

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ХЕРСОНСЬКИЙ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

Факультет комп'ютерних наук, фізики та математики

Кафедра комп'ютерних наук та програмної інженерії

Створення програмного модуля сайту новин ХДУ

Кваліфікаційна робота (проект)

на здобуття ступеня вищої освіти «бакалавр»

Виконав: здобувач 4 курсу 441 групи

Спеціальність: 121 Інженерія програмного
забезпечення

Освітньо-професійна програма: Інженерія
програмного забезпечення

Болюк Георгій Юрійович

Керівниця: Валько Наталія Валеріївна,
професор кафедри комп'ютерних наук та
програмної інженерії

Рецензент: Фомічов М.М., Quality assurance
engineer, DataArt

Івано-Франківськ – 2024

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. Теоретичні огляди сайтів новин та вивчення їх	5
1.1. Огляд новинних сайтів.....	5
1.2. Вимоги до новинного сайту.....	7
1.3. Огляд відомих CMS.....	8
1.4. Основні безпекові заходи для підтримки сайту.....	11
Висновки до першого розділу.....	15
РОЗДІЛ 2. Планування сайту та його структури	16
2.1 Основні функції сайту.....	16
2.2 Структура сайту.....	18
2.3 Різниця доступів для ролей.....	19
2.4 Організація вхідних/вихідних даних.....	21
Висновки до другого розділу.....	24
РОЗДІЛ 3. Розробка додаткового функціоналу сайту	26
3.1. Реалізація функцій сайту.....	26
3.2. Використання contrib модулів та їх призначення.....	29
3.3. Розробка власних модулів.....	30
3.4. Забезпечення безпеки та швидкодії.....	35
Висновки до третього розділу.....	36
ВИСНОВКИ	38
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	39

ВСТУП

Основна потреба сайту «Новинний сайт університету» полягає в вирішенні проблем з доступом та розповсюдженням локальних, та не тільки, новин університету, та нашого міста, а саме: доступність — будь-хто, будь-де може дізнатися останні новини або знайти ті які його цікавлять, тематика — сайт буде мати різні категорії новин, детальний опис новин — на сторінці будуть цитати, є можливість інтегрувати відео, фото, звязок з власником сайту — є можливість надсилати онлайн листи на пошту не виходячи з сайту новин. Завдяки цьому сайту ми вирішимо ці проблеми та зможемо розвивати інформаційний простір Херсонського державного університету та нашого міста. Дивлячись на сьогоднішню тенденцію оцифровувати все, що нас оточує ми також приєднуємось. Очевидно, що все більш глибоке проникнення інформаційних технологій в усі аспекти сучасного суспільства, темпи якого в останні десятиліття значно прискорилися, привели до необхідності перегляду парадигми ведення бізнесу – трансформації підприємства в технологічну, digitalорієнтовану систему. Впровадження новітнього обладнання, програм, платформ й ІТ-інструментів формує фундамент конкурентних переваг підприємства за рахунок оптимізації бізнес-процесів, а цифрові технології дозволяють утримувати компаніям конкурентні позиції на ринку[1]. Найбільш інтенсивно оцифрування проникає в будні нашого життя які тісно пов'язані з користувачем: банки, ресторани, магазини і т.д. Завдяки оцифривізації бізнес стає більш простим, бо легше зібрати статистику переглядів та проводити аналітику даних.

Об'єкт роботи: інформаційні ресурси.

Предмет роботи: новинний сайт ХДУ.

Мета роботи: створення ресурсу для публікування та відслідковування новин ХДУ.

Для досягнення мети були поставлені наступні **завдання:**

1. Ознайомитись з новинними сайтами та визначити структуру

побудови сервісу.

2. Розробити і обґрунтувати технічне завдання.
3. Спроекувати та створити новинний сайт.

В першому розділі ми зробили огляд інших новинних ресурсів. Визначено як ми будемо розробляти цей ресурс. Розглянемо основні переваги обраних сервісів та мов програмування для розробки сайту. Визначимо ключові принципи та завдання розробки бекенду сайту. Визначимо яким шляхом ми будемо вирішувати поставленні завдання.

В другому розділі зробимо висновок з аналізу інших новинних сервісів і визначимось зі стилістикою та функціоналом сайту. Розробимо структуру сайту, додаткові додаткові модулі.

Практичне значення одержаних результатів: завдяки реалізації ресурсу ми зможемо розповсюджувати новини про наш університет та місто. Зручний доступ до інформації — студенти, викладачі та інші особи які зацікавлені в активності Херсонського державного університету матимуть зручний доступ до актуальних новин. Новинний сайт дозволить швидко публікувати та оновлювати інформацію про події, акції та інші важливі події в університеті. Виділяти основні важливі новини.

РОЗДІЛ 1. Теоретичні огляди сайтів новин та вивчення їх

1.1 Огляд новинних сайтів.

Для дослідження новинних сайтів університетів ми обрали такі ресурси: Волинський національний університет ім. Лесі Українки, Луцький національний технічний університет, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Львівський національний університет та сайт Херсонського державного університету. Див. Таблиця 1.1.

Під час дослідження новинних сайтів було виявлено, що вони мають кілька ключових особливостей, що є загальними для багатьох з них. По-перше, вимоги аудиторії цих сайтів ставлять особливий акцент на швидкість розгортання та редагування контенту. Це пояснюється тим, що новини мають короткий термін актуальності, тому оперативне оновлення та публікація стають ключовими для забезпечення високої якості сервісу. Крім того, зручність навігації та взаємодії з сайтом також є важливими аспектами для користувачів. Вони очікують легкості у пошуку та перегляді новин, швидкого доступу до необхідної інформації, а також можливості взаємодії з сайтом через коментарі або реагування на матеріали. Належне представлення інформації є ще одним важливим аспектом. Це означає, що новинний сайт повинен відображати інформацію чітко та лаконічно, з використанням зручного для читача формату та структури. Крім того, важливо мати можливість швидко та ефективно звертатися до архівних матеріалів для дотримання принципу доступності інформації на сайті.

Порівняння функціоналу інших новинних сайтів університетів

	ВНУ[2]	ЛНТУ[3]	НТУ[4]	КПШ[5]	ХДУ[6]
Категорії новин	+	-	-	+	-
Головна сторінка з актуальними новинами	+	+	+	+	+
Опитування	+	-	-	-	-
Реєстрація	-	-	-	-	-
Пошук	+	+	-	+	-
Сторінка новини	+	+	+	+	+
Посилання на соціальні мережі	+	+	+	+	+
Зворотній зв'язок	+	+	-	-	-
Поширення новини	+	+	+	+	+
Останні новини на	+	+	+	+	-

сторінці новини					
Перехід на наступну новину	-	-	-	+	-
Вставлення відео та фото на сторінку новини	+	+	+	+	+
Коментарі	-	-	-	-	-

Отже, новинні сайти повинні враховувати ці ключові особливості, щоб забезпечити задоволення потреб аудиторії та зберегти конкурентоспроможність на ринку інформаційних ресурсів.

1.2 Вимоги до новинного сайту

Опираючись на пункт 1.1 та на визначений функціонал новинних сайтів університетів України ми визначили, що сайт має мати такі можливості:

- Головна сторінка з останніми новинами.
- Створення новин.
- Різні категорії новин та сторінки для них.
- Пошук на сайті.
- Вставка відео і фото на сторінку новини.
- Різні ролі.
- Різні теми новин.
- Перехід на власні соціальні мережі.
- Можливість поширити новину в соціальних мережах.
- Збірка основних новин.

- Зворотній зв'язок.
- Перехід на наступну новину зі сторінки новини.
- Реєстрація.
- Вхід.
- Форма опитування.
- Перехід на інші ресурси університету.

1.3 Системи розробки CMS

В минулому пункті ми зазначили вимоги до новинного сайту також ми зазначили, що більшість університетів використовує конструктори сайтів для швидшої розробки, легкого користування для користувача та редактора новин, підтримки сайту, додавання нового функціоналу та в основному своєму — це гнучкість. Багато функціоналу можна не писати власноруч, а завдяки спільноті є можливість використовувати те, чого не вистачає в самому конструкторі сайтів. Ми зазначили, що досить багато сайтів були написані на WordPress, Drupal, Joomla та інші менш відомі конструктори сайтів.

