

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики
(повне найменування назва факультету)

Кафедра інформаційних систем
(повна назва кафедри)

ДИПЛОМНА РОБОТА

Розробка системи управління Веб-публікаціями

Виконав: студент групи ПМІ-42
спеціальності 122 – комп'ютерні науки
(шифр і назва спеціальності)

Сивуляк Н. Н.

_____ (підпис)

_____ (прізвище та ініціали)

Керівник _____ Горлач В. М. _____
(підпис) (прізвище та ініціали)

Рецензент _____
(підпис) (прізвище та ініціали)

ЗМІСТ

ВСТУП.....	3
ЗАВДАННЯ ТА ВИМОГИ ДО ВЕБ-САЙТУ.....	5
Розділ 1. Архітектура проєкту.....	6
1.1 Use-case діаграма.....	6
1.2 Проєктування бази даних.....	7
1.3 Архітектурний шаблон коду.....	10
Розділ 2. Вибір стеку технологій.....	11
PHP.....	12
Laravel.....	13
HTML/CSS.....	14
JavaScript.....	14
MySQL.....	15
Figma.....	16
Розділ 3 Демонстрація роботи та коду проєкту.....	17
3.1 Реєстрація нового користувача.....	17
3.2 Авторизація.....	18
3.3 Скидання паролю.....	19
3.4 Головна сторінка веб-сайту.....	20
3.5 Перегляд окремої публікації.....	21
3.8 Створення публікації.....	22
3.7 Редагування публікації.....	23
3.8 Сортування публікацій по категоріях.....	23
3.9 Відображення всіх категорій.....	24
3.10 Відображення певної категорії.....	25
3.11 Створення та редагування нової категорії.....	26
3.12 Видалення категорії.....	27
3.12 Перегляд всіх створених дизайнів.....	27
3.13 Створення нового дизайну.....	28
3.14 Редагування існуючого дизайну.....	30
3.15 Перегляд всіх створених пунктів меню.....	31
3.16 Перегляд конкретного пункту.....	31
3.17 Редагування конкретного пункту.....	32
3.18 Перегляд всіх користувачів системи.....	33
3.19 Перегляд конкретного користувача.....	33
3.20 Редагування користувача.....	34
3.21 Відображення меню в шапці сайту.....	34
3.22 Створена сторінка з пункту меню.....	35
ВИСНОВОК.....	36
Список використаних джерел.....	37

ВСТУП

З розвитком Інтернету та веб-технологій, все більше компаній та організацій починають бачити потребу у власному веб-сайті та системі управління контентом. Такі сайти та системи дозволяють представляти своїм клієнтам та користувачам необхідну та актуальну інформацію та послуги. Будування системи управління веб-публікаціями є важливим етапом для будь-якої організації, що бажає забезпечити ефективне використання своїх ресурсів та надати своїм клієнтам актуальну та якісну інформацію.

Система управління веб-публікаціями (Content Management System, CMS) є програмним забезпеченням, яке дозволяє автоматизувати процеси створення, редагування та публікації веб-контенту. CMS дозволяє користувачам з різних департаментів та відділів оновлювати веб-сайт без спеціальних знань програмування та дизайну. Вона надає можливість створювати та редагувати контент, який може використовуватися на веб-сайті, включаючи текст, зображення, відео та аудіофайли.

Однією з переваг CMS є можливість забезпечити єдиний стиль та форматування веб-контенту на всьому сайті, що забезпечує його зрозумілість та легкість у використанні. Крім того, CMS дозволяє зберігати та організувати вміст сайту в логічних категоріях, що полегшує навігацію та пошук інформації для користувачів.

Іншою важливою перевагою CMS є можливість автоматичної оптимізації контенту для пошукових систем, що покращує видимість сайту в результатах пошуку та забезпечує більше відвідувань на веб-сайті.

За допомогою CMS можна ефективно використовувати ресурси та надавати своїм клієнтам та користувачам якісну та актуальну інформацію. Крім того, система дозволяє зменшити час та зусилля на оновлення інформації та забезпечити більш ефективне використання ресурсів.

Для розробки CMS у цій дипломній роботі вирішено використовувати потужний PHP фреймворк Laravel, який забезпечує

швидку розробку та підтримку великих проектів завдяки своїм вбудованим функціям та засобам, таким як міграції баз даних, маршрутизація та засоби аутентифікації.

ЗАВДАННЯ ТА ВИМОГИ ДО ВЕБ-САЙТУ

Потрібно розробити зручну CMS панель для адміністраторів, яка дозволить їм створювати публікації на веб-сайті, змінювати дизайн та відносити їх до певних категорій та пунктів меню, змінювати рівень доступу у користувачів. Користувачі, які не є адміністраторами та редакторами, повинні мати можливість переглядати опубліковані публікації та користуватися меню. Користувач з доступом “редактор” матиме можливість редагувати всі публікації, категорії, дизайни та пункти меню.

CMS панель буде розроблена з використанням PHP, MySQL та JavaScript технологій. При розробці слід звернути особливу увагу на забезпечення безпеки від злому та хакерських атак, використовуючи захист від SQL ін'єкцій та підтвердження прав доступу для кожного користувача.

CMS панель повинна мати наступні можливості:

- Зручна адмін панель для адміністрування елементів веб-сайту
- Авторизація та реєстрація користувачів
- Створення, редагування категорій, публікацій, дизайнів, пунктів меню
- Перегляд та редагування користувачів системи адміністратором
- Реалізувати пошук публікації за назвою та категоріями

Загальний вигляд CMS панелі повинен бути інтуїтивно зрозумілим та зручним у використанні. Ця CMS панель дозволить адміністраторам легко публікувати та редагувати публікації, встановлювати права доступу для користувачів, відносити їх до певних категорій та додавати теги. Крім того, користувачі зможуть легко знайти цікаву інформацію за допомогою пошуку за категоріями та тегами. Також, необхідно створити use-case діаграму та схему бази даних.

Розділ 1. Архітектура проєкту

1.1 Use-case діаграма

Для того, щоб мати належне розуміння можливостей, необхідних для роботи з майбутньою CMS, була розроблена use-case діаграма, яка відображає всі можливі взаємодії користувачів з системою.

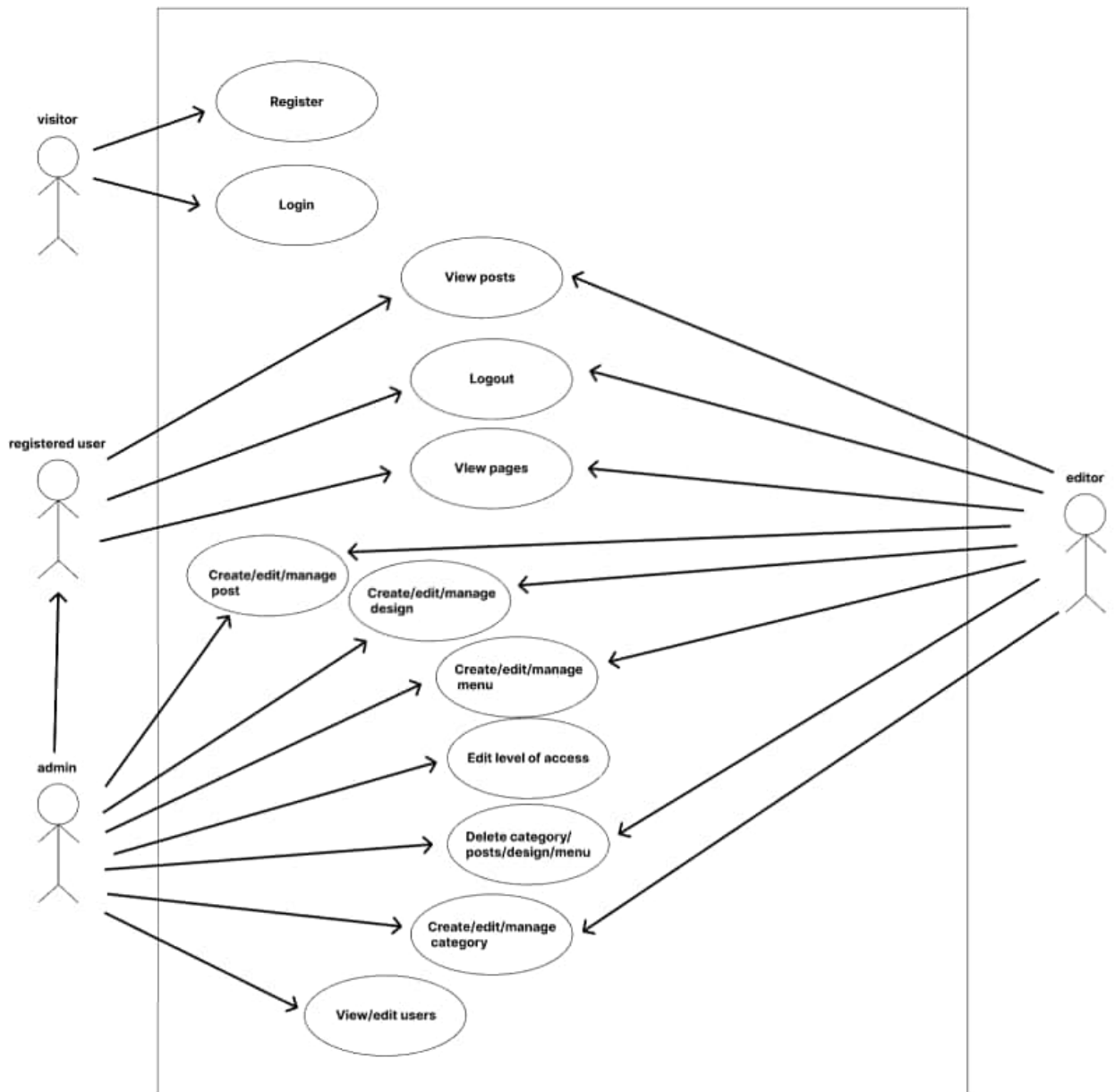
Діаграма включає наступні основні фактори:

- Адміністратор(admin): особа, яка відповідає за керування системою, створення категорій/публікацій/дизайнів/пунктів меню, редагування користувачів та надання доступу до певних функцій.
- Редактор: особа, яка редагує систему, створює категорії/публікації/дизайни/пункти меню
- Зареєстрований користувач: користувач з правами доступу для перегляду публікацій, категорій, сторінок веб-сайту
- Не зареєстрований користувач: особа, яка потрапила на сайт вперше або не увійшла до свого акаунту.

Крім того, на діаграмі представлені наступні основні взаємодії користувачів з системою:

- Додавання контенту на сайт.
- Редагування вже наявного контенту на сайті.
- Видалення контенту з сайту.
- Перегляд контенту на сайті.
- Реєстрація нового користувача на сайті.
- Авторизація зареєстрованого користувача на сайті.
- Керування налаштуваннями сайту адміністратором.

Ця use-case діаграма дозволяє наочно представити всі можливості користувачів при взаємодії з майбутньою CMS, а також допоможе мені правильно спланувати та реалізувати всі необхідні функціональні можливості системи.



(Рисунок 1.1)

1.2 Проєктування бази даних

Більшість проєктів не може обійтися без бази даних, мій не є виключенням. Першим кроком при проєктуванні бази даних є збір вимог до неї. Ці вимоги можуть включати різні типи даних, які потрібно зберігати, а також методи доступу до цих даних. Наприклад, для нашого проєкту ми можемо потребувати збереження імен користувачів, їхніх електронних адрес, інформацію про категорії, тощо. Крім того, ми повинні визначити та створити можливість

збереження інформації про доступи користувачів до веб-сайту.



(Рисунок 1.2)

Для забезпечення ефективного зберігання даних у моєму проєкті, я розробив шість таблиць, які будуть створені в базі даних:

Users

Ця таблиця буде зберігати інформацію про користувачів мого сайту. Вона

міститиме поля для зберігання даних про кожного користувача, такі як ім'я, електронну пошту, пароль, дату реєстрації та останнього оновлення, рівень доступу.

Designs

У цій таблиці зберігатимуться дані про стилі оформлення публікацій для мого сайту. Вона міститиме поля для зберігання інформації про кожен дизайн, такі як назва, кольори та шрифти текстів, кольори тексту категорії, фону та чи відображати список категорії.

Migrations

Табличка Migrations призначена для зберігання історії міграцій бази даних. Вона міститиме поля, які дозволять зберігати дані про кожну міграцію, включаючи назву, дату створення та статус виконання.

Posts

У цій таблиці будуть зберігатися дані про пости на моєму сайті. Вона міститиме поля для зберігання інформації про кожен пост, такі як заголовок, підзаголовок, основний текст, категорії до яких належить пост, дата публікації та останній час оновлення.

