

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інфокомунікації
(повна назва)

Кафедра Інформаційно-мережної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Вдосконалення системи управління контентом АЕМ для
розробки вебсайтів телекомунікаційних компаній
(тема)

Виконав:

здобувач 2 року навчання,
групи ІМІМ-23-1
Чистюк Д.С.

Спеціальність 172 Електронні комунікації
та радіотехніка
(код і повна назва спеціальності)

Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційно-мережна
інженерія
(повна назва освітньої програми)

Керівник доц., к.т.н. Чеботарьова Д.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Безрук В.М.
(прізвище, ініціали)

2025 р.

Не містить відомостей, заборонених до відкритого публікування

Здобувач	_____	<i>Чистюк Д.С.</i>
	(підпис)	(прізвище та ініціали)
Керівник	_____	<i>Чеботарьова Д.В.</i>
	(підпис)	(прізвище та ініціали)

Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій
(повна назва)

Кафедра Інформаційно-мережної інженерії
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 172 Електронні комунікації та радіотехніка
(код і повна назва)

Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційно-мережна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ІМІ _____
(підпис)

“ _____ ” _____ 2025 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

здобувачеві Чистюку Дмитру Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Вдосконалення системи управління контентом АЕМ для
розробки вебсайтів телекомунікаційних компаній

затверджені наказом університету від 28 жовтня 2024 року № 1148 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 14 січня 2025 р.

3. Вихідні дані до роботи _____

Розглянути основні компоненти систем управління контентом. Описати процес розвитку та архітектурні особливості цих систем. Визначити тенденції ринку систем управління контентом та проаналізувати сучасні CMS, такі як WordPress, Shopify, Wix. Дослідити функціональні особливості платформи АЕМ. Розробити проєкт вдосконалення системи АЕМ для телекомунікаційних компаній. Описати структуру, архітектуру та основні можливості розробленої програми.

4. Перелік питань, що потрібно опрацювати в роботі _____

Вступ.

1. Системи управління контентом.

2. Аналіз сучасних CMS.

3. Дослідження системи АЕМ.

4. Проєкт вдосконалення системи АЕМ для телекомунікаційних компаній.

Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Слайди у форматі Power Point (назва, мета і задачі роботи, визначення та компоненти CMS, розвиток CMS, архітектура CMS, тенденції ринку CMS, система управління контентом WordPress, система управління контентом Shopify, система управління контентом Wix, технологічний стек CMS АЕМ, інфраструктура АЕМ, створення та управління контентом, керування вебсайтом, автоматизація управління контентом, політики формування вартості, структура проекту, архітектура проекту, sling model, servlet, service, workflow, utils, додана функціональність для розробки вебсайтів телекомунікаційних компаній, excel документи, завантаження документа, діаграма діяльності процесу створення компонентів, компонент tariff comparator, компонент product slider, компонент offer, висновки)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів атестаційної роботи	Строк виконання етапів роботи	Примітка
1	Ознайомлення із завданням. Уточнення ТЗ	28.10.24	виконано
2	Підбір літератури за темою роботи	28.10 - 02.11.24	виконано
3	Виконання розділу 1	03.11 - 19.11.24	виконано
4	Виконання розділу 2	20.12 – 02.12.24	виконано
5	Виконання розділу 3	03.12 – 19.12.24	виконано
6	Виконання розділу 4	20.12 - 02.01.25	виконано
7	Оформлення пояснювальної записки	03.12 - 07.01.25	виконано
8	Оформлення презентаційного матеріалу, підготовка до захисту у ЕК	08.01 - 14.01.25	виконано

Дата видачі завдання 28.10.2024 р.

Здобувач

_____ (підпис)

Чистюк Д.С.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Чеботарьова Д.В.

_____ (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка: 72 с., 33 рис., 19 джерел, 2 додатки.

Об'єкт дослідження – система управління контентом АЕМ.

Мета роботи – дослідити системи управління контентом АЕМ та розробити проєкт вдосконалення АЕМ для розробки сайтів телекомунікаційних компаній.

Результати – в роботі розглянуто визначення та основні компоненти систем управління контентом. Описано процес розвитку та архітектурні особливості цих систем. Визначено тенденції ринку систем управління контентом та проаналізовано сучасні CMS: WordPress, Shopify, Wix. Детально досліджено функціональні особливості платформи АЕМ. Розроблено проєкт вдосконалення системи АЕМ для телекомунікаційних компаній. Описано структуру, архітектуру та основні можливості розробленої програми.

АЕМ, CMS, КОНТЕНТ, УПРАВЛІННЯ, СИСТЕМА, ВЕБСАЙТ,
ТЕЛЕКОМУНІКАЦІЙНА КОМПАНІЯ, ПРОЄКТ, АРХІТЕКТУРА

THE ABSTRACT

Explanatory note: 72 p., 33 fig., 19 sources, 2 app.