Що спільного між цими конструкторами сайтів[7]:

- Вони безкоштовні та мають відкритий початковий код.
- Усі три написані на PHP.
- Усі підтримують MySQL, як систему управління базами даних. Але WordPress підтримує лише MySQL в той час, як Joomla та Drupal підтримують ще інші системи управління базами даних.
- Усі три використовують теми, шаблони та модулі для швидкого оформлення сайтів, а також плагіни та розширення.

Їх недоліки:

1. WordPress:

- Деякі плагіни можуть не працювати під час оновлення платформи.
- Відносно слабкий код. Проте з кожним роком ситуація стає кращою.

- Відсутність вбудованої гнучкості для більш важливих та складних сайтів.
- Ядро WP вимагає частого оновлення, яке може призвести до поломки існуючих плагінів.

2. Joomla:

- Вбудована продуктивність дуже повільна/незграбна.
- Слабке логування.
- Більшість доступних корисних плагінів платні.
- На відміну від інших CMS, вона не орієнтована на розробників.
- Обмежений ринок розширень у порівнянні з Drupal та WordPress.
- Пропонує дуже мало параметрів функцій і можливостей, що настраюються.
- Плагіни не завжди добре працюють разом, тому для виправлення ситуації знадобляться знання в області кодування.

3. Drupal:

- Платформа досить складна в освоєнні порівняно з WordPress чи Joomla.
- Потрібно хостинг та доменне ім'я, оскільки він не пропонує такі хостингові служби, як WordPress та Joomla.
- Процес встановлення дуже трудомісткий та тривалий, якщо ви використовуєте багато модулів.
- Ви повинні мати базові знання HTML та інших мов веб-кодування.
- Оновлення розширень не завжди сумісні із серверною частиною.

Але саме Drupal зміг задовільнити наші вимоги до новинного сайту через те, що:

- Створення веб-сторінок, таких як панелі, уявлення та веб-форми, стало простим та високо захищеним.
- Drupal пропонує модулі із привабливими можливостями налаштування.
- Веб-сайт, створений за допомогою Drupal CMS, більш безпечний у порівнянні з іншими CMS.
- Це програмне забезпечення для блогів пропонує простий для розуміння та зрозумілий основний код.
- Допомагає керувати комплексно, багатодоменне управління.
- Безліч вбудованих можливостей керування сайтом.
- Доступ до тисяч модулів та тем.
- Можливість налаштувати кожний аспект сайту.

Дослідження різних систем управління контентом показало, що CMS Drupal відповідає найкраще вимогам для створення та управління новинним сайтом університету ХДУ. Враховуючи особливості, виявлені під час аналізу інших новинних сайтів, і зокрема, акцент на швидкість розгортання та редагування контенту, Drupal виявився ідеальним варіантом для втілення потреб проекту.

Гнучкість та розширюваність: Drupal - це потужна система управління контентом, яка надає широкий набір функціональних можливостей та можливість легко розширювати функціонал за допомогою модулів.

Велике співтовариство користувачів і розробників: Drupal має велике співтовариство користувачів та розробників, що забезпечує підтримку, розвиток та велику кількість готових рішень для швидкого впровадження. Drupal надає ефективне управління кешуванням та оптимізацію швидкодії, що дозволяє забезпечити високу продуктивність для великих обсягів контенту та великої кількості користувачів.

Аналітика та звітність: Бекенд повинен забезпечувати збір та обробку даних для подальшого аналізу та створення звітів про

відвідуваність, залучення аудиторії та інші метрики ефективності сайту. Спільнота та підтримка: Drupal має активну спільноту користувачів та розробників, яка забезпечує підтримку, розвиток та постійні оновлення платформи.

Готові рішення: Велика кількість готових модулів та тем для Drupal дозволяє швидко впроваджувати необхідний функціонал без необхідності розробки з нуля. Широкі можливості налаштування та розширення: Drupal надає гнучкі можливості налаштування та розширення, що дозволяє адаптувати платформу під конкретні потреби новинного сайту.

Загалом, Drupal відповідає усім вимогам для успішного розгортання та управління новинним сайтом університету ХДУ, і врахування його переваг зробить процес розробки більш ефективним та продуктивним.

1.4. Основні безпекові заходи для підтримки та стабільної роботи сайту.

На час коли розробляється наш продукт, Drupal сайтів небагато більше ніж 2% від усіх сайтів, а тобто 2 мільйони сайтів які використовують Drupal. Ми побачили, що проблеми з безпекою в Drupal дуже схожі на ті, які ми можемо знайти в будь-якій іншій CMS, але є кілька більш специфічних аспектів. Drupal надає нам змогу багато чого робити зі своїм веб-сайтом. Нехай ця зручність не дозволить вам бути самовдоволеними цими важливими вразливими місцями та загрозами.

Коли розробники випускають оновлення для тем і модулів, вони часто включають виправлення безпеки для відомих уразливостей Drupal. Затримка цих оновлень може викликати атаки зловмисників. Оновлення тем і розширень допомагає уникнути загроз веб-сайту Drupal. Перш ніж починати будь-яке оновлення вашого веб-сайту Drupal, потрібно переконатися, що у ми маємо надійні резервні копії, конфіги та дампи бази даних. Це особливо важливо, якщо ми встановлюємо тему чи

модуль уперше. Ми можете перевірити наявність оновлень у адміністратора Drupal. Натиснути треба «Звіти», а потім — «Доступні оновлення». Звідси ми можемо запускати оновлення маючи роль адміністратора. Модулі Drupal дозволяють легко розширити функціональність нашого веб-сайту, і ми використовуємо кілька з них, вдосконалюючи свій веб-сайт. Однак важливо пам'ятати, що кожен модуль може мати власні вразливості, про які нам слід знати. Періодично потрібно проводити перевірку всіх модулів, які ми завантажили, щоб переконатися, що вони все ще потрібні та є найкращим доступним рішенням для бажаної функції. Знайти кожен модуль у репозиторії модулів Drupal, а потім треба оцінити: Супроводжувачі — в ідеалі має бути кілька таких людей, які зобов'язуються підтримувати модуль з оновленнями та виправленнями помилок. Якщо є модуль із меншою кількістю проблем, це може бути кращим вибором. Звіт про помилку — перевіряти частоту повідомлень про помилки, а також час між повідомленням і вирішенням. Документація. Якщо вам потрібна допомога з модулем, якісна документація значно полегшить життя. Розробка — чи активно підтримується цей модуль якщо ні, треба використовувати якийсь аналогічний, але кращий модуль.

Дізнавшись, як покращити безпеку сайту Drupal, ми можемо зосередитися на поточному веденні сайту, а не постійно вирішувати проблеми в міру їх виникнення.

Розглянемо ключові аспекти захисту сайту:

- Обмежити доступ до сайту. Кожна особа, яка має доступ до нашого веб-сайту Drupal, представляє низку ризиків, які загрожують сайту в цілому. Тут доцільно застосувати принцип найменших привілеїв або обмежити доступ лише особам, які цього потребують, і лише на найменший необхідний час. На нашому новинному сайті ми обмежили доступ до адміністративних сторінок усім, окрім адміністратора, а надали можливість керувати

контентом, змінювати терміни таксономії, керувати обліковими записами, створювати контент типи для ролі редактор, що надає нам можливість не перейматись, що людина яка займається публікацією новин зможе потрапити на сторінки, які кардинально змінять, щось на нашому веб-сайті.