Categories

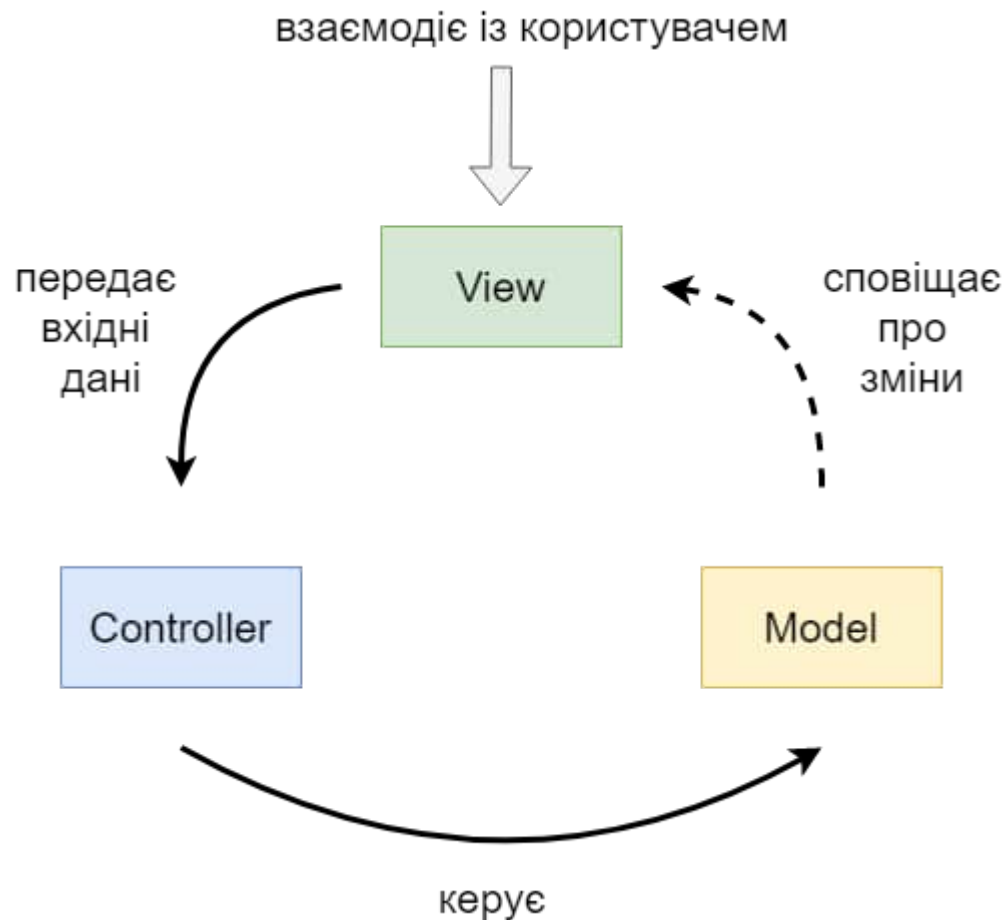
Ця таблиця міститиме інформацію про категорії, які я використовуватиму для організації та сортуванню публікацій на моєму сайті. Вона містить поля для зберігання назв категорій.

Menus

Ця таблиця міститиме інформацію про пункти меню: їхню назву, відображення в навігації веб-сайту.

Загалом, ці шість таблиць дозволять мені ефективно зберігати та організувати дані на моєму сайті.

1.3 Архітектурний шаблон коду



(Рисунок 1.3)

В розробці веб-сайтів був використаний основний архітектурний патерн - Model-View-Controller (MVC), який знайшов широке застосування в багатьох технологіях та системах. Перший значний приклад втілення MVC з'явився у мові програмування SmallTalk, коли розробники поставили перед собою завдання розділити графічний інтерфейс від бізнес-логіки та бізнес-логіку від даних. Отже, в класичному варіанті, MVC складається з цих трьох ключових компонентів.

Модель відповідає за управління даними та бізнес-логікою. Вона зберігає дані, виконує операції з ними та надає інтерфейс для їх доступу та зміни. Модель представляє собою абстракцію даних і бізнес-логіки, що дозволяє розділити їх від решти системи.

Представлення відповідає за відображення даних користувачу та взаємодію з ним. Воно відповідає за створення графічного інтерфейсу, відображення даних

у зручному форматі та обробку користувацьких дій. Представлення не має власної логіки, воно лише відображає дані з моделі та передає користувацькі дії контролеру.

Контролер відповідає за керування потоком даних та взаємодію між моделлю та представленням. Він обробляє користувацькі дії, виконує відповідні операції з моделлю та оновлює представлення. Контролер дозволяє розподілити логіку програми між моделлю та представленням, забезпечуючи їх взаємодію без прямого зв'язку.

Застосування патерну MVC дозволяє створювати добре організовані, модульні та легко змінювані веб-додатки. Розділення логіки додатку на модель, представлення та контролер дозволяє підтримувати різні варіації інтерфейсу та змінювати логіку без впливу на інші компоненти системи.

Розділ 2. Вибір стеку технологій

При виборі стеку технологій для розробки проекту, було враховано велика кількість факторів, таких як вимоги до проекту описані раніше в цьому документі, мій досвід у написанні веб-додатків, потребу масштабованості у майбутньому, сумісність всього стеку технологій та швидкодія проекту.

Значна увага приділялась врахуванню специфічних вимог до функціональності і надійності проекту. Було розглянуто можливості технологій у розширенні функціоналу, забезпеченні безпеки даних та підтримці високої продуктивності. Враховувалися вимоги до інтеграції з існуючими системами або сервісами, а також можливості швидкого розвитку та майбутнього розширення проекту. Для ефективного створення проекту я також враховував мої власні знання і досвід у використанні технологій. Використання знайомих технологій дозволить ефективніше вирішувати завдання та впроваджувати нові функції, зменшуючи час, необхідний для розробки і розгортання. Окрім цього, було важливо забезпечити сумісність між усіма компонентами стеку технологій. Це означає, що обрані технології повинні взаємодіяти між собою без проблем, забезпечуючи надійний обмін даними та взаємодію користувача з системою.

Сумісність між різними технологіями є ключовим фактором для успішного написання проекту та забезпечення його функціональності. Крім того, швидкодія проекту також залежить від сумісності технологій. Вибір оптимізованих технологій, які дозволяють працювати з великими обсягами даних та забезпечувати ефективну обробку запитів, може позитивно вплинути на роботу системи і задовольнити вимоги щодо продуктивності.

У підсумку, було обраний такий стек технологій:

PHP

PHP є мовою програмування, яка використовується для створення веб-додатків. Вона має широкі можливості та гнучкість, що дозволяє розробникам швидко та ефективно створювати веб-сайти та додатки різної складності. PHP має велику кількість фреймворків, таких як Laravel, Symfony, CodeIgniter та інші, що забезпечують швидку та просту розробку. Фреймворки дозволяють використовувати готові компоненти та шаблони, що спрощує процес розробки і підвищує продуктивність команди.

Одним з найпопулярніших фреймворків для PHP є Laravel. Він надає зручні інструменти для роботи з базами даних, маршрутизації, аутентифікації, кешування та багатьма іншими функціями, що спрощують процес розробки і підвищують якість коду. Laravel також має широку спільноту розробників, яка постійно поповнює фреймворк новими розширеннями та плагінами.

PHP також відомий своєю високою продуктивністю та швидкодією. Завдяки вбудованим оптимізаціям та кешуванню, PHP може ефективно обробляти запити та швидко відповідати на них. Крім того, PHP має багато розширень та бібліотек, які дозволяють розробникам розширити його функціональність та пристосувати до своїх потреб.

Загалом, використання PHP у стеку технологій для розробки проекту має багато переваг. Вона надає широкі можливості для швидкої та ефективної розробки веб-додатків, має велику кількість фреймворків для спрощення роботи та забезпечення безпеки, а також дозволяє досягти високої продуктивності та

швидкодії проекту.

Laravel

Laravel є однією з найбільш популярних та ефективних фреймворків для розробки веб-додатків. Він має широкий спектр можливостей та гнучкість, що дозволяє розробникам швидко та ефективно створювати веб-сайти та додатки різної складності.

Однією з ключових переваг використання Laravel є його вбудована підтримка шаблонізатора Blade. Blade є потужним інструментом, який дозволяє відокремити логіку та представлення веб-сторінок. Використання Blade спрощує процес створення шаблонів та включення динамічних даних в HTML-код. Завдяки простому та інтуїтивно зрозумілому синтаксису, розробники можуть швидко створювати вигляд сторінок та управляти даними, що виводяться.

Шаблонізатор Blade також надає розширені можливості, такі як наслідування шаблонів (layouts) та використання компонентів (components) для створення повторюваних елементів інтерфейсу. Це дозволяє створювати чистий та добре організований код, а також спрощує процес розробки та підтримки проекту.

Крім того, Laravel і Blade активно підтримуються та оновлюються спільнотою розробників. Це означає, що у вас є доступ до багатьох корисних розширень, плагінів та ресурсів, які допоможуть вам ефективніше розробляти проект.

Загалом, поєднання Laravel і шаблонізатора Blade надає розробникам потужні інструменти для створення веб-додатків з високою продуктивністю, безпекою та гнучкістю. Вони допомагають прискорити процес розробки, спростити управління даними та представленням, а також забезпечити чистий та організований код.

HTML/CSS

HTML та CSS є основними мовами для створення веб-сторінок, і вони

відіграють важливу роль у розробці проекту. HTML (HyperText Markup Language) використовується для структурування та організації вмісту веб-сторінок. Він надає елементи, такі як заголовки, абзаци, списки, зображення, посилання та інші, які дозволяють відтворити бажаний контент на сторінці.

CSS (Cascading Style Sheets) відповідає за зовнішній вигляд веб-сторінок. Використовуючи CSS, розробники можуть задавати стилі, кольори, розміри, відступи та інші властивості елементів HTML. Це дозволяє керувати виглядом сторінки і створювати привабливі та професійні дизайни.

Поєднання HTML та CSS дозволяє розробникам створювати різноманітні макети та дизайни, що відповідають вимогам проекту. Вони забезпечують зручність та ефективність для користувача, оскільки дозволяють легко сприймати та навігувати по сторінкам. Будучи легкими у використанні, HTML та CSS також дозволяють розробникам швидко та просто робити зміни в дизайні та макетах сторінок. Це забезпечує гнучкість та швидкість розробки, особливо коли потрібно вносити коригування або адаптувати веб-сторінки до різних пристроїв та екранів.

HTML та CSS є незамінними компонентами веб-розробки. Вони дозволяють створювати зручні та ефективні веб-сторінки з привабливим дизайном, а також забезпечують легкість управління та модифікації. Знання та використання HTML та CSS допоможуть вам створити якісний та професійний проект.

JavaScript

JavaScript є надзвичайно потужною мовою програмування, яка використовується для розробки динамічних та інтерактивних веб-сторінок. Він надає розробникам безліч можливостей для створення складних функцій та взаємодії з користувачем, що робить веб-сайти більш зручними та ефективними у використанні.

JavaScript використовується для реалізації різноманітних функцій, таких як валідація форм, анімація елементів, реакція на події користувача, взаємодія з

сервером через AJAX і багато іншого. Завдяки JavaScript, розробники можуть створювати багатофункціональні веб-додатки, які реагують на дії користувача миттєво та надають інтерактивність, що поліпшує загальний досвід використання веб-сайту.

Ще однією перевагою JavaScript є його широке використання та підтримка. Він є стандартом веб-розробки та підтримується всіма сучасними веб-браузерами. Крім того, існує велика спільнота розробників, яка активно співпрацює та ділиться знаннями, бібліотеками та фреймворками, що полегшують розробку та підтримку проектів на JavaScript.

У підсумку, JavaScript відіграє важливу роль у веб-розробці, дозволяючи створювати динамічні, інтерактивні та високофункціональні веб-сторінки. Використання JavaScript розширює можливості проекту, роблячи його більш привабливим та користувачозорієнтованим. Знання JavaScript є важливим для сучасного розробника, оскільки воно відкриває безл

MySQL

MySQL є однією з найбільш популярних та потужних систем управління баз даних (СУБД). Вона забезпечує надійне зберігання, організацію та керування великим обсягом даних. MySQL використовується для забезпечення швидкого та ефективного доступу до інформації, що робить його незамінним компонентом веб-розробки.

Одна з важливих переваг MySQL полягає у його високому рівні безпеки та стабільності. Він має вбудовані механізми для захисту даних, такі як автентифікація, контроль доступу та шифрування. Це дозволяє розробникам впевнено працювати з конфіденційною інформацією та забезпечувати безпеку веб-додатку.

MySQL також відомий своєю надійністю та масштабованістю. Він може обробляти великий потік запитів та підтримувати велику кількість одночасних підключень. Це особливо важливо для проектів, які очікують зростання обсягу даних та виконання запитів з багатьма користувачами одночасно.

Знання MySQL дозволяють розробникам ефективно створювати, оптимізувати та керувати базами даних для своїх веб-додатків. Використання MySQL разом з іншими компонентами стеку технологій забезпечує зручне та потужне середовище для розробки веб-проектів.

MySQL є надійною, безпечною та масштабованою системою управління баз даних, яка грає важливу роль у веб-розробці. Використання MySQL допомагає забезпечити швидкий доступ до даних, надійність веб-додатку та ефективно управління інформацією.

Figma

Figma є потужним та інноваційним інструментом для дизайну та прототипування веб-інтерфейсів. Він надає можливість розробникам та дизайнерам співпрацювати в реальному часі, створювати векторні графічні елементи, встановлювати взаємозв'язки між сторінками та взаємодіями, а також тестувати прототипи з реальними користувачами.

Одна з головних переваг Figma полягає в його хмарній структурі, що дозволяє зручно спільно працювати над проектами. Користувачі можуть одночасно редагувати дизайн, додавати коментарі та спілкуватися в реальному часі. Це забезпечує ефективну колаборацію та уникнення проблем зі синхронізацією файлів між різними користувачами. Figma також підтримує велику кількість розширень та інтеграцій з іншими інструментами. Це дозволяє розробникам і дизайнерам працювати в їхньому улюбленому середовищі та використовувати додаткові функціональні можливості для покращення робочого процесу.

Figma також має велику спільноту користувачів, де можна знайти безліч ресурсів, шаблонів та навчального матеріалу. Це дозволяє розробникам та дизайнерам швидше впроваджувати нові ідеї, використовуючи готові рішення та кращі практики.

Загалом, Figma є потужним та універсальним інструментом для дизайну та прототипування веб-інтерфейсів. Його зручний інтерфейс, можливості спільної

роботи та розширення робочих можливостей роблять його вибором багатьох професіоналів у сфері дизайну та розробки.