The object of study is the AEM content management system.

Objective - to explore AEM content management systems and develop a project to improve AEM for the development of telecommunications companies' websites.

Results - the work considers the definition and main components of content management systems. The development process and architectural features of these systems are described. Trends in the content management system market are identified and modern CMS are analyzed: WordPress, Shopify, Wix. The functional features of the AEM platform are studied in detail. A project for improving the AEM system for telecommunications companies has been developed. The structure, architecture, and main features of the developed program are described.

AEM, CMS, CONTENT, MANAGEMENT, SYSTEM, WEBSITE,
TELECOMMUNICATIONS COMPANY, PROJECT, ARCHITECTURE

ЗМІСТ

	С.
ПЕРЕЛІК СКОРОЧЕНЬ.....	7
ВСТУП.....	9
1 СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ	10
1.1 Визначення та компоненти CMS.....	10
1.2 Розвиток CMS.....	11
1.3 Архітектура CMS	13
2 АНАЛІЗ СУЧАСНИХ CMS	17
2.1 Тенденції ринку CMS	17
2.2 Система управління контентом WordPress	18
2.3 Система управління контентом Shopify	20
2.4 Система управління контентом Wix	22
3 ДОСЛІДЖЕННЯ СИСТЕМИ АЕМ.....	24
3.1 Технологічний стек.....	24
3.2 Інфраструктура АЕМ.....	25
3.3 Створення та управління контентом.....	27
3.4 Керування вебсайтом.....	28
3.5 Автоматизація управління контентом	29
3.6 Політики формування вартості	31
4 ПРОЄКТ ВДОСКОНАЛЕННЯ СИСТЕМИ АЕМ ДЛЯ РОЗРОБКИ ВЕБСАЙТІВ ТЕЛЕКОМУНІКАЦІЙНИХ КОМПАНІЙ.....	33
4.1 Структура проєкту	33
4.2 Архітектура проєкту	36
4.3 Додана функціональність для розробки вебсайтів телекомунікаційних компаній.....	42
ВИСНОВКИ.....	50
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	51
ДОДАТОК А СЛАЙДИ ПРЕЗЕНТАЦІЇ.....	53
ДОДАТОК Б ПУБЛІКАЦІЇ ЗА ТЕМАТИКОЮ РОБОТИ.....	69

ПЕРЕЛІК СКОРОЧЕНЬ

ADI (Artificial design intelligence) – штучний інтелект спеціалізований на виконанні дизайн задач;

AEM (Adobe Experience Manager) – система управління контентом від компанії Adobe;

AEMaaS (AEM as a Cloud Service) – хмарна система управління контентом від компанії Adobe;

API (Application Programming Interface) – інтерфейс програмування застосунків;

CDA (Content Delivery Application) – програма доставки контенту;

CMA (Content Management Application) – програма управління контентом;

CMS (Content Management System) – система управління контентом;

CRM (Customer Relationship Management) – система управління відносинами з клієнтами;

CRX (Content Repository Extreme) – репозиторій для контенту в AEM;

DAM (Digital Asset Management) – управління цифровими ресурсами;

DXP (Digital Experience Platform) – платформа для керування цифровими ресурсами;

ERP (Enterprise Resource Planning) – планування ресурсів підприємства;

FTP (File Transfer Program) – програма для передачі файлів;

HTL (HTML Template Language) – мова шаблонів HTML;

IoT (Internet of Things) – платформа інтернет речей;

JCR (Java Content Repository) – репозиторій java коду;

JSON (JavaScript Object Notation) – запис об'єктів JavaScript;

JSP (Jakarta Server Pages) – сторінки Jakarta сервера;

MSM (Multi Site Managment) – управління вебсайтами;

OSGi (Open Service Gateway Initiative) – ініціатива відкритих сервісних шлюзів;

POM (Project Object Model) – об'єктна модель проекту;

POS (Point-of-Sale) – точка продажу;

REST (Representational State Transfer) – передача репрезентативного стану;

RSS (Really Simple Syndication) – формат простого отримання інформації;

SEO (Search Engine Optimization) – пошукова оптимізація вебсайту;

SSI (Server Side Includes) – серверне включення файлів;

UML (Unified modeling language) – уніфікована мова моделювання;

URL (Uniform Resource Locator) – уніфікований локатор ресурсів;

VPN (Virtual Private Network) – віртуальна приватна мережа;

WCMS (Web Content Management System) – система управління веб-контентом;

WYSIWYG (What-You-See-Is-What-You-Get) – що бачиш це те, що отримуєш;

XML (Extensible Markup Language) – розширяйма мова розмітки;

ШІ – штучний інтелект.