- Керувати обліковими записами користувачів. Перевіряти облікові записи користувачів, щоб знати, хто може входити, який тип доступу вони мають і чи все ще їм потрібен цей доступ. Призначили цим людям ролі, які можуть використовувати параметри за замовчуванням або налаштувати, щоб включати: адміністратор — виконує будь-які дії на сайті, наприклад додає вміст, оновлює ролі та дозволи, додає блоки, встановлює модулі, юзер — доступ лише для читання без можливості додавати чи змінювати вміст. редактор — додає вміст з можливістю його змінювати, може змінювати аккаунти користувачі та інше.
- Використовувати надійні паролі. Як і будь-який веб-сайт, Drupal використовує надійні паролі, щоб гарантувати, що кожен обліковий запис стійкий до атак грубої сили. Паролі, які легко запам'ятати або промовити, можуть бути зручнішими, але вони експоненціально більш вразливі для автоматизованих сценаріїв, призначених для злому паролів. Через ці характеристики паролі важче вгадати, а також запам'ятати. Щоб допомогти вам відстежувати свої паролі та керувати ними, використовуйте засіб збереження паролів для зберігання й автоматичного введення паролів. Існує низка доступних безкоштовних додатків із надійною репутацією, зокрема LastPass, KeePass і Myki. Обмеження спроб входу інтерфейсами адміністратора допомагає запобігти атакам грубої сили, коли автоматизовані сценарії намагаються використовувати комбінації імені користувача та пароля, доки один із них не надасть доступ.

- Обмежити доступ Drupal до автентифікованих URL-адрес. Ми використовуємо брандмауер веб-сайту, щоб створити білий список IP-адресів, які мають доступ до нашої сторінки входу. Це гарантує, що лише люди, яких ми знаємо та яким довіряємо, зможуть обійти безпеку та увійти, фактично відсікаючи цей вектор атак, який використовують зловмисники.
- Налаштувати двофакторну автентифікацію. Двофакторна автентифікація – це захід безпеки, який залежить від пристрою, яким ви фізично володієте. Облікові дані для входу можна вкрати або вгадати, але для 2FA потрібен доступ до такого пристрою, як настільний комп'ютер або мобільний пристрій. З 2FA ви також повинні ввести код, надісланий на цей пристрій, щоб увійти у свій обліковий запис. Ми не реалізовували цей безпековий процес, але його використання на сайті який є в інтернеті дуже важливо. Ви можете налаштувати двофакторну автентифікацію через Drupal admin або додавши модуль Two-Factor Authentication (2FA).

Модулі безпеки Drupal. В друпалі ми можемо використати наступні модулі безпеки:

- CAPTCHA — реалізує тест виклик-відповідь, коли користувачі намагаються надіслати дані.
- reCAPTCHA — покращує систему CAPTCHA.
- Резервне копіювання та міграція — створює резервні копії та відновлює бази даних і код MySQL, а також забезпечує простішу міграцію сайту.
- Автоматичний вихід — виходить із системи користувачів після певного періоду бездіяльності.
- Двофакторна автентифікація (TFA) — додає другий етап входу в систему, окрім імені користувача/пароллю, наприклад код, надісланий на мобільний пристрій.

Під час розробки ми будемо використовувати такі модулі, як автоматичний вихід та резервне копіювання відновлювати будь-яку з цих резервних копій за потреби. Деякі модулі Drupal, наприклад Backup і Migrate, обробляють цю функцію. Та багато додатків до модуля Migrate, які набагато спрощують міграцію на новішу версію конструктора сайтів.

Висновки до першого розділу

Перед розробкою сайту, ми вирішили розділити такі ролі, як «адміністратор» та «редактор» та залишити «авторизований користувач», «анонімний користувач». В ядрі Drupal можна одразу створювати ролі користувачів та настроювати їх доступи до різних контентів або сторінок. Тому ми обмежили всі можливі редагування на сайті для звичайних користувачів та редакторів, але надали можливість редактору змінювати, видаляти, редагувати коментарі, статті та профілі користувачів. У авторизованого користувача на відмінну від анонімного є можливість коментувати новини на сайті та мати свій індивідуальний профіль на сайті, додати опис, додати свої соціальні мережі, встановити фотографію.

У цьому розділі ми продивились багато новинних сайтів університетів України та зробили висновки, щодо функціоналу та дизайну сайту. Ми вирішили не відходити від ідеї строгого дизайну для сайту, через те, що зазвичай на новинних сайтах університетів використовується досить простий, але приємний для користувача інтерфейс та інтуїтивно зрозумілий, а щодо функціоналу основні завдання це було: створити роль редактор, надати цій ролі потрібний доступ до різних сторінок та можливостей на сайті. Реалізувати публікування новин та його вигляд.

Drupal задовільнив наші вимоги до створення цього сайту та надав низку можливостей для подальшого підтримання та розроблення нового функціоналу на сайті.

Розглянувши увесь функціонал цього конструктора сайтів ми

прийшли до висновків, що досить не погана ідея використовувати саме його. Так, як через те, що ми не зможемо підтримувати цей сайт, то додавання додаткового функціоналу буде набагато простіше та ефективніше.

РОЗДІЛ 2. Планування сайту та його структури

2.1 Основні функції сайту

Для розробки сайту ми використовуємо версію Drupal 10.3, на час розробки всі актуальні модулі, які використовуються на версії більше 7 підтримують версію 9-10. Зазвичай, якщо навіть остання версія модуля не підтримує вже 9.1, то більш старіші версії саме цього модуля ми можемо знайти та використовувати разом з нашою версією, але скоріш за все усі модулі будуть підтримувати цю версію. Весь шлях по встановленню є на офіційному сайті, де й можна обрати версію. Ми встановлювали за допомогою composer, а локально запускали за допомогою Docker. Також використовували drush, що значно підвищує швидкість виконання будь-яких дій на сайті, такі як оновлення кешу, встановлення модулів.

Під час розробки ми реалізували такі функції для авторизованого користувача:

- Реєстрація
- Авторизування
- Зв'язок з власником сайту
- Можливість коментування
- Перегляд новин
- Пошук новин
- Встановлення свого фото в профілю
- Редагування свого профілю
- Перехід на соціальні мережі
- Поширення новини
- Різні теми новин та сторінки для них

Функціонал для редактора:

- Створення контенту
- Редагування контенту
- Створення опитувань

- Редагування облікових записів користувачів
- Редагування коментарів
- Блокування користувачів
- Видалення контенту
- Вибір чи є новина головною.
- Вибір, яка тема новини.
- Закрити коментарі.
- Розширений текстовий редактор для більшої можливості для написання новин та редагування їх.
- Чорновик. Це поняття надає можливості редакторам викладати свої новини коли вони мають це зробити, а не коли вони були написані. Якщо новина не опублікована, то вони так само зможуть її редагувати.

Окрім цих функцій ми додали:

- Можливість обирати погоду на сторінці користувача, щоб кожен користувач міг бачити важливу та інформативну для нього погоду.
- Додали блок з останніми коментарями, де усі користувачі можуть побачити, що саме в останнє прокоментували та хто. При натиску на коментар користувача перенаправляє на сторінку де саме був написаний цей коментар.
- На сторінці новини збоку справа користувач побачить три останніх схожих новини за тематикою. Вони будуть представлені у вигляді, як на основній сторінці.

Завдяки функціоналу ми добилися того, що він повністю задовільняє приємне, інтерактивне та ефективне користування сайтом для користувача та редактора.