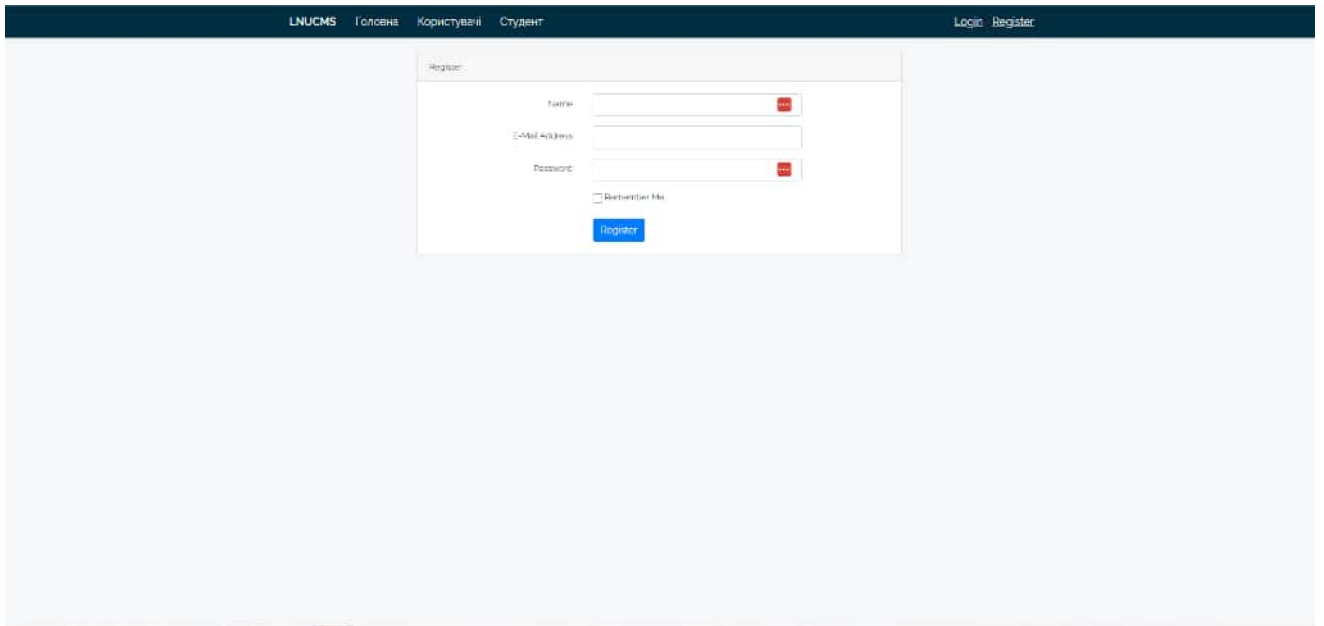
Розділ 3 Демонстрація роботи та коду проєкту

Для написання та редагування коду я використовував IDE-у PHPStorm. Це є потужним інструментом для написання та редагування коду. Вона надає розширену підтримку для мови програмування PHP та інших технологій, що використовуються веб-розробкою. PHPStorm має багато корисних функцій, таких як автодоповнення коду, перевірка синтаксису, налагоджування коду, вбудовану систему контролю версій та багато іншого. Вона також надає зручне інтерфейсне середовище, яке сприяє продуктивній роботі розробника та полегшує взаємодію з проєктом. З використанням PHPStorm розробка веб-додатків стає швидшою, ефективнішою та більш організованою.

3.1 Реєстрація нового користувача

У розробленому веб-додатку був реалізований функціонал реєстрації нового користувача. Користувач має можливість створити свій обліковий запис шляхом заповнення реєстраційної форми. Під час реєстрації користувач повинен вказати свої особисті дані, такі як ім'я, прізвище, електронну пошту та обрати унікальне користувача та пароль.

Після введення необхідної інформації та натискання кнопки "Зареєструватися", дані користувача перевіряються на валідність. Якщо введені дані відповідають вимогам системи, то обліковий запис користувача створюється:

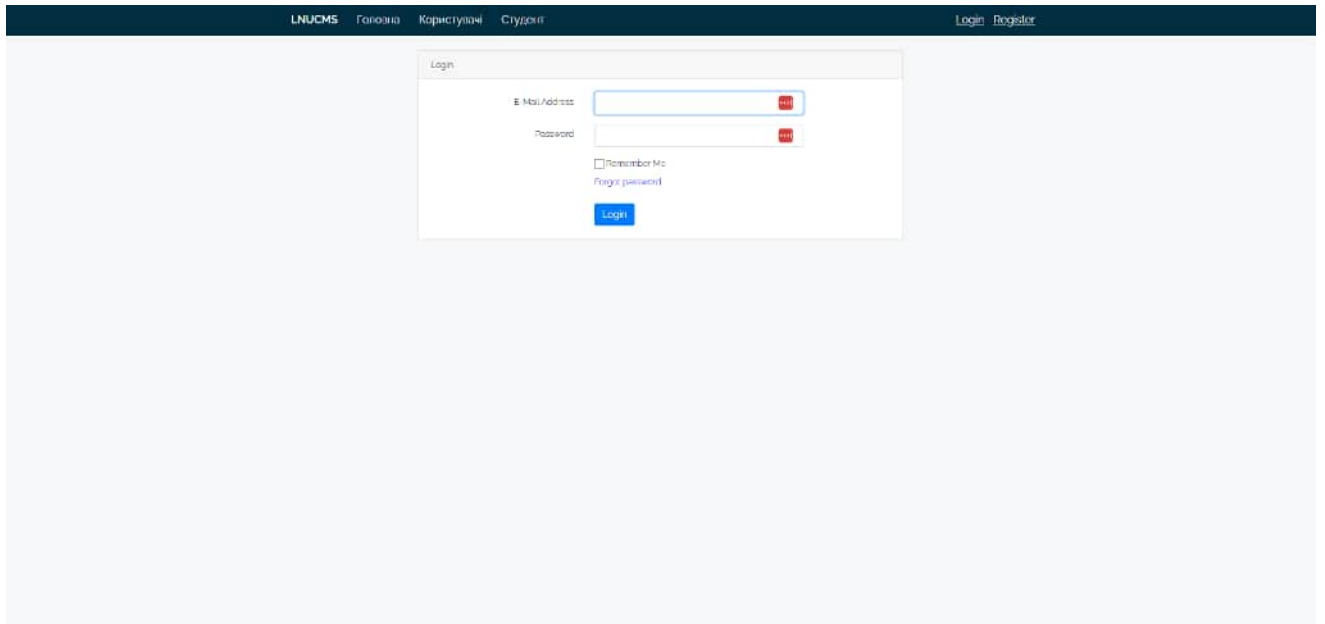


(Рисунок 1.4)

3.2 Авторизація

Раніше зареєстрований користувач може ввести свої облікові дані, такі як: електронна адреса та пароль, які користувач використовував при реєстрації. Після введення облікових даних, веб-додаток перевіряє їх правильність та наявність у базі даних. Якщо введені дані вірні і користувач зареєстрований в системі, то відбувається процес авторизації, і користувач отримує доступ до свого особистого облікового кабінету або інших функцій, залежно від ролі користувача. У разі неправильного введення облікових даних або відсутності користувача в системі, веб-додаток повідомляє про помилку та надає можливість повторного введення або відновлення пароля. Скинути пароль можна одразу, натиснувши на відповідну фразу після полів для авторизації - “forgot password”.

Також реалізована функція “Запам’ятати мене”, щоб система не виходила з облікового запису користувача автоматично.

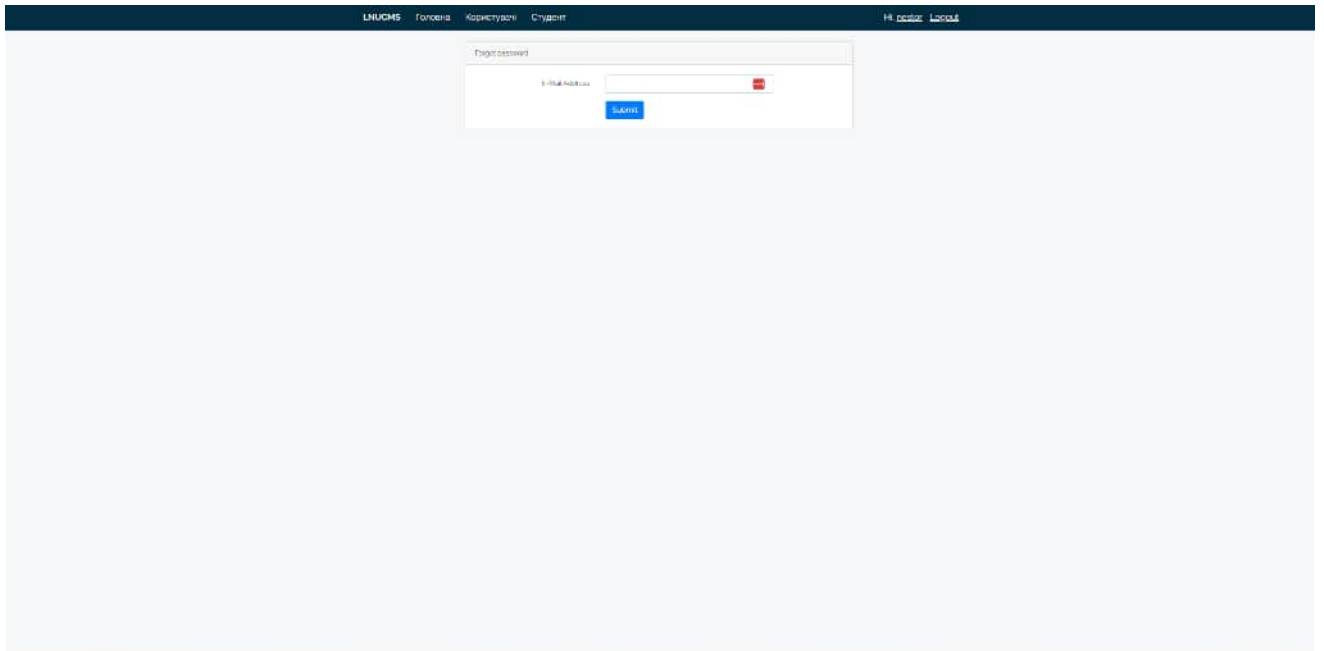


(Рисунок 1.5)

3.3 Скидання паролю

Реалізований функціонал скидання паролю для користувачів. Якщо користувач забув свій пароль, він може використати функцію скидання паролю, щоб відновити доступ до свого облікового запису. Для скидання паролю користувачу потрібно ввести свою електронну пошту на відповідній формі.

Після введення пошти і відправлення форми, веб-додаток генерує унікальне посилання для скидання паролю та надсилає його користувачу на вказану поштову скриньку. Користувач отримує електронного листа з інструкціями та посиланням для скидання паролю. Це посилання містить унікальний ідентифікатор, який використовується для ідентифікації користувача та підтвердження його запиту на скидання паролю.



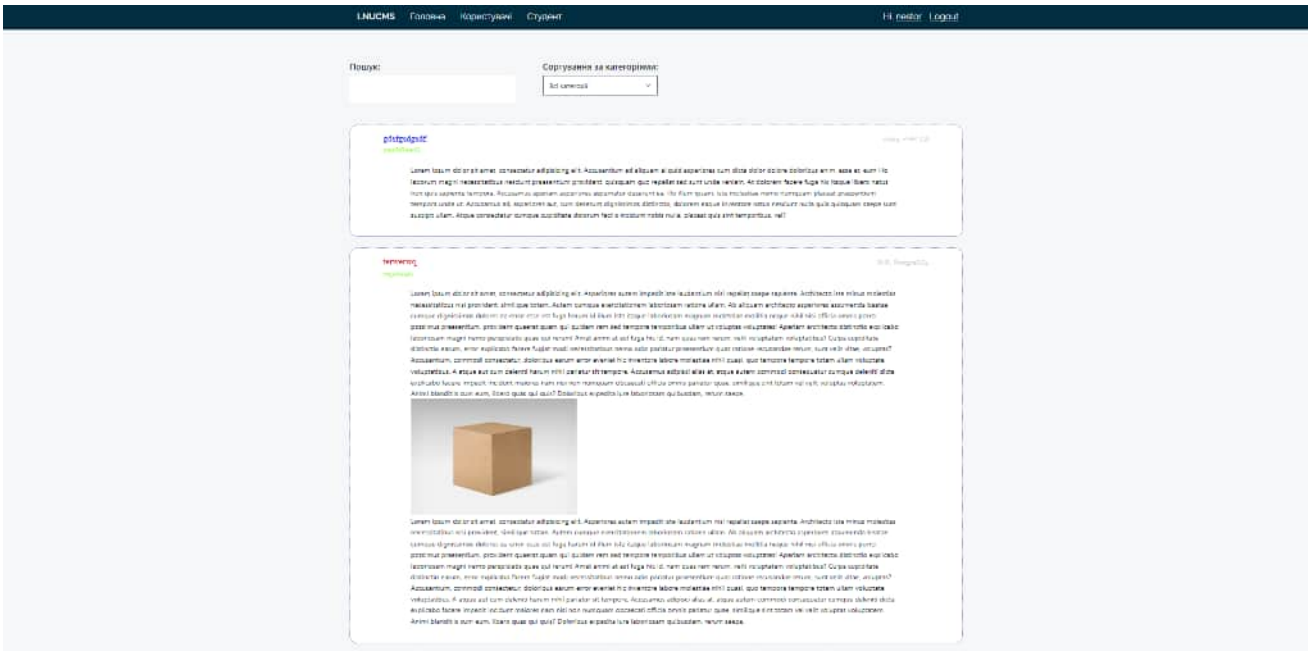
(Рисунок 1.6)

3.4 Головна сторінка веб-сайту

На цій сторінці користувач може переглядати всі доступні публікації на веб-сайті.

Кожна публікація представлена, як окремий елемент у списку і містить заголовок, підзаголовок, короткий опис, категорії. Користувач може прокручувати список публікацій, переглядаючи їх по одному або застосовуючи фільтри та сортування для відображення певних типів публікацій відповідно до категорій або скористатися пошуком за назвою.

При кліку на окрему публікацію, користувача переадресовує на сторінку цієї публікації, де він може прочитати повний зміст.

