Іванович
ПРОСТОРОВА ЛОКАЛІЗАЦІЯ АНОМАЛІЙ НА МЕТАЛЕВИХ
ПОВЕРХНЯХ ЗА ДОПОМОГОЮ АНАЛІЗУ ОДНОВИМІРНИХ
ПРОЕКЦІЙ ІНТЕНСИВНОСТІ..... | 24 |
| Гуцул Олександр Станіславович, Танасюк Юлія Володимирівна
ІНТЕЛЕКТУАЛЬНИЙ МОБІЛЬНИЙ ЗАСТОСУНОК ДЛЯ
ЦИФРОВОЇ АВТОМАТИЗАЦІЇ ТРЕНАЖЕРНИХ ЗАЛІВ..... | 27 |
| Дерев'ягин Ярослав Вікторович
АПРОКСИМАЦІЯ КРИВИХ ДЛЯ АВТОМАТИЗАЦІЇ
ГРАФОАНАЛІТИЧНОГО РОЗРАХУНКУ | 32 |
| Комар Мирослав Петрович, Горбаль Юлія-Марія Мар'янівна
ПРОЄКТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ
СИСТЕМИ КЕРУВАННЯ КОНТЕНТОМ МІЖНАРОДНОГО
ОСВІТНЬОГО ПРОЄКТУ..... | 35 |
| Костоглод Анастасія Юрійівна
АРХІТЕКТУРА ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ ОПТИМІЗАЦІЇ
РЕЖИМІВ РОБОТИ ІОТ-ПРИСТРОЇВ..... | 38 |
| Красовський Андрій Олександрович
ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ МАШИННОГО
НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ НАРАТИВІВ..... | 41 |
| Майків Ігор Мирославович, Стецюк Арсен Ігорович
АВТОМАТИЗАЦІЯ ЗРОШЕННЯ НА БАЗІ ДАНИХ ІОТ-СЕНСОРІВ
ТА МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ..... | 46 |
| Майків Ігор Мирославович, Цьомик Олег Ігорович | |

Література:

Mattingly, J. Aircraft engine design / J. Mattingly, H. Heiser, T. Pratt. – Washington: AIAA, DC. – 2002. – 687 p.

Гусев, А. В. Математичне моделювання в техніці / А. В. Гусев, С. В. Гусев. Київ: Юрайт, 2023. – 352 с.

*Комар Мирослав Петрович,
доктор технічних наук, професор,
Західноукраїнський національний університет, м. Тернопіль*

*Горбаль Юлія-Марія Мар'янівна, бакалавр,
Західноукраїнський національний університет, м. Тернопіль*

ПРОЄКТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ КЕРУВАННЯ КОНТЕНТОМ МІЖНАРОДНОГО ОСВІТНЬОГО ПРОЄКТУ

Інтернет-адреса публікації на сайті:

<http://www.konferenciaonline.org.ua/ua/article/id-2586/>

У сучасних умовах цифрової трансформації освітнього середовища веб-орієнтовані інформаційні системи набули важливого значення для організації, супроводу та поширення матеріалів міжнародних освітніх ініціатив. Зростання кількості міжнародних програм, партнерських мереж, навчальних платформ, інформаційних кампаній і спільних цифрових ресурсів зумовило потребу у створенні зручних засобів централізованого керування контентом. Особливої актуальності це набуває для міжнародних освітніх проєктів, у межах яких необхідно забезпечити регулярне оновлення новин, публікацію матеріалів, підтримку багатомовності, представлення партнерів, висвітлення заходів, а також доступність інформації для різних категорій користувачів [1].

Традиційне розміщення матеріалів без єдиного підходу до їх структурування, оновлення та адміністрування ускладнює інформаційний супровід проєкту. За відсутності спеціалізованої веб-орієнтованої системи виникають труднощі з підтримкою актуальності контенту, його логічною організацією, повторним використанням, розмежуванням прав доступу та забезпеченням однакового стилю подання інформації. У таких умовах система керування контентом розглядається не лише як технічний інструмент публікації матеріалів, а як засіб упорядкування інформаційних потоків, підтримки комунікації та представлення результатів міжнародного освітнього проєкту в цифровому середовищі [1, 2].

Актуальність теми визначається також тим, що сучасна веб-орієнтована інформаційна система має відповідати не лише функціональним вимогам, а й вимогам адаптивності, безпеки, масштабованості, зручності використання та доступності веб контенту. Під час проєктування таких систем враховуються підходи до організації веб контенту, принципи адаптивного дизайну, вимоги до веб форм, правила міжнародного та багатомовного представлення інформації, а також рекомендації щодо веб доступності й захисту даних. Це зумовлює необхідність комплексного підходу, у межах якого поєднуються засоби веб розробки, моделі керування контентом, архітектурні рішення та практичні методи реалізації.