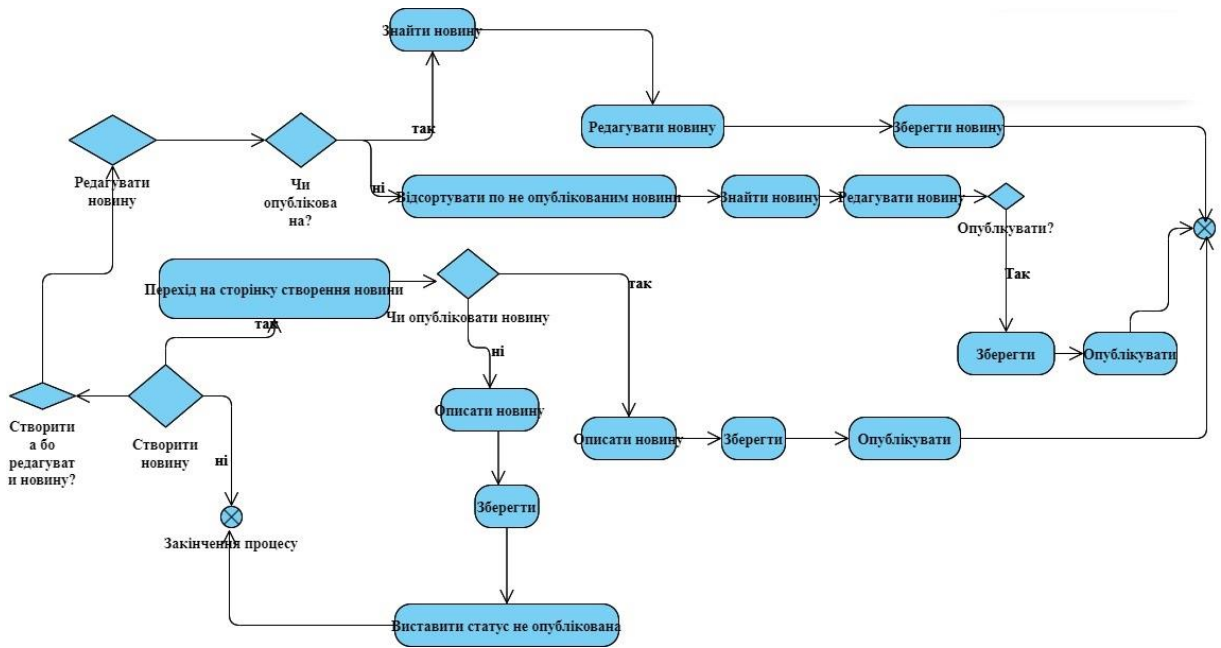


Рисунок 2.1 — структура сайту.

2.2 Структура сайту

Опираючись на визначені функціональні характеристики 1.2, де ми розглядали, що саме має бути на сайті та яку структуру має мати наш сайт потрібно було розробити структуру сайту, яка задовільняє наші вимоги. Див. Рис. 2.1. Тож для цих вимог ми створили таку мапу дій для користувача де буде описано, що саме користувач може робити на сайті і як він це буде реалізовувати. Див. Рис. 2.2.

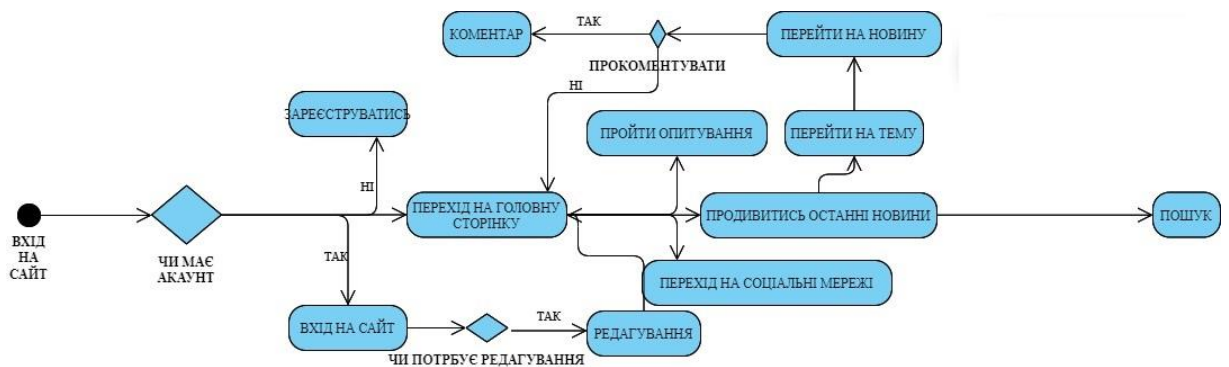


Рисунок 2.2 — мапа дій користувача

Та таку мапу дій для нашої структури для редактора. Див. Рис. 2.3.

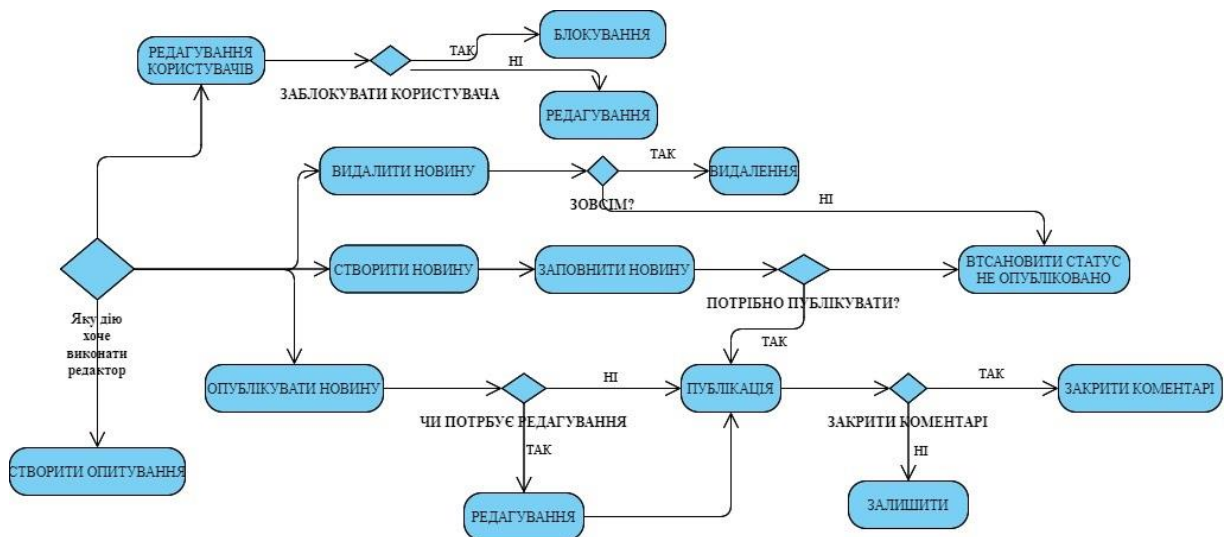


Рисунок 2.3 — мапа дій для редактора

Завдяки такій структурі сайт не виглядає перегруженим та інтуїтивно зрозумілим для нових користувачів або редакторів, що є набагато ефективніше, якщо редактор не зможе зробити чогось лишнього на сайті. Навіть, якщо редактор десь під час написання новини не знайшов, або не зміг зробити, то новину завжди можна не публікувати, а залишити на стані редакції.

На нашу думку — ця структура повністю задовільняє ідею того, що користувачу має бути просто, швидко, інформативно перегляд новин. А редактору ефективно, легко та функціонально.

2.3 Різниця доступів для ролей

У Drupal, інформація про ролі користувачів та їх дозволи зберігається у відповідних таблицях бази даних. Механізм контролю доступу використовує ці дані для визначення прав доступу при кожному запиті. Реалізація цього механізму забезпечується через спеціальний механізм розширень та хуків в Drupal. В Drupal ми можемо створювати нові ролі, які нам захочеться, налаштовувати для них доступи, а саме адміністратор має повний доступ до всіх функціональних можливостей системи без обмежень. Це означає, що він має право на виконання будь-яких дій в адміністративному інтерфейсі Drupal, включаючи управління користувачами, контентом, налаштуваннями сайту та іншими аспектами. Редактор має обмежений, але важливий набір дозволів, що дозволяє

керувати контентом сайту. Серед цих дозволів - управління коментарями, редагування та видалення новин, створення нового контенту, включаючи новини, а також управління полями та статистикою. Для редактора у нас є доступ до такого функціоналу:

- Управління коментарями
- Управління полями:
 1. Мультимедіа
 2. Коментарі
 3. Контент
 4. Параграф
 5. Опитування
 6. Терміни таксономії
 7. Користувач
- Використовує текстовий редактор
- Створення нових матеріалів для новин
- Видалення новин
- Редагування новин
- Відновлення редакцій
- Видалення редакцій
- Перегляд редакцій
- Управління статистикою
- Огляд лічильника переглядів новин
- Використання адміністративної теми для адміністративних сторінок
- Адміністрування налаштувань сторінок
- Управління користувачами

Користувач має обмежений набір дозволів, що обмежується головним чином до перегляду та спілкування з контентом сайту. Він може змінювати свої особисті дані, використовувати деякі інструменти, а саме

- Зміна свого ім'я
- Використання адміністративної теми
- Використання ярликів
- Використання пошуку
- Огляд опублікованих матеріалів.
- Продивитись медіа
- Використання контактної форми
- Додавання коментарів
- Редагування своїх коментарів

Для досягнення цих обмежень доступу та налаштування прав користувачів використовується вбудована система управління дозволами Drupal. Крім того, для розширення можливостей контролю доступу до сторінок, використовується модуль Rabbit Hole, який дозволяє налаштовувати доступ до конкретних сторінок для різних ролей користувачів. Це важливий інструмент для забезпечення конфіденційності та безпеки даних на веб-сайті.