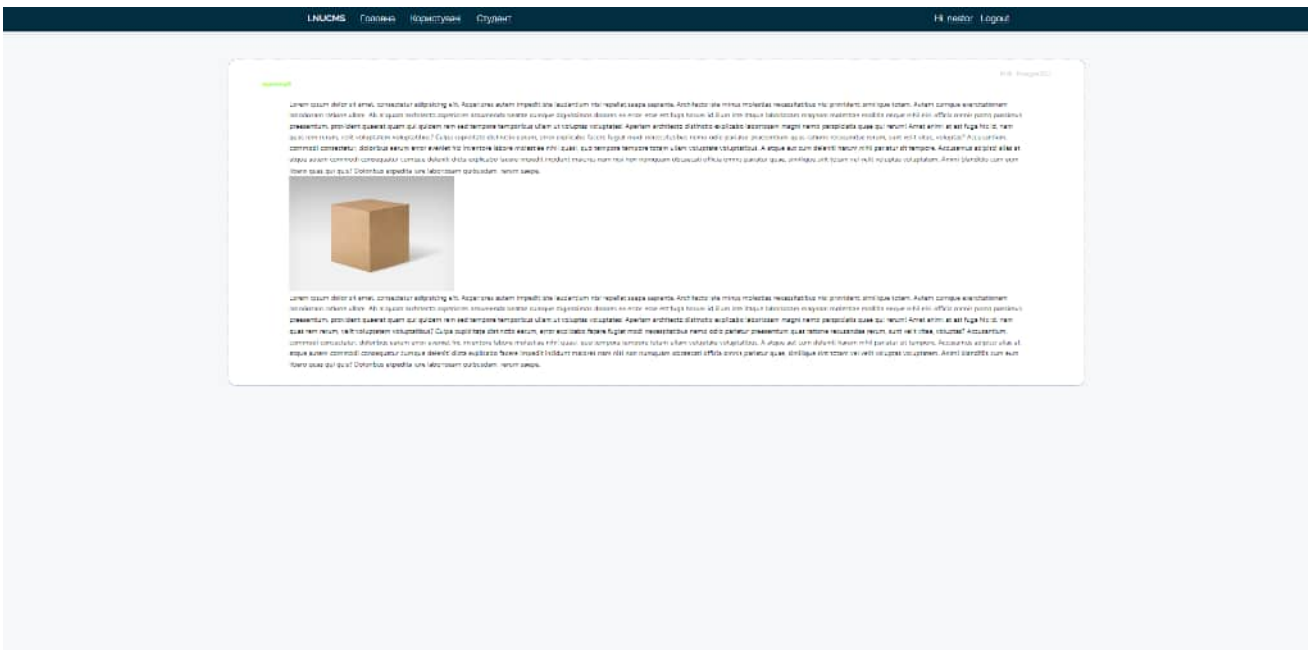


(Рисунок 1.7)

3.5 Перегляд окремої публікації

Функціонал перегляду окремої публікації у веб-додатку надає користувачам можливість переглядати повний зміст конкретної публікації.

При кліку на певну публікацію зі списку постів або за допомогою унікального посилання, користувач переходить на окрему сторінку цієї публікації. На цій сторінці користувач може бачити всю інформацію, пов'язану з даною публікацією. Сторінка публікації містить повний текст або контент публікації, який може включати текст, зображення, відео, аудіо чи інші медіаелементи.



(Рисунок 1.8)

3.8 Створення публікації

Користувач з правом доступу може використовувати спеціальну форму або інтерфейс створення публікації для введення необхідної інформації. Це включає такі поля, як заголовок, підзаголовок, текст, зображення, відео, аудіо чи інші медіаелементи у відповідному редакторі, категорія та можливість відображення коментарів.

Користувач може використовувати форматування тексту, додавати розмітку, стилізувати зміст публікації, додавати гіперпосилання, завантажувати фотографії або відео, надавати посилання на відео з медіа ресурсів та використовувати інші можливості редактора для створення бажаного вигляду та формату публікації.

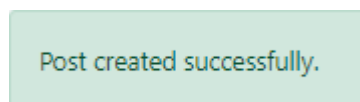
The screenshot shows a web application interface for creating a new publication. At the top, there is a navigation bar with the text 'LNUCHS' and several menu items: 'Головна', 'Відомості', 'Статті', 'Н. Бабич', and 'Лого'. Below the navigation bar, the form is divided into several sections:

- Нова публікація:** A text input field with the placeholder 'Введіть назву'.
- Короткий опис:** A text input field with the placeholder 'Введіть короткий опис'.
- Стиль оформлення:** A dropdown menu with the placeholder 'Вибір стилю'.
- Вибірть до якого меню віднести публікацію:** A dropdown menu with the placeholder 'Вибір меню'.
- Опис:** A rich text editor with a toolbar containing icons for bold, italic, underline, link, unlink, list, and other text formatting options.
- Теги публікації:** A list of checkboxes for tags: 'PFT', 'CF', 'NoName', 'PostgreSQL', and 'restor2'.

At the bottom of the form, there is a large green button labeled 'Опублікувати'.

(Рисунок 1.9)

При успішному створенні публікації, система повідомляє користувача про це сповіщенням:

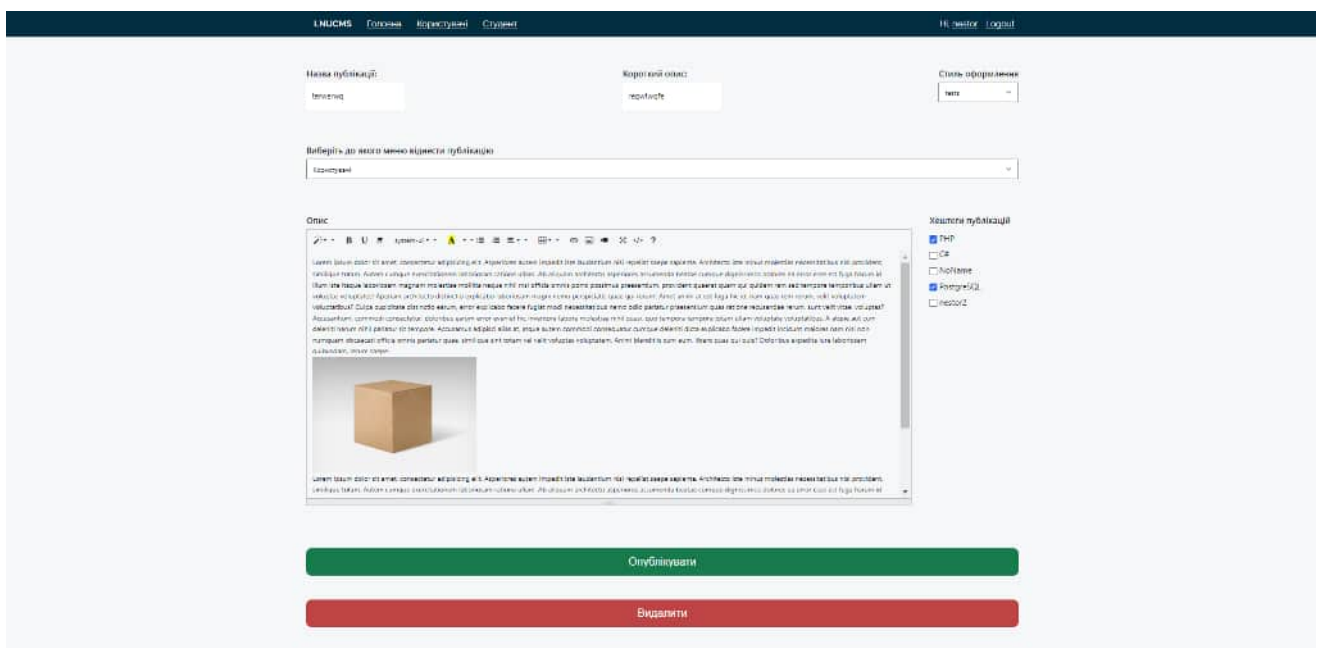


(Рисунок 1.10)

3.7 Редагування публікації

Функціонал редагування публікації в розробленому веб-додатку дозволяє користувачам вносити зміни до вже опублікованих матеріалів. Це дає можливість виправити помилки, доповнити вміст або оновити інформацію в публікації.

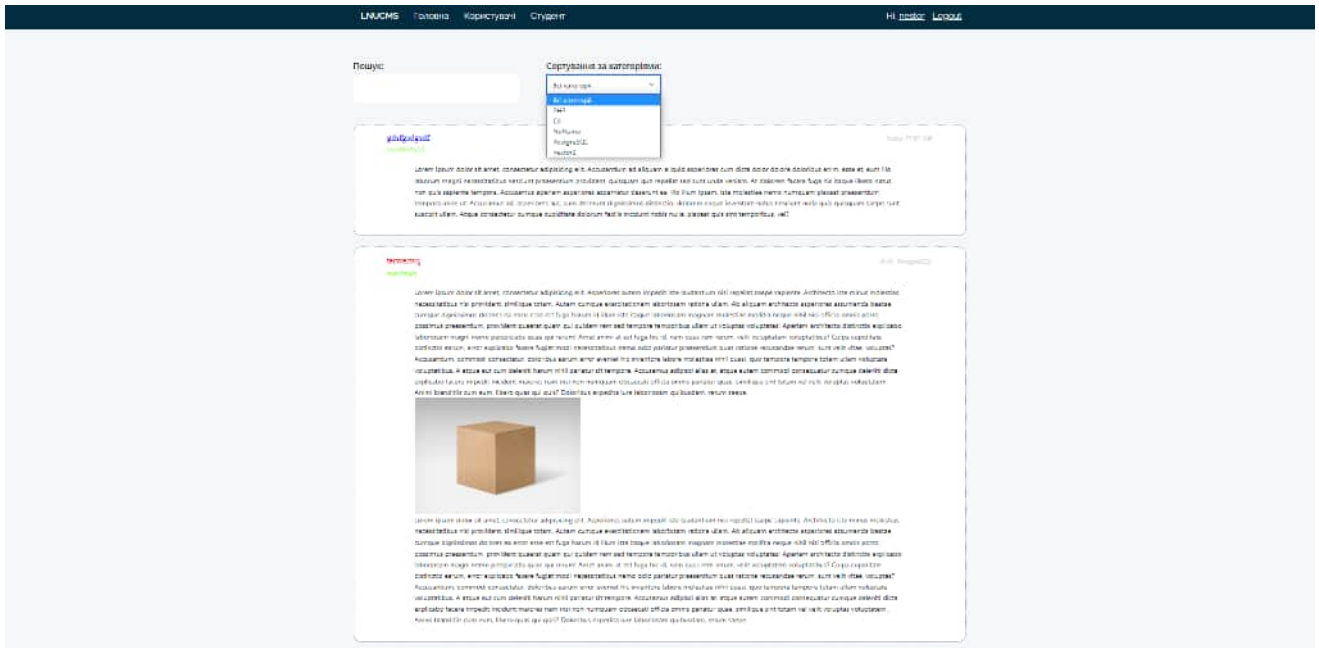
Для редагування публікації користувач повинен мати доступ до власних публікацій або мати відповідні права доступу. При редагуванні публікації, користувач може змінювати будь-який аспект публікації, такий як заголовок, підзаголовок, текст, зображення, відео, аудіо чи інші медіаелементи у відповідному редакторі, категорія та можливість відображення коментарів.



(Рисунок 1.11)

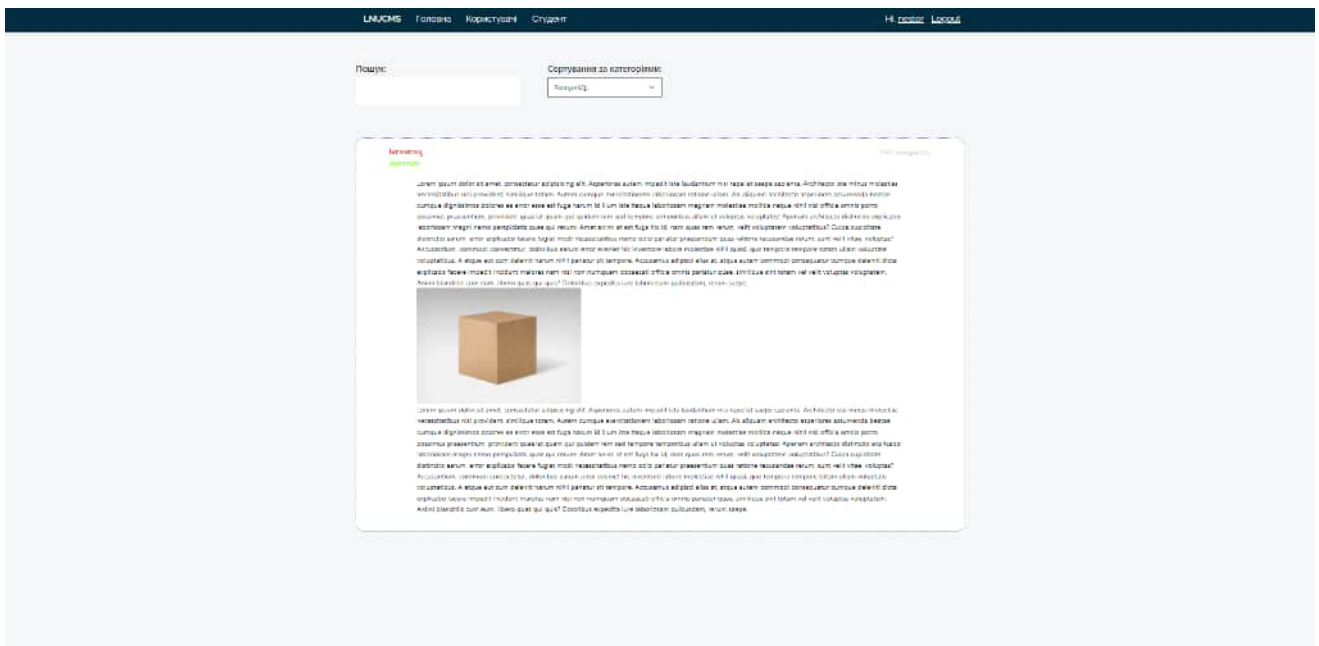
3.8 Сортування публікацій по категоріях

У розробленому веб-додатку кожна публікація має призначену категорію або декілька категорій. При відображенні списку публікацій на сторінці, користувач має можливість обрати бажану категорію з випадаючого списку.