До функціональних вимог системи належать: відображення основних розділів сайту; публікація новин, сторінок, матеріалів, даних про партнерів і галереї; створення, редагування та видалення контенту через адміністративну панель; підтримка багатомовного інтерфейсу; переклад контенту після створення; динамічне підвантаження списків даних; робота форм зворотного зв'язку; розмежування прав доступу для адміністраторів і редакторів.

До нефункціональних вимог належать: зрозумілий і адаптивний інтерфейс; зручність адміністрування; швидке завантаження сторінок; цілісність зберігання даних; безпечне оброблення введеної інформації; можливість розгортання системи в уніфікованому середовищі; придатність до подальшого розширення.

Архітектура системи побудована за багаторівневим принципом. Основу серверної частини становить Laravel 12, який забезпечує маршрутизацію, оброблення запитів, роботу з моделями та бізнес-логікою. Формування сторінок публічної частини виконується за допомогою Blade-шаблонів із використанням HTML, CSS і JavaScript. Для зберігання структурованих даних використовується MySQL. Адміністративну панель реалізовано засобами Filament v5, а динамічне підвантаження даних без повного перезавантаження сторінки забезпечується через Livewire 4. Розгортання системи здійснюється за допомогою Docker.

У структурі системи виділено такі основні компоненти (рисунк 1):

- публічна частина сайту – забезпечує доступ користувачів до інформаційних сторінок, новин, матеріалів і галереї;
- адміністративна панель – використовується для керування контентом;

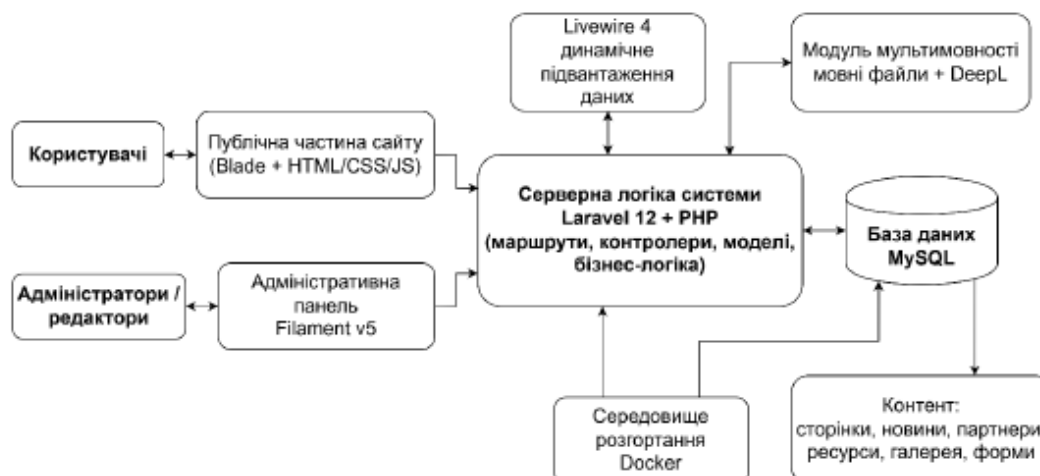


Рисунок 1 – Архітектура веб-орієнтованої інформаційної системи керування контентом міжнародного освітнього проєкту

- модуль мультимовності – відповідає за статичні переклади та переклад контенту;
- модуль динамічної взаємодії – забезпечує часткове оновлення сторінок;
- база даних – зберігає контентні сутності, користувачів і службову інформацію;
- середовище розгортання – забезпечує запуск системи в контейнеризованій інфраструктурі.

Отже, сформовано цілісне архітектурне та функціональне рішення, орієнтоване на централізоване адміністрування інформаційних матеріалів, підтримку багатомовності та зручне подання контенту в публічній частині сайту.

Література:

1. Vlasenko K. V., Volkov S. V., Lovianova I. V., Chumak O. O., Sitak I. V. and Bobyliev D. Y. Management of online platform development and support process. CTE Workshop Proceedings. 2021. Vol. 8. P.333-345.
2. Rola P., Kuchta D. A Content Management System as an Information Management System in Interdisciplinary Research. *Acta Universitatis Lodzianensis. Folia Oeconomica*. 2020. Vol. 5(350). P. 65-81.