2.4 Організація вхідних/вихідних даних

Під час реєстрації нового користувача в системі Drupal, введені користувачем дані, такі як ім'я, адреса електронної пошти та пароль, передаються на сервер за допомогою форми, звичайно, зашифровані. Пароль зберігається у базі даних в захешованому вигляді, щоб забезпечити безпеку. Зазвичай для цього використовуються криптографічні хеш-функції, такі як SHA-256 або bcrypt, які забезпечують надійний захист паролів.

Авторизація існуючого користувача:

- Передача облікових даних: Користувач вводить свої облікові дані, такі як ім'я користувача та пароль, у відповідні поля на сторінці входу.
- Перевірка ідентифікації: введені дані перевіряються на відповідність обліковим записам в базі даних. Для перевірки

паролів, зазвичай, застосовується метод хешування, щоб порівняти захешований варіант пароля, збережений у базі даних, зі значенням, яке ввів користувач.

- Авторизація: якщо ідентифікація успішна і облікові дані вірні, користувач автоматично авторизується в системі. В цей момент встановлюється сеанс (session), який зберігає інформацію про авторизованого користувача та його права доступу.
- Створення та збереження логів: під час процесу авторизації система може створювати логи, які містять інформацію про дії користувача, такі як спроби входу, час та IP-адреса. Це допомагає відстежувати та аналізувати активність користувачів для забезпечення безпеки та виявлення потенційних загроз.
- Управління сеансами: Під час авторизації, керування сеансами грає важливу роль у забезпеченні безпеки. Drupal зазвичай використовує механізми токенів та куки для ідентифікації авторизованих сеансів та зберігання інформації про них у базі даних.

Коментарі зберігаються у базі даних разом з відповідною статтею або новиною і включають в себе інформацію про автора, дату та час створення. Крім того, можуть бути застосовані правила модерування коментарів для забезпечення безпеки та уникнення спаму.

Механізм публікації новин: У системі управління контентом Drupal, процес публікації новин передбачає використання адміністративного інтерфейсу для створення та редагування контенту. Модуль "Контент" забезпечує зручний інтерфейс для введення інформації, включаючи заголовки, текст та інші характеристики новини. Редактори та адміністратори мають можливість додавати нові матеріали через цей інтерфейс, що робить процес публікації доступним та ефективним. У процесі створення та редагування контенту через адміністративний інтерфейс Drupal, дані вводяться у форму, яка

використовує мову розмітки форм JSON (JavaScript Object Notation) або формату асоціативного масиву. Ці дані обробляються PHP-сценарієм на серверному боці, який перетворює їх у структуру даних Drupal і виконує необхідні перевірки перед збереженням у базі даних.

Зберігання новин: новини, разом з іншим контентом, зберігаються у базі даних, яка використовується Drupal для зберігання інформації. Ця база даних, зазвичай реляційна, а саме MySQL є центральним елементом системи зберігання. Вона забезпечує структуроване та ефективно зберігання даних, включаючи новини, що дозволяє швидкий доступ до них та забезпечує цілісність інформації. Дані у базі даних Drupal зберігаються у вигляді записів у відповідних таблицях, які відповідають структурі сутностей Drupal, таких як "node" (для матеріалів), "field_data_field" (для полів) тощо. Кожен запис має унікальний ідентифікатор (ID), який використовується для швидкого доступу до даних, а також індекси, що полегшують пошук та сортування.

Організація доступу: Для забезпечення безпеки та контролю доступу до новин та іншого контенту, Drupal використовує систему ролей та дозволів. Ця система дозволяє адміністраторам налаштовувати права доступу для різних типів користувачів відповідно до їхніх обов'язків та функцій. Таким чином, адміністратор може контролювати, хто має доступ до новин та іншого контенту, а також які дії вони можуть виконувати з цим контентом. Механізм контролю доступу в Drupal базується на системі ролей та дозволів, яка використовує реляційну базу даних для зберігання прав користувачів. Таблиці "role" та "permission" містять інформацію про ролі користувачів та дозволи відповідно. При кожному запиті до контенту Drupal, виконується перевірка дозволів користувача з урахуванням його ролей та прав.

Оновлення даних: Процес оновлення даних, включаючи новини, може відбуватися як вручну, так і автоматично. Вручне оновлення передбачає редагування контенту через адміністративний інтерфейс. З

іншого боку, автоматичне оновлення може бути налаштоване за допомогою різних модулів та інтеграцій, що дозволяють імпортувати або синхронізувати дані з інших джерел, таких як зовнішні бази даних або інші веб-сайти. Для автоматизації оновлення даних у Drupal використовуються різні інструменти, такі як модулі інтеграції з API, імпорт даних через формати як CSV або XML, а також крон-завдання для планування автоматичних оновлень. При цьому важливо враховувати перевірки на валідність даних перед внесенням їх у базу даних, щоб уникнути можливих проблем з цілісністю та безпекою.

Нормалізація: Нормалізація даних в системі Drupal забезпечує структурованість та цілісність бази даних шляхом усунення дублювання інформації та забезпечення правильної організації даних. Цей процес включає в себе розбиття даних на окремі таблиці та встановлення відносин між ними, щоб забезпечити ефективне зберігання та обробку інформації. Нормалізація даних є важливою складовою процесу розробки та підтримки системи управління контентом. У процесі нормалізації даних у Drupal, може бути використана структура таблиць з різними рівнями нормалізації, такими як перша, друга, або третя нормальна форма. Крім того, для оптимізації продуктивності можуть бути застосовані індекси для часто використовуваних запитів до бази даних.

Висновки до другого розділу

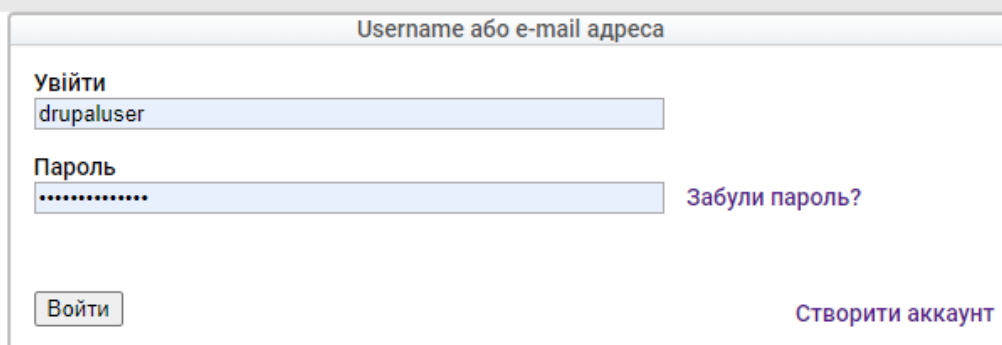
У цьому розділі було ретельно розглянуто процес створення функціоналу веб-сайту, побудованого на платформі Drupal, а також його структуру. Виявлено, що ключовими аспектами є організація вхідних та вихідних даних, а також регулювання доступу до функціоналу сайту залежно від ролі користувача. Підкреслено важливість правильної організації даних, що забезпечує ефективну роботу веб-сайту та оптимізацію його продуктивності. Пояснено, що структурованість даних дозволяє швидко та ефективно здійснювати доступ до інформації та

обробляти її. Також акцентовано увагу на важливості регулювання доступу до функціоналу відповідно до ролі користувача. Встановлення різних рівнів доступу дозволяє забезпечити безпеку інформації та зменшити ризик несанкціонованого доступу до сайту. У висновку, розділ надає повні інструкції щодо створення функціонального та ефективного веб-сайту на основі платформи Drupal, зокрема описуючи процес організації даних та регулювання доступу до функціоналу. Ці кроки є важливими для забезпечення безпеки та швидкодії сайту, а також задоволення потреб його користувачів.