(Рисунок 1.12)

Після вибору категорії, веб-додаток відображає лише ті публікації, які належать до обраної категорії.

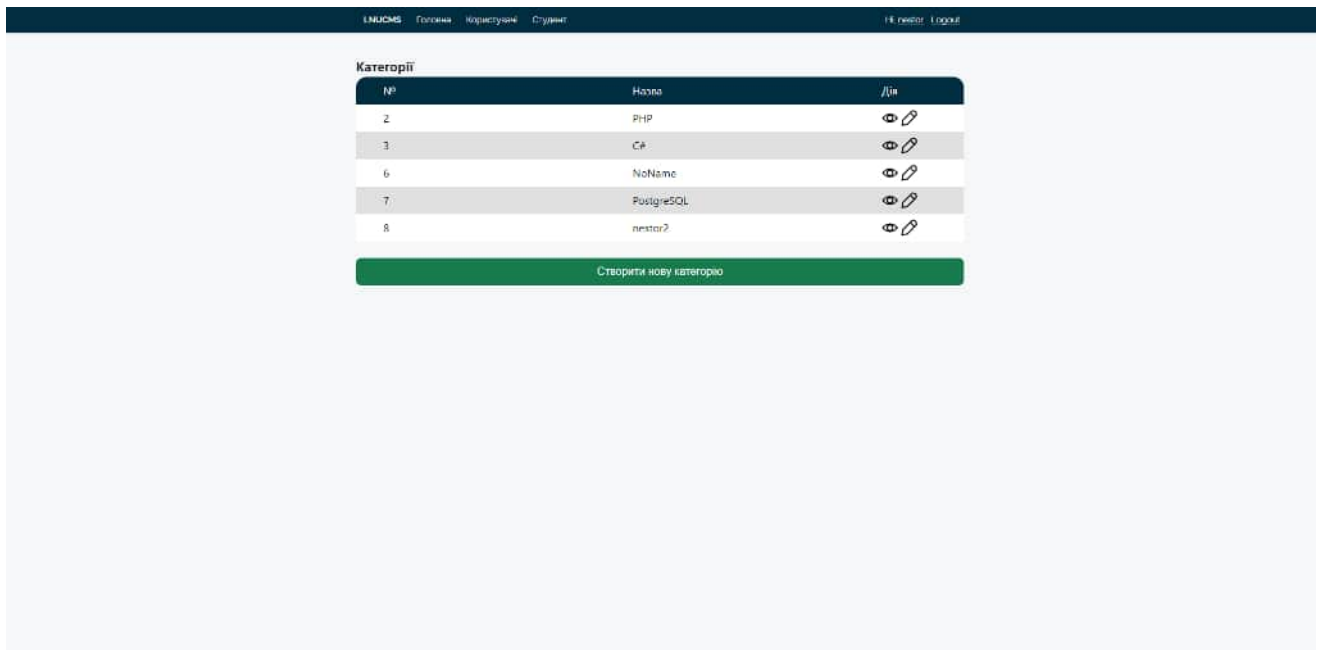


(Рисунок 1.12)

3.9 Відображення всіх категорій

Функціонал відображення всіх категорій дозволяє користувачам переглянути список усіх наявних категорій, що використовуються в системі. При натисканні на категорію, система перенаправляє на відповідну категорію для модифікації.

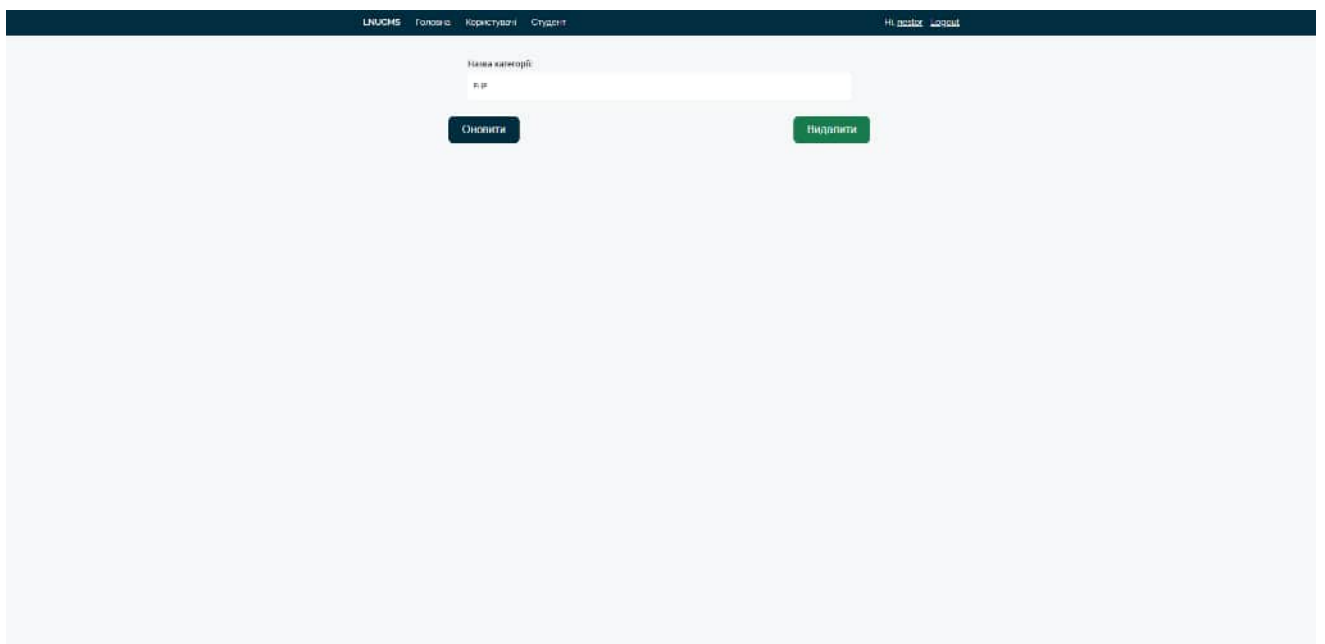
Також доступна кнопка для створення нової категорії - “Create Category”



(Рисунок 1.13)

3.10 Відображення певної категорії

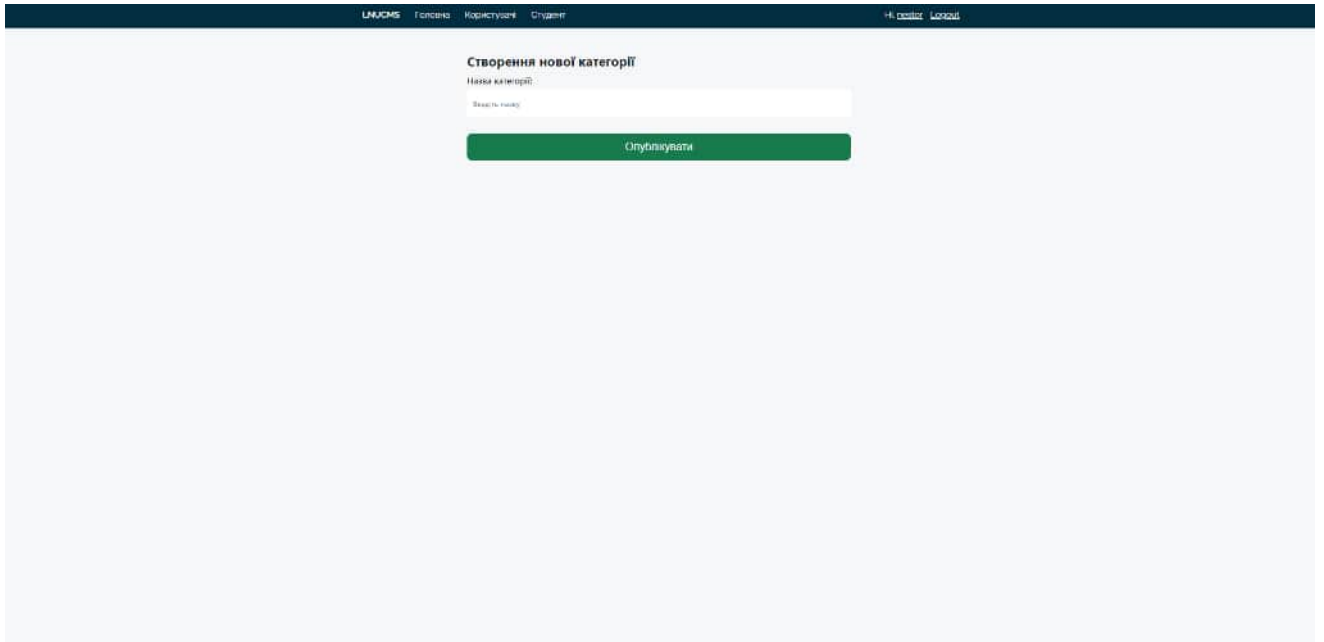
При кліку на іконку перегляду категорії, ми можемо переглянути інформацію про неї, а також можливість оновити інформацію або видалити категорію:



(Рисунок 1.14)

3.11 Створення та редагування нової категорії

Функціонал створення нової категорії для можливості подальшого сортування публікацій. Ці категорії відобразатимуться у створенні або редагуванні публікації, де користувач може обрати категорії, які відноситимуться до неї.

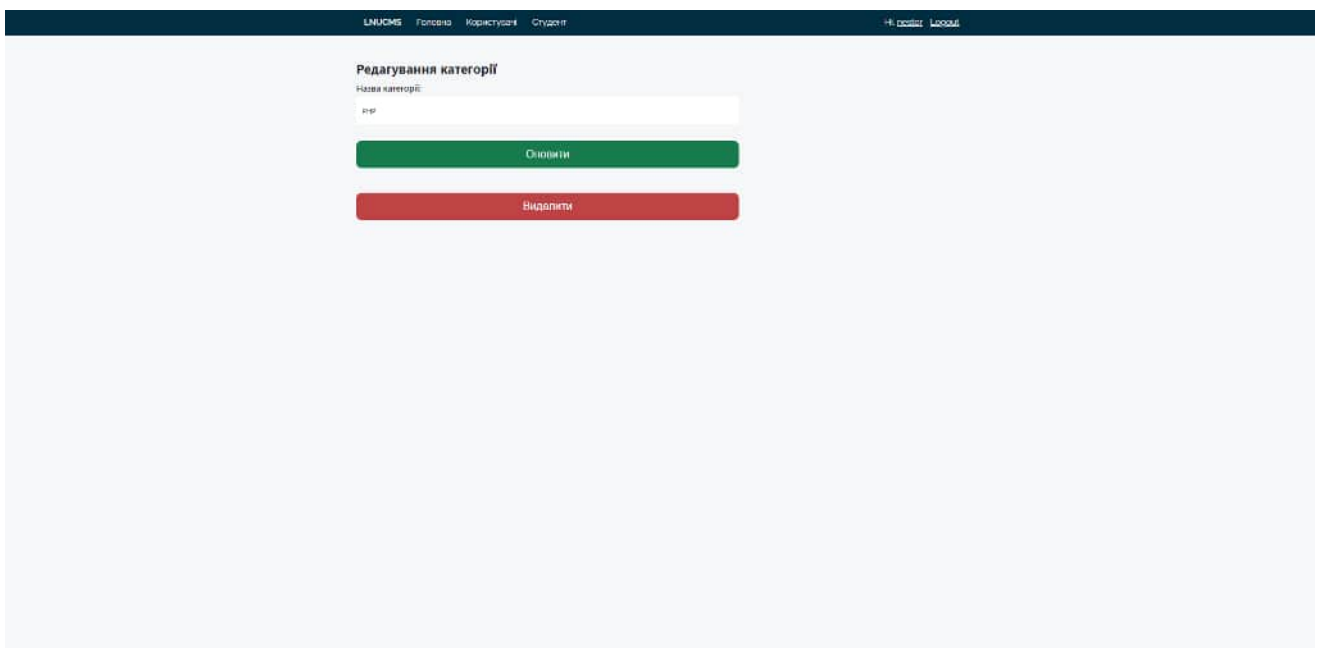


The screenshot shows a web interface for creating a new category. At the top, there is a dark blue navigation bar with the text 'LNUJCMS' and links for 'Головна', 'Користувач', and 'Студент'. On the right side of the bar are links for '+ реєстр' and 'Logout'. The main content area is white and contains the following elements:

- Створення нової категорії** (Create new category)
- Назва категорії: (Category name): A text input field.
- Введіть назву (Enter name): A label for the input field.
- Опублікувати (Publish): A green button.

(Рисунок 1.15)

Функціонал редагування категорії:



The screenshot shows a web interface for editing a category. At the top, there is a dark blue navigation bar with the text 'LNUJCMS' and links for 'Головна', 'Користувач', and 'Студент'. On the right side of the bar are links for '+ реєстр' and 'Logout'. The main content area is white and contains the following elements:

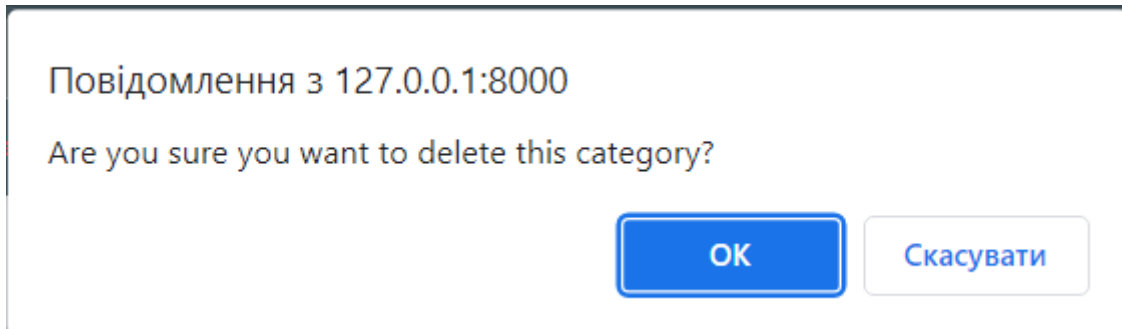
- Редагування категорії** (Edit category)
- Назва категорії: (Category name): A text input field.
- №: A text input field.
- Оновити (Update): A green button.
- Видалити (Delete): A red button.