РОЗДІЛ 3. Розробка додаткового функціоналу сайту

3.1 Реалізація функцій сайту

Реєстрація та авторизація є в самому Drupal, але ми вирішили функціонально розширити та доповнити форму реєстрації. В нашому випадку ми вирішили не прибрати цей функціонал та залишити його, ми додали до реєстрації можливість додати своє фото в профіль та опис. Також реалізували можливість додавання своїх соціальних мереж та посилання на них.

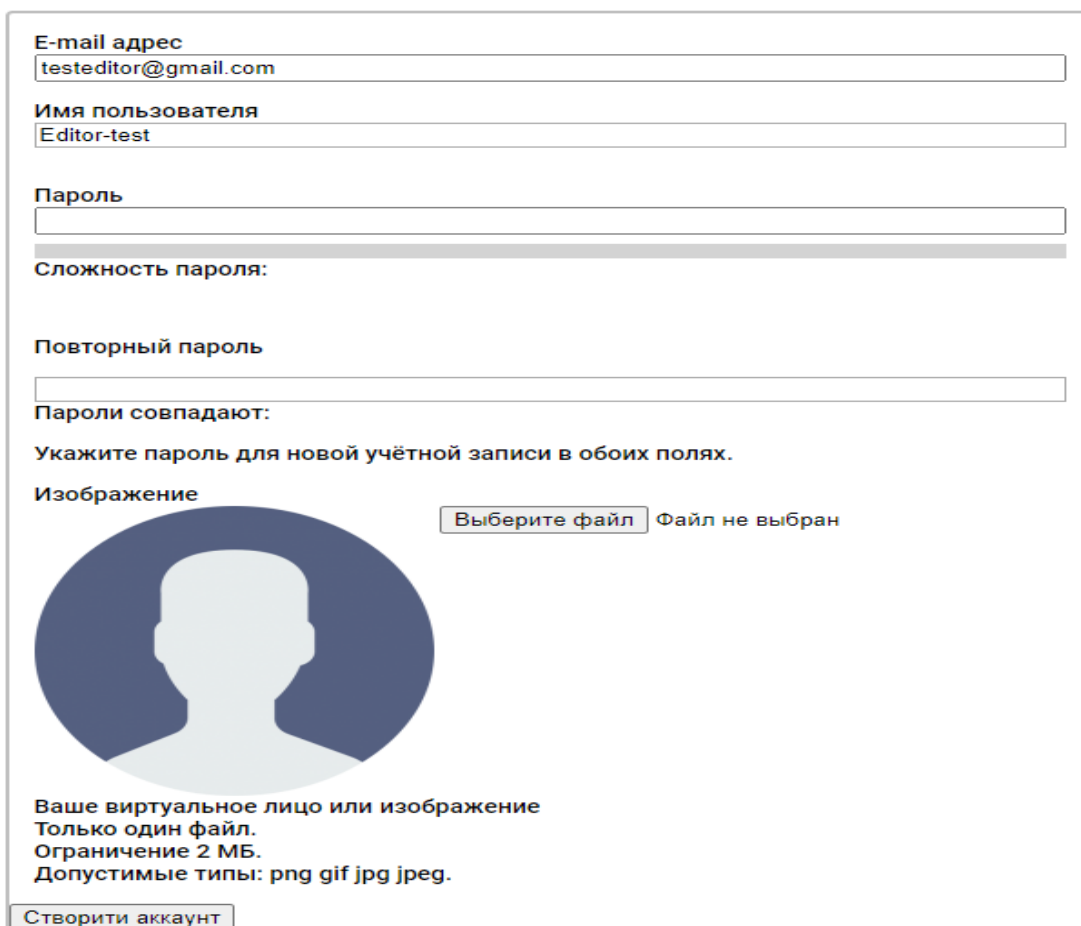


Username або e-mail адреса

Увійти
drupaluser

Пароль
..... [Забули пароль?](#)

Рисунок 3.1 – авторизація на сайті.



E-mail адрес
testeditor@gmail.com

Имя пользователя
Editor-test


Пароль
.....

Сложность пароля:

Повторный пароль
.....

Пароли совпадают:
Укажите пароль для новой учётной записи в обоих полях.

Изображение Файл не выбран



Ваше виртуальное лицо или изображение
Только один файл.
Ограничение 2 МБ.
Допустимые типы: png gif jpg jpeg.

[Повернутись на сторінку входу.](#)

Рисунок 3.2 – реєстрація на сайті.

Зв'язок з сайтом також йде в самому Drupal, але для того, щоб він працював потрібно ще немало чого зробити, а саме інструкція по налаштуванню зворотнього зв'язку в системі має на меті забезпечити ефективний та зручний спосіб для користувачів надсилати свої відгуки та повідомлення адміністраторам веб-сайту. Нижче наведено кроки, які були виконанні для належного налаштування цієї функціональності:

- Встановлення та налаштування модуля Contact.
- Створення форми зворотнього зв'язку.
- Налаштування отримувачів повідомлень.
- Налаштування відображення форми.
- Тестування.
- Моніторинг та підтримка.

Завдяки цьому ми можемо завжди отримати зворотній зв'язок від наших користувачів та їх рекомендації. Надаємо можливість підписатися на розсилку в пошті для нових актуальних новин та можливість відновлення паролю або надсилання користувачам попереджувальних листів. Правильно налаштований зворотній зв'язок сприяє покращенню комунікацій з користувачем, сприяє виявленню проблем та допомагає забезпечити задоволення їх потреб.

Коментарі можна виключити в налаштуваннях новини, але ми вирішили їх залишити по стандарту, саме в Drupal бекенд реалізація коментарів не сама легка та функціонально звужена, тому зазвичай, якщо коментарі залишають, то змінюють лише шаблон самого коментаря, а саму форму не чіпають.

Перегляд новин — це окрема сторінка новини, яка має повний контент новини, а не обмежений, як на головній сторінці. Під час натискання на заголовок, або картинку новини на головній сторінці користувача перенаправляє на сторінки новини, а також при натисканні на термін таксономії, тобто теми, то його перенаправляє на сторінку

теми, де всі новини сортуються по цій темі та показуються лише актуальні для користувача інтересів теми.

Пошук новин у нас відбувається за допомогою звичайного Drupal пошуку, але й можна використовувати Solr для більш глибоко та точного пошуку.

Встановлення свого фото в профілю відбувається під час реєстрації, саме Drupal дозволяє завантажити лише три типи файлу: jpg, jpeg, png. Тобто gif, svg і т.д. не можливо завантажити, але якщо є в цьому потреба завжди можна знайти contrib модуль який надасть можливість розширити типи файлів.

Ми надали можливість користувачам редагувати свій профіль. Він може змінити нікнейм, фото, електронну пошту, пароль, опис.

Опираючись на досліди першого розділу ми зазначили, що зазвичай всі сайти також мають посилання на свої різні соціальні мережі. Тож ми вирішили використати ці здобуті дослідницькі знання та використати їх також у нашому сайті та показувати у нижній частині чотири соціальні мережі університету, а саме: instagram, facebook, youtube, twitter. Та при натисканні на іконки цих соціальних мереж користувача перенаправляє на сторінку соціальної мережі на акаунт університету.

Для різних термінів таксономії(тем сайту) ми використовуємо різні сторінки. Для всіх цих термінів є сторінки, де всі останні актуальні новини сортуються за цією темою та виводяться на екран користувача.

3.2 Встановлення contrib модулів.

Drupal надає можливість використання широкого спектру contrib модулів, які розроблені спільнотою розробників. Ці модулі дозволяють розширити функціональність сайту без необхідності написання власного коду. Наприклад, ви можете використовувати модуль Views для створення різних типів списків, модуль Pathauto для автоматичного генерування URL-адрес для сторінок контенту, модуль Webform для

створення форм зворотного зв'язку та багато інших. Крім того, ви також можете використовувати ваші власні створені модулі.

Список встановлених contrib модулів та їх призначення:

- **CKEditor**: цей модуль дозволяє переміщувати текстові поля за допомогою WYSIWYG редагування. Цей HTML редагувальник має в собі багато функціоналу для форматування різних видів тексту. Використовується на нашому сайті задля публікації та редагування новин в потрібно для нас вигляді.
- **Layout Builder**: цей модуль надає можливість переміщувати block, menu, view як ми хочемо на будь якій сторінці та створювати секції з автоматичним розподілом на колонки і їх розмірами.
- **Statistic**: модуль який надає можливість слідкувати за активністю на різних контент-типах.
- **Admin Toolbar** та його залежності: використовується для поліпшення використання адмін-панелі. Та надання більшого функціоналу.
- **Rabbit Hole**: завдяки цьому модулю ми можемо обмежувати доступ до різних сторінок таких як: сторінка профілю адміністратора, сторінка яка перебуває в розробці і т.д.
- **Super Login**: модуль який розширює функціонал звичайної сторінки авторизації
- **Easy-breadcrumb**: додає до нашого сайту меню з історією переходу на інші сторінки.
- **Mail-System**: модуль, що надає можливість надсилати на пошту сайту електронні письма.
- **Views Slideshow**: додає format для view, щоб створювати Slideshow для різних видів та в різних місцях.
- **Asset Injector**: додає можливість додавати на сайт JavaScript, CSS код не заходячи в його код.

3.3 Розробка модулів для збільшення функціоналу.

Спираючись на пункт 1.2 ми вирішили додати на сайт погоду. Для цього ми розробили 2 модулі.

У модуля з погодою така структура, що у вкладенні розташування блоків ми повинні розташувати наш блок там де він має бути та обрати стандартне значення міста та ввести API ключ. Для створення модуля з погодою для початку треба ініціалізувати його, щоб Drupal розпізнав його, як новий модуль і дав встановити його. Для ініціалізації ми повинні в файлі `.info.yml` описати ім'я модуля, тип, опис, для яких версій конструктора сайтів він підходить та категорія модуля

Для реалізації цього модуля я створив блок. Для цього мені потрібно створити клас з імям `ModuleNameBlock.php`. Цей клас має успадковувати `BlockBase` для створення саме блоку та реалізовувати `ContainerFactoryPluginInterface` – це дає нам змогу використовувати `dependency injections`, а це в свою чергу сервіс який дозволяє нам використовувати об'єкт який управляється контейнером сервісів. `Dependency Injections` значно полегшує написання та роботу unit тестів для нашого модуля

Наш блок створює нову таблицю в базі даних де ми записуємо та отримуємо значення міст, погоди та вибір користувачів. Для того, щоб створити таблицю ми повинні реалізувати `hook_schema()` в `module_name.install`.

Для того, щоб наш блок відображав бажані нами поля на сторінці ми повинні реалізувати функцію `build`, де ми спочатку отримуємо поточного користувача та його `id`, завантажуюмо з бази даних його вибір міста, а якщо його немає валідуємо та повертаємо пустий масив в цій функції, а тобто це призведе до того, що блок не буде показуватись зовсім. Тож якщо ми отримали значення з бази даних. Потім ми перевіряємо чи є закешовані дані з цим містом зараз в кешу[8], якщо є — віддаємо закешовані дані, якщо немає — викликаємо метод `getDataFromApi` сервіса `FetchApiData.php`, який нам в свою чергу

повертає масив в якому масив з ключем — назвою міста, значенням — його температура, та ще один масив з ключем стан погоди і значенням — стан погоди на англійській мові. Після цього ми отримуємо з конфігурації апі ключ, та перевіряємо, якщо апі ключ пустий і повертає ‘no key’ то ми повертаємо пустий масив, щоб помилки не з’являлись на екрані користувача, а в журнал помилок надсилаємо помилку яку нам повертає сервер. Якщо апі ключ повернувся правильний і сервер не відповів помилкою кешуємо нові дані та повертаємо користувачу.

В самому кінці функції ми повертаємо масив для відтворення нашого блоку. Де ми описуємо, який шаблон для блоку буде використовуватись, яке наповнення блоку, та в нашому випадку ще тип погоди та використання css файлу.

У кешування в Drupal є три змінні, max-age – час в секундах на який він має кешуватись, context – під час яких дій або змін він повинен кешуватись, та tags – відносно чого він повинен кешуватись. Ми в даній задачі використовували лише tags та max-age, бо при будь-яких змінах Drupal розуміє, що дані застаріли та потребують оновлення. Для встановлення цих змінних ми при кешуванні даних одразу можемо передати наші значення, `$this->cacheBackend->set($cache_id, $api_data, time() + 3600, $tags);` `$tags` — масив який ми отримуємо з метода `getCacheTags()`. Для того, щоб налаштування модуля та його погоди були доступні для всіх, а не тільки для адміністратора — я створив додатково до блоку ще форму. Для її ініціалізації ми повинні успадковувати `FormBase` де маємо реалізувати три методи `buildForm` – уже створена форма на сторінці, `getFormId` – метод який повертає унікальне ім’я форми, `submitForm` – метод який реагує на натискання кнопки `submit` та робить якісь дії. В `buildForm` ми використовуємо метод `getCitiesArray` раніше підключеного сервіса `GetSetCityUser.php`, щоб отримати масив з усіма містами України. Після цього створюємо форму з описом, ще одну форму де користувач може обрати з випадуючого списку місто погоду

якого він хоче бачити та ще одна форма, яка створює кнопку submit. В кінці цього метода ми повертаємо \$form.

При натисканні кнопки submit ми отримуємо id користувача, та його вибір в формі. Якщо вибір не пустий ми викликаємо метод saveCitiesForUser сервіса GetSetCityUser.php який в свою чергу перевіряє чи є в базі даних запис для цього користувача, якщо є — то видаляє та вставляє в поля user_id id користувача, city_name — обране місто. Після чого ми отримуємо з конфігурації API ключ та викликаємо з сервісу FetchApiData.php метод getDataFromApi де ми передаємо значення обраного міста та значення API ключа та отримуємо значення погоди для обраного міста і записуємо ці значення в базу даних.

Для того, щоб наша форма десь з'явилась нам потрібно прописати шлях по якому буде доступна ця форма для цього потрібно реалізувати module_name.routing.yml де ми описуємо який шлях до форми буде, який клас ми використовуємо для цієї форми, який заголовок та хто має доступ.

Реалізація module_name.routing.yml

Після правильного створення модулю, я приступив до розробки його. Я отримую данні погоди за допомогою API. Для цього я створив class CityForm який наслідує FormBase моя форма має такі методи:

- getFormId, яка повертає мою машинну назву моєї форми.
- buildForm, яка створює форму, описує які поля будуть в цій формі та повертає її. Також я там викликаю метод в змінній \$weather_data метод getWeatherData.
- GetWeatherData, має масив \$cities зі всіма містами України та змінну \$api_key з api ключем. Після чого я проходжу по всьому масиву і в кожному проході циклу, використовуючи елемент масиву та api ключ, формую url посилання на сторінку з актуальною інформацією про місто та його погоду, отримую значення з посилання, декодую json формат та записую в змінну

\$data. Використовую логічний вираз та записую місто — значення в новий масив. Та повертаю цей масив.

- `FormatWeatherData`, метод який форматує, мій масив з ключем-значенням в той формат, що мені треба.

Після цього створив `class WeatherBlock` який наслідує `BlockBase`, для того, щоб Drupal визначив мій модуль, як блок і надав можливості блоку, а тобто використовувати його де мені треба. В цьому класі в мене є метод: `build` цей метод описує як буде виглядати мій блок, які будуть поля та повертає це.

Register

Form

Цей модуль реалізує кастомну реєстраційну форму для користувачів Drupal, що дозволяє збирати додаткові дані під час реєстрації.