(Рисунок 1.16)

3.12 Видалення категорії

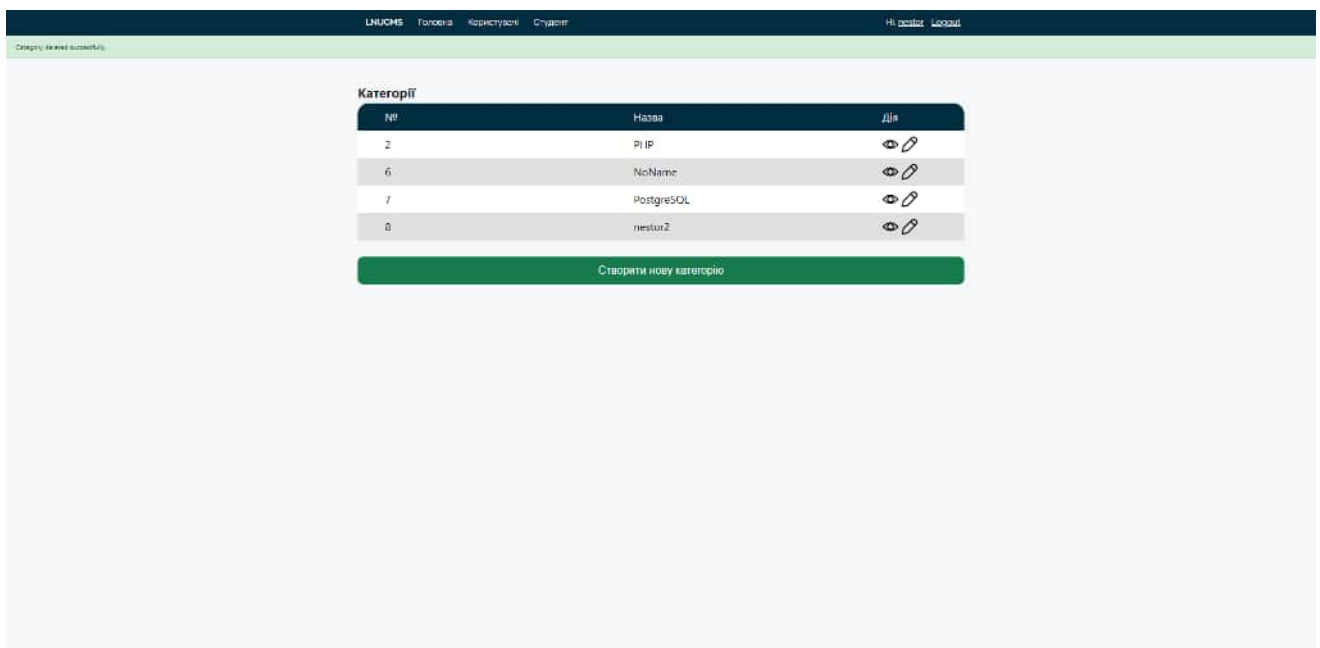
При натисканні на кнопку видалення категорії - з'являється віконце де потрібно підтвердити видалення категорії

(Рисунок 1.17)



(Рисунок 1.18)

Повідомлення про успішне видалення категорії:

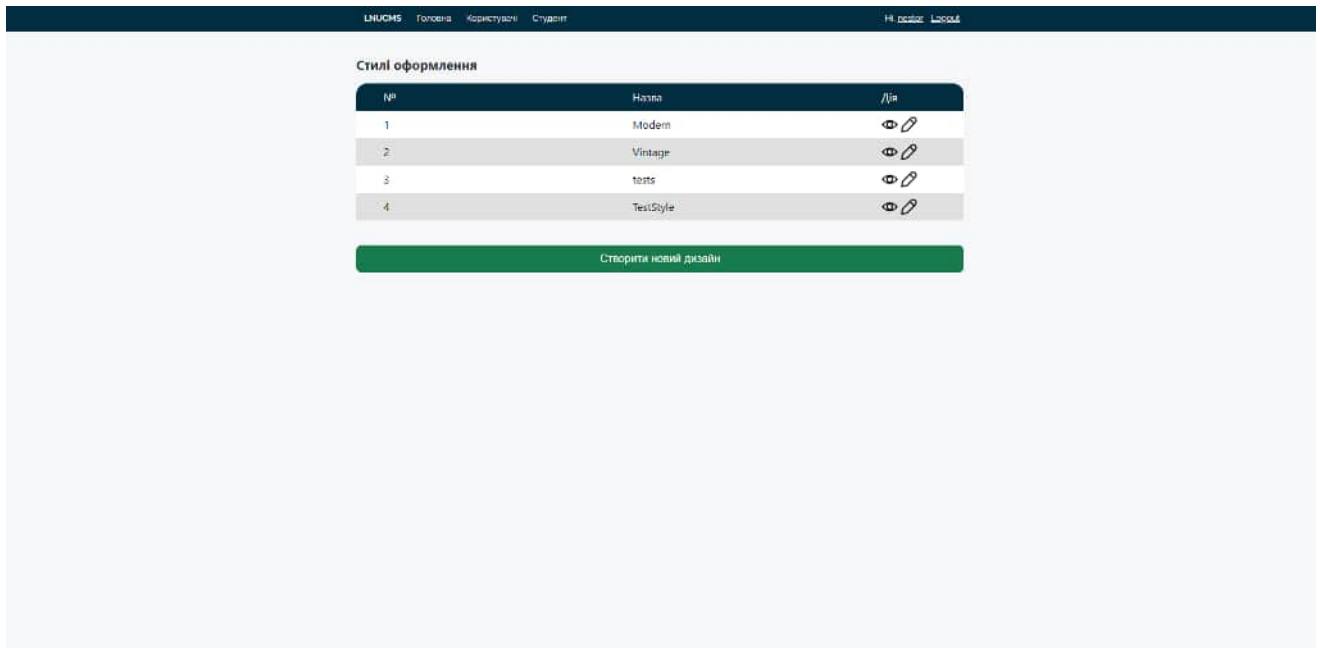


(Рисунок 1.19)

3.12 Перегляд всіх створених дизайнів

Функціонал "Перегляд всіх створених дизайнів" дозволяє користувачам переглянути список всіх дизайнів, які вони створили. Це забезпечує зручний спосіб організації та відстеження всіх створених

дизайнів в системі.



(Рисунок 1.20)

3.13 Створення нового дизайну

Створення нового дизайну надає користувачам можливість створити унікальні візуальні концепції для своїх публікацій. Цей функціонал включає ряд можливостей, що дозволяють користувачам налаштовувати різні аспекти дизайну з метою створення естетично привабливого та професійного вигляду.

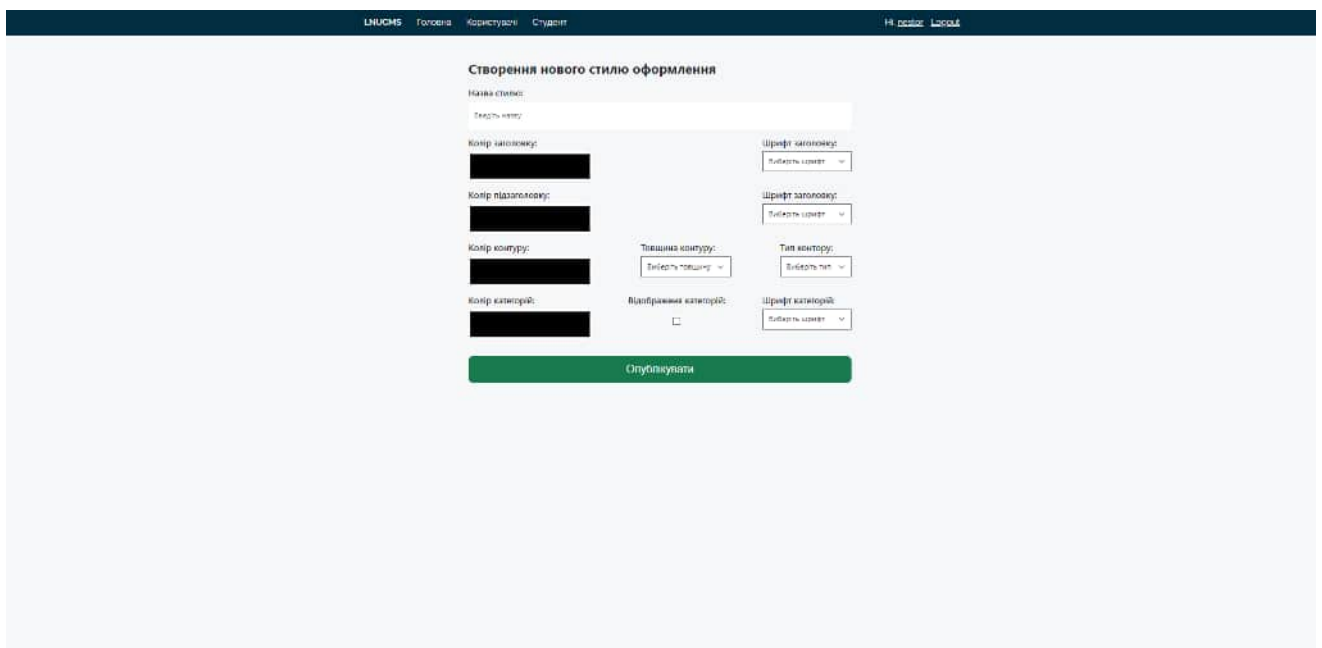
Починаючи зі створення нового дизайну, користувачі можуть задати назву, яка відповідає змісту або тематиці публікацій. Це дозволяє створити ідентифікаційний елемент, який відобразить суть та характер дизайну. Одним із ключових аспектів, які можна редагувати, є вибір кольорів текстів та шрифтів.

Користувачі мають можливість вибрати палітру кольорів, що найкраще відповідає їхньому стилю та бренду, а також встановити відповідний шрифт для створення гармонійного та збалансованого вигляду тексту. Для задання фону публікації, користувачі можуть вибрати один з варіантів, таких як однотонний фоновий колір або завантажити власне зображення. Це дозволяє створити візуальну атмосферу, яка підкреслює

тему або настрій публікацій.

Додатковою можливістю є редагування контуру елементів дизайну. Користувачі можуть встановити товщину, тип лінії та колір контуру для надання визначеності та акценту окремим елементам. Крім цього, користувачі можуть визначити розмір, тип та колір контуру, що дозволяє змінювати вигляд та стиль контуру елементів. Створення нового дизайну також дозволяє вибрати, чи відображатимуться категорії публікацій у цьому дизайні.

Користувачі можуть включати або виключати категорії залежно від своїх потреб та уподобань, що допомагає забезпечити зручну організацію та навігацію по вмісту. Загалом, створення нового дизайну надає користувачам велику свободу та контроль над виглядом своїх публікацій. Цей функціонал дозволяє створювати привабливі, стильні та персоналізовані дизайни, які допоможуть привернути увагу аудиторії та підкреслити унікальність контенту.



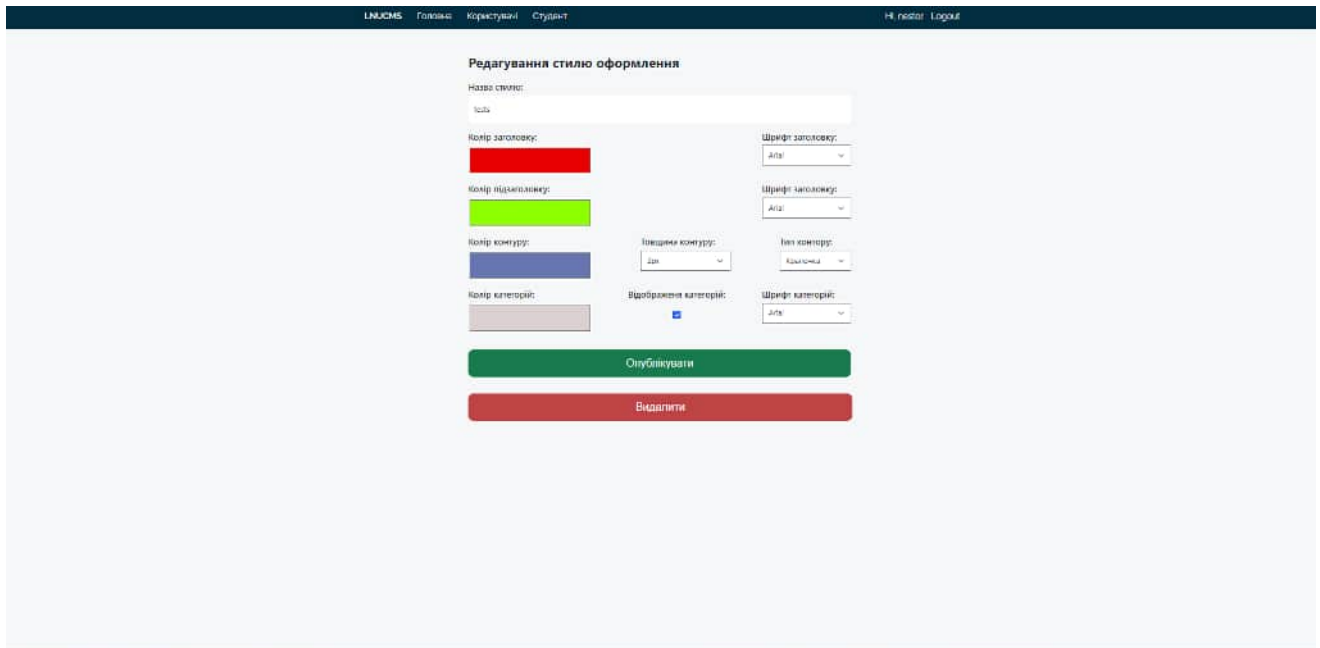
The screenshot shows a web interface for creating a new design. The title is "Створення нового стилю оформлення". The form contains the following fields and options:

- Назва стилю: (text input)
- Тема меню: (text input)
- Колір заголовку: (color picker)
- Колір підзаголовку: (color picker)
- Колір контуру: (color picker)
- Колір категорій: (color picker)
- Шрифт заголовку: (font dropdown)
- Шрифт підзаголовку: (font dropdown)
- Товщина контуру: (line thickness dropdown)
- Тип контуру: (line style dropdown)
- Відображення категорій: (checkbox)
- Шрифт категорій: (font dropdown)

A green "Опублікувати" button is located at the bottom of the form.