Клас `NewUserRegisterForm` відповідає за створення кастомної реєстраційної форми для користувачів. Він успадковує стандартну форму Drupal `RegisterForm` та додає додаткові поля та функціонал для збору додаткової інформації.

Конструктор класу приймає різноманітні залежності, такі як сервіси для роботи з базою даних, керування сесіями та інші сервіси. Ці залежності використовуються для виконання операцій збереження та обробки даних користувачів.

Метод `form` визначає вигляд та поведінку реєстраційної форми. Він додає поля для вибору країни, міста та інтересів користувача, які доповнюють стандартні поля форми Drupal.

Методи `getCountryOptions` та `getTaxonomyTerms`. Ці методи використовуються для отримання списку країн та термінів таксономії, які використовуються у вибірках для полів країни та інтересів в реєстраційній формі.

Методи `submitForm` та `save`. Ці методи відповідають за обробку та збереження даних, введених користувачем у формі. Вони виконують

операції збереження додаткових полів користувача в базу даних для подальшого використання.

Клас `RegisterFormPopup` реалізує блок для відображення посилання на реєстраційну форму у вигляді спливаючого вікна. Це дозволяє залучати користувачів до реєстрації на сайті, представляючи їм зручний спосіб доступу до форми.

Конструктор класу приймає різноманітні залежності, такі як сервіси для генерації посилань та керування поточним користувачем. Ці залежності використовуються для створення посилання на реєстраційну форму та відображення його у вигляді блоку.

`build` цей метод визначає вміст блоку та включає посилання на реєстраційну форму у спливаючому вікні. Він перевіряє, чи користувач є анонімним, тобто неавторизованим, та відображає посилання на реєстраційну форму у відповідності до цього.

Сервіс `ExtraFieldsHandler`. Цей сервіс відповідає за збереження додаткових полів користувача у базу даних. Він використовується для збереження інформації, введеної користувачем у додаткові поля реєстраційної форми, які не включені у стандартну схему користувача Drupal.

Модуль кастомної реєстраційної форми для Drupal є потужним інструментом для збору додаткової інформації від користувачів під час реєстрації на сайті.

3.4 Забезпечення безпеки та швидкодії.

Задля комфортного користування сайтом для користувачів та редакторів обов'язково було реалізувати безпекові та швидкодійні заходи на сайті. Для забезпечення безпеки сайту було використано ряд заходів, спрямованих на запобігання можливим загрозам та вразливостям:

1. Використання HTTPS: Сайт використовує протокол HTTPS для захищеної передачі даних між клієнтом та сервером, забезпечуючи

конфіденційність та цілісність інформації.

2. Оновлення залежностей: Здійснено регулярне оновлення всіх залежностей, включаючи модулі Drupal, з метою виправлення вразливостей та забезпечення безпеки системи.
3. Захист від SQL-ін'єкцій: Застосовані безпечні методи обробки даних та запитів до бази даних для запобігання SQL-ін'єкціям та вразливостям, пов'язаним з цим.
4. Захист від XSS-атак: Використані безпечні практики виведення даних на сторінці для запобігання міжсайтовому скриптіngu (XSS) та іншим атакам на вразливості в браузері.
5. Автентифікація та авторизація: Застосовані належні механізми автентифікації та авторизації, забезпечуючи доступ до функціоналу сайту лише авторизованим користувачам та контролюючи їхні права.

Для підвищення швидкодії та продуктивності сайту використано наступні заходи:

1. Кешування: Використані механізми кешування для збереження копій часто використовуваних даних та ресурсів, що дозволяє прискорити завантаження сторінок та зменшити навантаження на сервер.
2. Оптимізація зображень і файлів: Здійснено оптимізацію зображень та інших статичних файлів для зменшення їхнього обсягу та покращення швидкості завантаження.
3. Мінімізація та злиття CSS та JavaScript файлів: Використано підходи до мінімізації та злиття CSS та JavaScript файлів з метою зменшення їхнього обсягу та покращення швидкодії сайту.
4. Асинхронна загрузка ресурсів: Застосовано асинхронну загрузку ресурсів, таких як JavaScript, для забезпечення паралельного завантаження та покращення швидкодії сторінок.
5. Кешировані запити до бази даних: Використано механізми

кешування запитів до бази даних для збереження результатів запитів та зменшення часу відповіді сервера.

6. CDN (Content Delivery Network): Використано CDN для доставки статичних ресурсів користувачам з найближчого сервера, що сприяє прискоренню завантаження контенту.

Висновки до третього розділу

У даному розділі було ретельно розглянуто процес реалізації функцій сайту. Розглянуті ключові аспекти, такі як використання важливих модулів Drupal та їхнє призначення, розробка власних модулів для вирішення конкретних завдань, а також заходи, спрямовані на забезпечення безпеки та швидкодії сайту. Описані методи та підходи до забезпечення безпеки, такі як використання HTTPS, оновлення залежностей, захист від SQL-ін'єкцій та XSS-атак, автентифікація та авторизація, виявляються надійними стратегіями для захисту веб-додатків від потенційних загроз. Поряд з цим, використання кешування, оптимізація ресурсів, мінімізація та злиття файлів CSS та JavaScript, а також інші методи оптимізації сприяють покращенню швидкодії сайту та забезпеченню позитивного користувацького досвіду. У цілому, дана розділ технічної документації надає повну картину процесу розробки та захисту веб-сайту на основі Drupal, вказуючи на важливість правильного підходу до вибору засобів та заходів для забезпечення безпеки та швидкодії.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було проведено глибокий аналіз новинних сайтів та їх функціональності, що дозволило нам чітко визначити вимоги до нашого новинного сайту. Огляд відомих систем управління контентом (CMS) дозволив нам вибрати Drupal як оптимальний варіант для реалізації проекту, забезпечуючи широкі можливості та підтримку спільноти.

Планування структури сайту включало в себе розробку основних функцій, визначення структури та організацію доступу для різних ролей. Реалізація додаткового функціоналу сайту була виконана шляхом реалізації необхідних функцій, використання відповідних модулів та розробки власних модулів для задоволення специфічних потреб.

Забезпечення безпеки та швидкодії сайту було однією з ключових задач під час розробки. Використання безпечних практик програмування, регулярне оновлення модулів та застосування оптимізації дозволило забезпечити високий рівень захисту та ефективну швидкодію ресурсу.

У висновку, виконана робота дозволила належним чином спланувати, реалізувати та забезпечити безпеку та швидкодію нашого новинного сайту, що відповідає вимогам сучасного інтернет-простору та забезпечує комфорт користувачів під час використання ресурсу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

1. Устенко, М. О., and А. О. Руських. "Діджиталізація: основа конкурентоспроможності підприємства в реаліях цифрової економіки." Вісник економіки транспорту і промисловості 68 (2019).
2. Волинський національний університет імені Лесі Українки. URL: <https://vnu.edu.ua/uk> (дата звернення 18.02.2024)
3. Новини. Луцький національний технічний університет. URL: <https://lntu.edu.ua/uk/media/news> (дата звернення 18.02.2024).
4. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського». URL: <https://kpi.ua/> (дата звернення 18.02.2024).
5. Новини. Львівський національний університет. URL: <https://lnu.edu.ua/news/> (дата звернення 18.02.2024).
6. Новини. Херсонський державний університет. URL: <https://www.kspu.edu/> (дата звернення 18.02.2024)
7. PATEL, Savan K.; RATHOD, V. R[†]; PARIKH, Satyen. Joomla, Drupal and WordPress-a statistical comparison of open source CMS. In: *3rd International Conference on Trendz in Information Sciences & Computing (TISC2011)*. IEEE, 2011. p. 182-187.
8. PATEL, Savan K.; RATHOD, V. R[†]; PARIKH, Satyen. Joomla, Drupal and WordPress-a statistical comparison of open source CMS. In: *3rd International Conference on Trendz in Information Sciences & Computing (TISC2011)*. IEEE, 2011. p. 182-187.
9. REDDING, Jacob. *Beginning Drupal*. John Wiley & Sons, 2010.
10. BYRON, Angela, et al. *Using drupal*. " O'Reilly Media, Inc.", 2008.