(Рисунок 1.21)

3.14 Редагування існуючого дизайну



(Рисунок 1.22)

Функціонал "Редагування існуючого дизайну" надає користувачам широкий спектр можливостей для налаштування і вдосконалення вже створених дизайнів. Завдяки цьому функціоналу, користувачі можуть вносити зміни до різних аспектів дизайну, що дозволяє їм персоналізувати вигляд та відповідність дизайну їхнім потребам і бажанням.

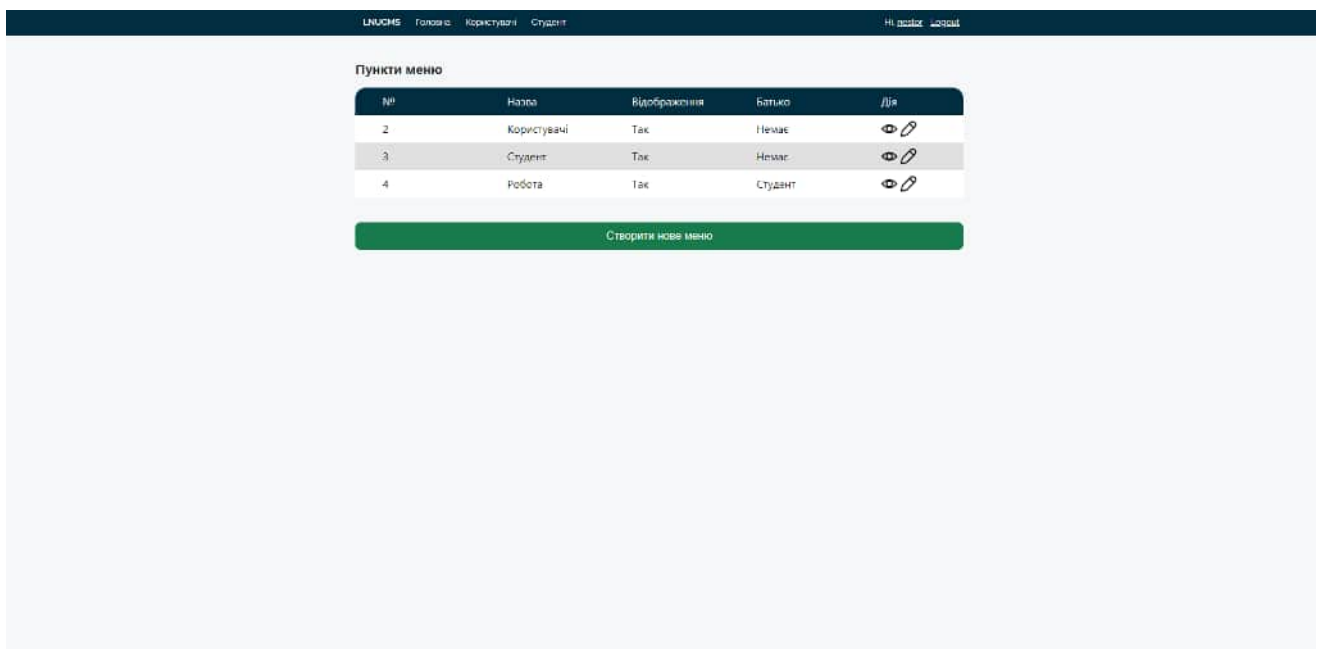
Одним із ключових аспектів, які можна редагувати, є задання назви для дизайну. Це дозволяє користувачам вказати зрозумілу і відповідну назву, що відображатиме суть дизайну або його контекст. Крім того, користувачі можуть налаштовувати кольори текстів та шрифти, що дозволяє їм вибрати відповідні кольорові схеми та типографіку, що найкраще підходять до їхніх потреб і стилю. Вони можуть вибирати з широкого спектру кольорів і шрифтів, а також встановлювати розмір тексту для досягнення оптимальної читабельності та впливу. Додатковою можливістю є зміна фону публікації.

Користувачі можуть вибрати певний фоновий колір або навіть завантажити своє власне зображення, що дозволяє створити бажаний настрій та візуальний ефект для дизайну. Функціонал "Редагування існуючого дизайну" також надає можливість редагувати контур елементів дизайну.

Користувачі можуть встановлювати параметри контуру, такі як товщина, тип лінії та колір, щоб виділити окремі елементи або надати їм візуальну структуру. Окрім цього, користувачі мають можливість вибирати, чи відображатимуться категорії в цьому дизайні публікацій. Вони можуть включати або виключати відображення категорій залежно від своїх потреб і вимог.

3.15 Перегляд всіх створених пунктів меню

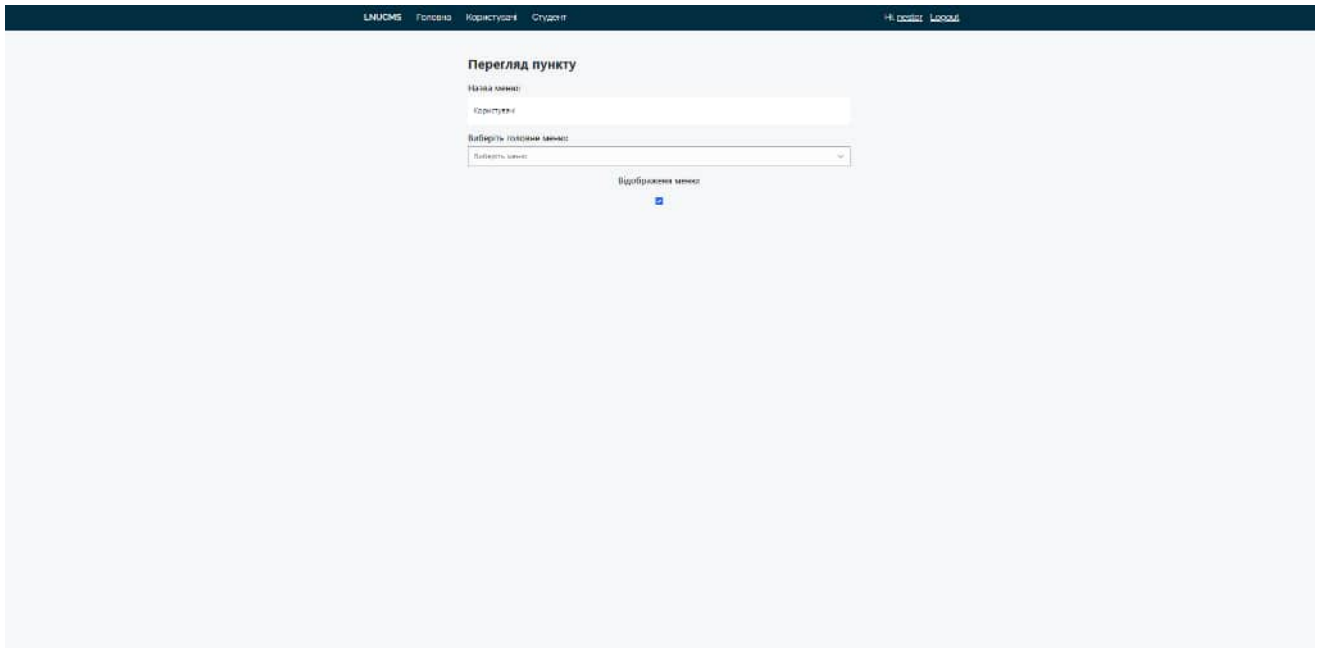
Цей функціонал дозволяє переглянути всі створені пункти меню, інформацію про їхню назву, чи відображається пункт в навігації сайту в “шапці” сайту та чи він є підпунктом до головного пункту меню.



(Рисунок 1.23)

3.16 Перегляд конкретного пункту

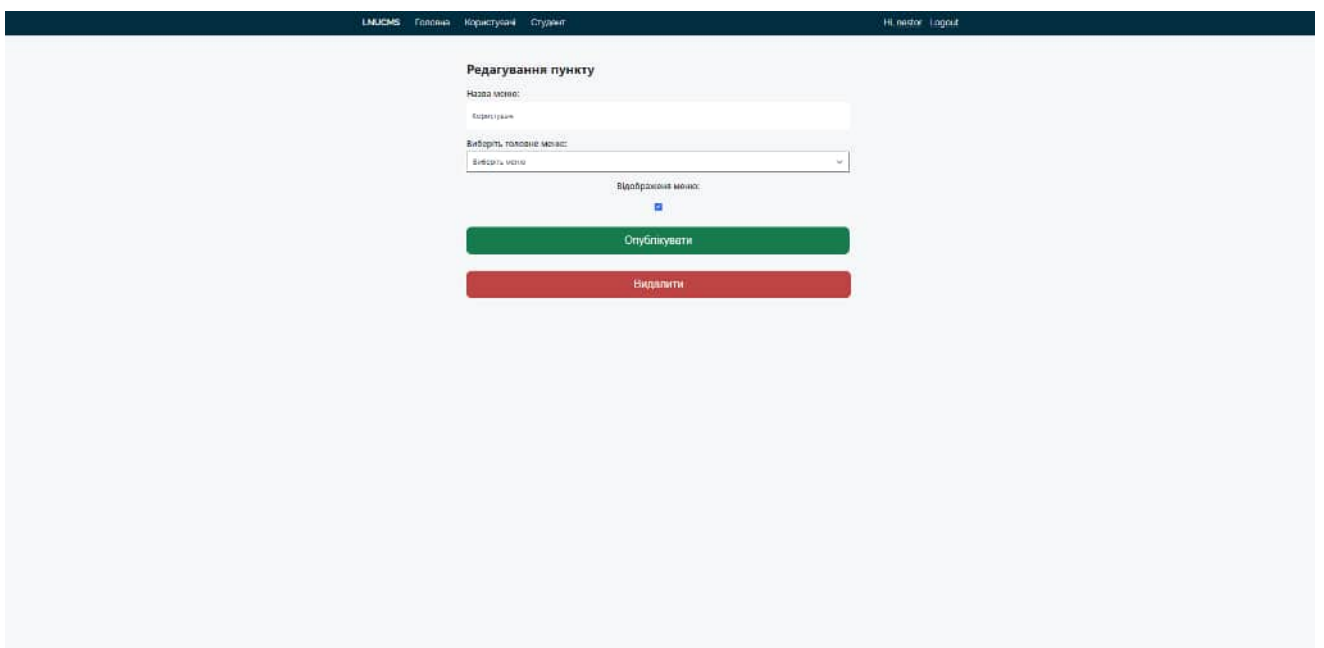
При кліку на іконку перегляду пункту меню, ми можемо переглянути інформацію про цей пункт: назву, головне меню та чи відображається це меню в “шапці сайту”:



(Рисунок 1.24)

3.17 Редагування конкретного пункту

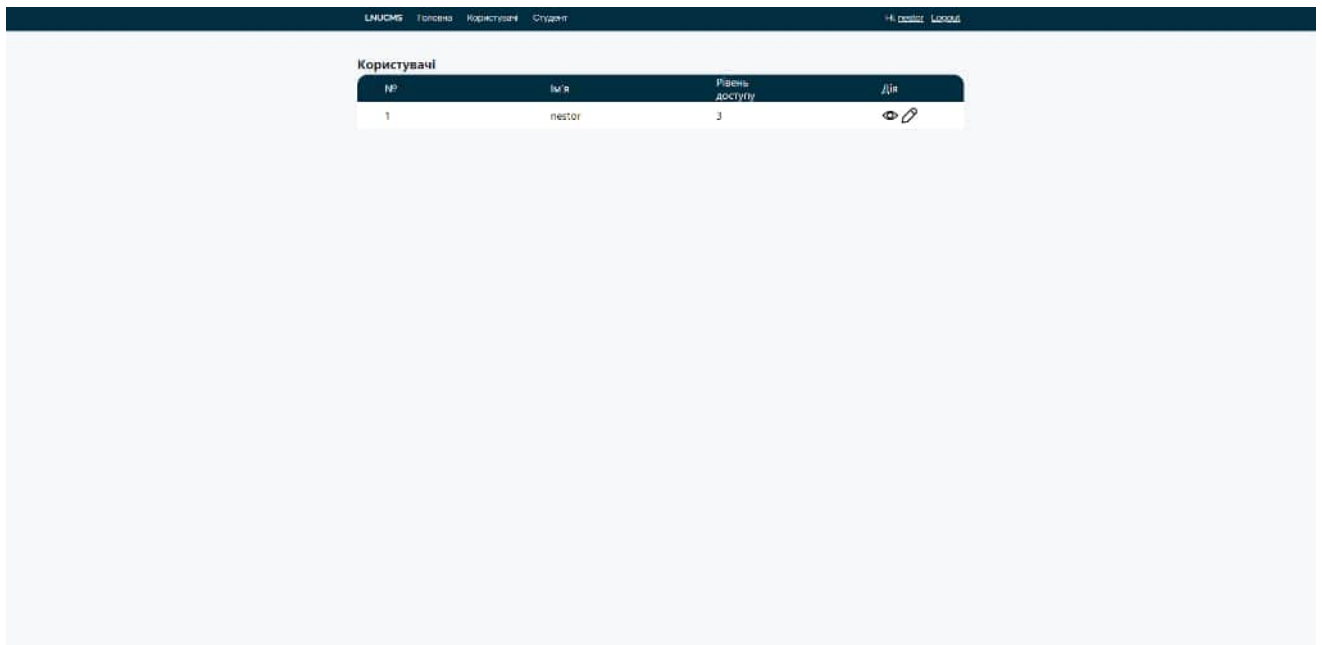
При кліку на іконку редагування пункту меню, ми можемо змінити доступну нам інформацію про цей пункт та натиснути опублікувати, щоб застосувати зміни. Також ми можемо видалити цей пункт меню.



(Рисунок 1.25)

3.18 Перегляд всіх користувачів системи

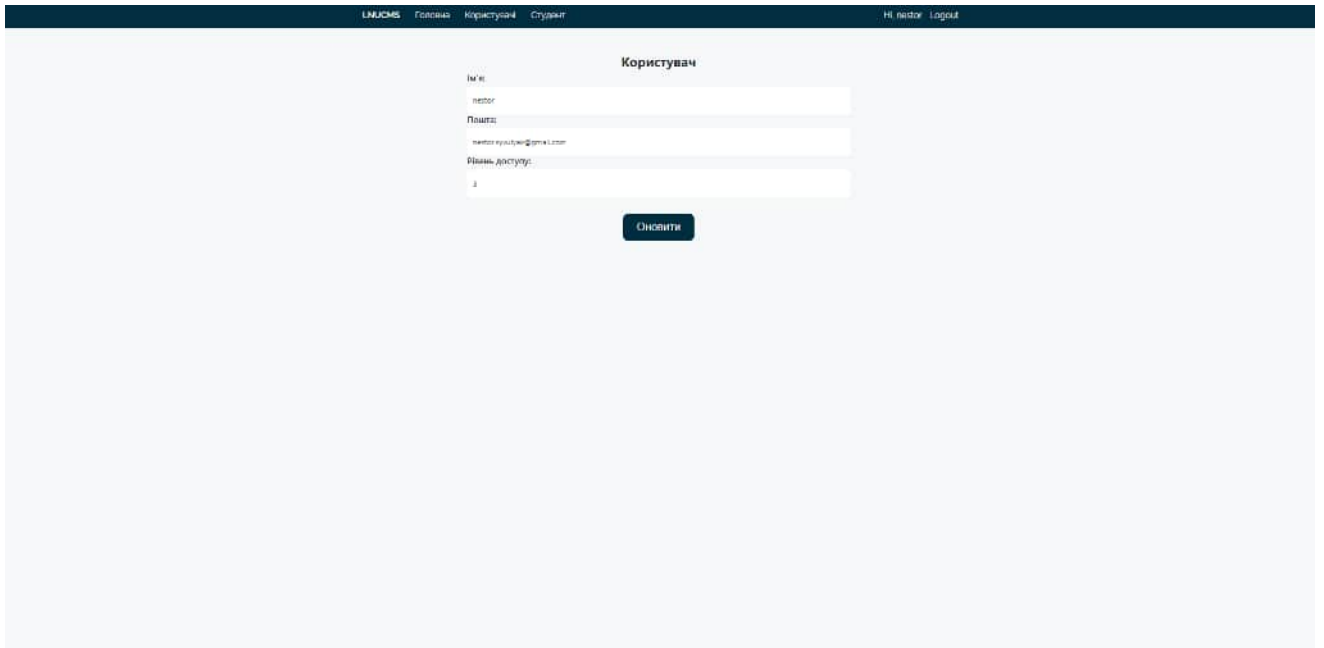
Функціонал "Перегляд всіх користувачів системи" надає адміністраторам можливість переглянути всіх користувачів у системі, переглянути кожного окремо та редагувати інформацію про користувача при кліку на відповідні ікони в меню.



(Рисунок 1.26)

3.19 Перегляд конкретного користувача

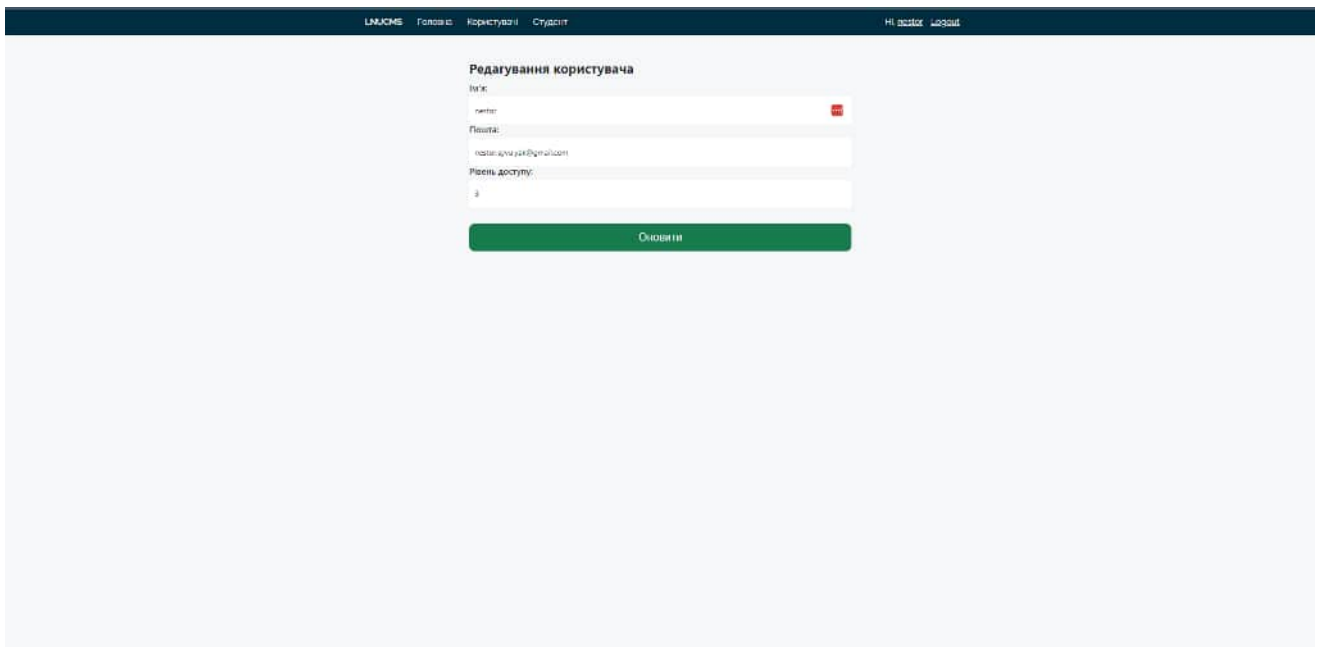
При кліку на відповідну іконку перегляду користувача, ми можемо побачити основну інформацію про нього: електронна адреса, ім'я та рівень доступу. Адміністратори мають можливість змінити користувачу рівень доступу відповідно до потреб.



(Рисунок 1.27)

3.20 Редагування користувача

При кліку на відповідну іконку редагування користувача, ми можемо змінити рівень доступу та зберегти зміни у нашій базі даних.

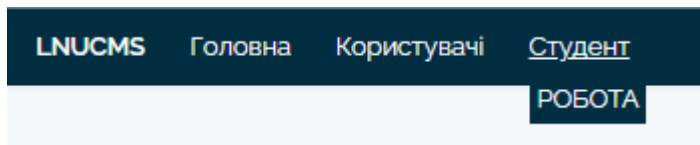


(Рисунок 1.28)

3.21 Відображення меню в шапці сайту

Навігаційне меню відображається відповідно до заданих параметрів у системі: редагування пунктів меню. Адміністратор може добавляти, змінювати та видаляти пункти меню відповідно до своїх потреб. Для

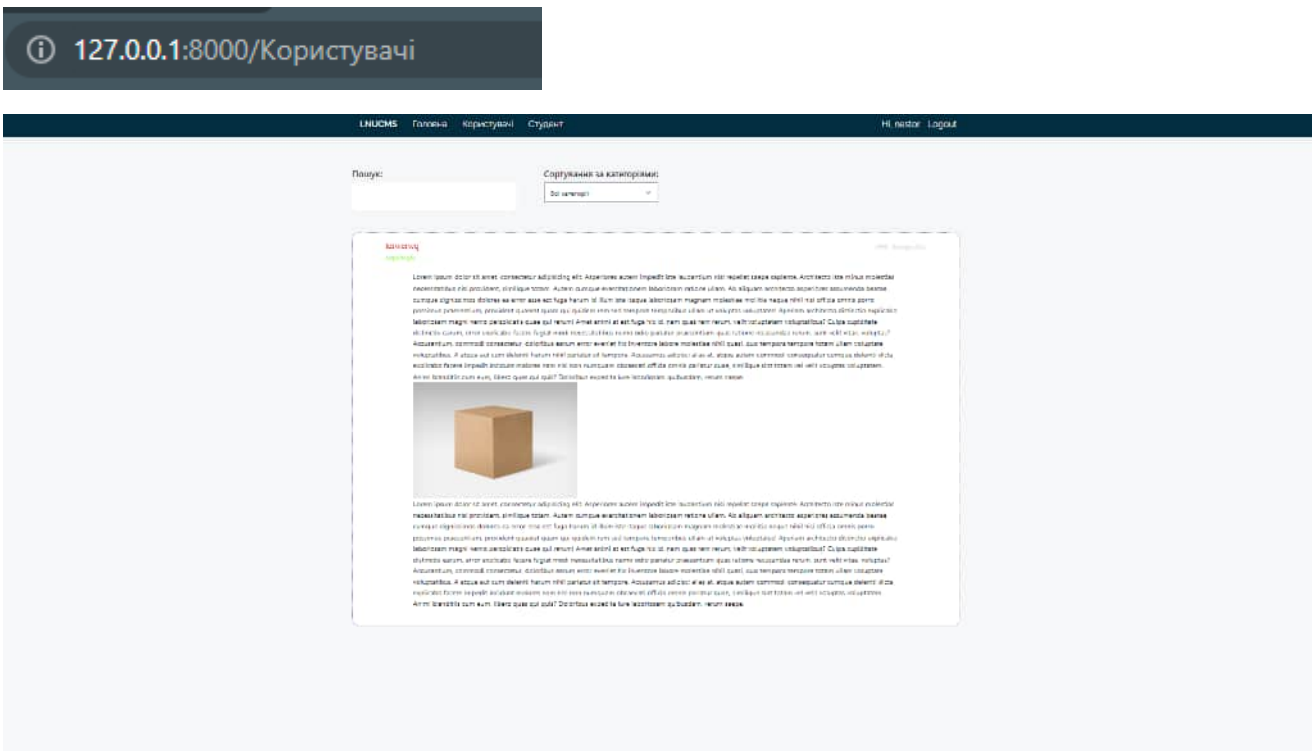
кожного пункту меню створюється окрема сторінка з відповідним шаблоном відображень публікацій.



(Рисунок 1.29)

3.22 Створена сторінка з пункту меню

Адміністратор створив сторінку - “Користувачі” та вказав відобразити у шапці веб-сайту. При кліку на цей пункт меню нас переспрямовує на цю сторінку та відображає всі публікації які зв’язані з нею. Таким чином, є можливість створювати свої сторінки з відповідними публікаціями та відносити їх до цих пунктів меню для зручності та для створення більш структурованості веб-сайту.



(Рисунок 1.30)

ВИСНОВОК

Розробка системи управління веб-публікаціями є невід'ємною частиною сучасного інтернет-середовища. Застосування відповідних технологій та інструментів дозволяє ефективно керувати вмістом, забезпечувати його структурованість, зручну навігацію та персоналізацію для користувачів.

У розробці системи управління веб-публікаціями був використаний широкий стек технологій, який включав PHP, Laravel, HTML/CSS, JavaScript, Ajax та Figma. PHP та Laravel забезпечують потужність та функціональність для розробки веб-додатків, HTML/CSS дозволяють створювати елементи інтерфейсу та візуальний дизайн, JavaScript та Ajax додають динамічність та взаємодію з користувачем, а Figma використовується для дизайну та прототипування.

Результатом розробки системи є потужний інструмент, який дозволяє користувачам зручно керувати веб-публікаціями. Функціонал системи включає створення, редагування та видалення публікацій, керування категоріями, сортування та фільтрацію публікацій, а також налаштування вигляду та дизайну, створювати та редагувати пункти меню, переглянути всіх користувачів та надавати доступ за потреби.

Використання такого стеку технологій дозволяє забезпечити ефективну та надійну роботу системи, швидкість завантаження сторінок, зручну інтерактивність та гнучкість у налаштуванні. Користувачі системи можуть ефективно керувати своїми веб-публікаціями, вибираючи зручний інтерфейс та налаштування за допомогою відповідних інструментів.

Розробка системи управління веб-публікаціями є важливим етапом для підтримки активності та розвитку веб-середовища. Вона дозволяє зосередитися на змісті та представленні інформації, спрощує процеси керування та публікації, а також надає користувачам багатофункціональні інструменти для ефективної роботи з веб-платформою.

Список використаних джерел

1. Development web application [електронний ресурс] -
<https://www.fingent.com/blog/web-application-development-a-detailed-guide/>
2. .Офіційна документація Laravel - [електронний ресурс]
<https://laravel.com/docs/>
3. Code Complete: McConnell, Steve
4. Офіційна документація MySQL- [електронний ресурс]
<https://dev.mysql.com/doc/>
5. PHP Manual - [електронний ресурс]
<https://www.php.net/manual/en/>
6. Діаграма use-case [електронний ресурс]
<https://www.quality-assurance-group.com/use-case-diagrams/>
7. Composer - [електронний ресурс]
<https://getcomposer.org/doc/>
8. Head First Design Patterns книга, Е. Фрімен, Е.Робсон, Б. Бейтс, К. Сьєрра
9. JavaScript [Електронний ресурс]: 2023 MDN contributors:
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
10. SQL and PostgreSQL: The Complete Developer's Guide [електронний ресурс]
-
<https://www.udemy.com/course/sql-and-postgresql/>
11. Webpack.js [електронний ресурс]
<https://webpack.js.org/>
12. Database migration [електронний ресурс]
<https://www.prisma.io/dataguide/types/relational/what-are-database-migrations>
13. SASS/SCSS [електронний ресурс]
<https://sass-lang.com/>
14. Книга - “Чистий кодер” Роберт Мартін.