ВСТУП

У сучасну епоху цифрових технологій попит на швидкі та надійні телекомунікаційні послуги продовжує зростати. Здатність надавати персоналізований, ефективний і масштабований вебсервіс стає ключовою конкурентною перевагою. Індивідуальний підхід дозволяє збільшити результативність компаній, підвищити продажі та знизити витрати на залучення нових клієнтів. Для досягнення цього важливу роль відіграють системи керування контентом (CMS).

Керування вебсайтом без CMS може бути складним завданням. Проблеми змінюються від редагування складних статичних вебсторінок до тривалого очікування впровадження оновлень. Зазвичай цей процес займає дорогоцінний час, який розробники не можуть дозволити собі втрачати в поточному темпі розвитку вебсайтів.

Система керування контентом може допомогти у створенні, управлінні та публікації вмісту вебсторінки. Маркетологи зможуть створювати контент і керувати наповненням вебсайту самостійно, без постійних запитів до розробників, оскільки навіть незначні налаштування та оновлення можуть призводити до значних витрат часу на непотрібну роботу.

Однією з нових потужних та гнучких CMS є Adobe Experience Manager (AEM). Головною особливістю AEM є можливість керувати великими обсягами контенту та забезпечувати його адаптацію для різних платформ, що є критично важливим для телекомунікаційних компаній. Тоді як можливість інтеграції з іншими продуктами Adobe, наприклад Analytics, Photoshop, Campaign, дозволить організаціям отримувати повноцінний огляд взаємодії користувачів з наповненням вебсторінки та забезпечувати більш адаптований матеріал для кожного клієнта.

Однак, незважаючи на те, що AEM пропонує великий набір функціональних можливостей, його конфігурація за замовчуванням часто є загальною, а отже вимагає суттєвих налаштувань та модифікацій для задоволення конкретних потреб. Зважаючи на це та потреби телекомунікаційних бізнесів тема вдосконалення системи управління контентом AEM для розробки вебсайтів телекомунікаційних компаній є надзвичайно актуальною у сучасному цифровому світі.

1 СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ

1.1 Визначення та компоненти CMS

Система управління контентом (CMS) – це програмне забезпечення, яке допомагає користувачам створювати, керувати, зберігати та змінювати контент вебсайту. Контент у CMS зберігається в базі даних і відображається на основі набору шаблонів, ізольованих компонентів та інших цифрових ресурсів.

Функціонально CMS поділяється на два основні компоненти:

– програма управління контентом (Content Management Application, CMA),

– програма доставки контенту (Content Delivery Application, CDA) [1].

CMA забезпечує зручний інтерфейс, що дозволяє користувачам створювати різні типи контенту або змінювати наявний. Зміни можна переглянути в реальному часі завдяки функціям попереднього перегляду платформи. Контент-менеджери можуть структурувати контент, розподіляючи його за категоріями, тегами або рубриками. Це робить контент більш організованим та доступним для подальшого використання. Окрім цього CMA дозволяє встановлювати різні рівні доступу для різних користувачів або груп користувачів, забезпечуючи контроль над тим, хто оперує конфіденційною інформацією. Це особливо важливо для великих організацій, де кілька команд або відділів працюють над одними проектами [2].

CDA відповідає за публікацію цього контенту. Програма зберігає контент, завантажений у CMA в базі даних, і забезпечує динамічну доставку на різні платформи та пристрої. Не менш важлива роль CDA – оптимізація продуктивності вебсайту. CDA відповідає за налаштування політики кешування, балансування навантаження, обмеження доступу до певних ресурсів [2].

Разом ці компоненти утворюють ядро CMS, яке об'єднує процес управління контентом та його відображення.

1.2 Розвиток CMS

Перші вебсайти створювались у формі зроблених вручну статичних вебсторінок, побудованих на простих HTML файлах. За допомогою програми FTP вебсторінки копіювалися в директорії з файлами на працюючому вебсервері [3].

Технологічні інновації минулого створювали все більш візуальну та функціональну мережу. Невдовзі веббраузери почали підтримувати відображення зображень та CSS стилей, а запровадження Server Side Includes (SSI) дозволило відокремлювати частини сайту, наприклад верхній і нижній колонтитул, від основного вмісту. Однак створення та оновлення контенту вебсайтів все ще відбувалось здебільшого вручну. Такий підхід був неспроможним у довгостроковій перспективі [3].

Перші технології, схожі на CMS, полягали у використанні серверних сценаріїв для створення вмісту, який надсилається із сервера у веб-браузер. Багато організацій і компаній створили власні CMS на основі існуючих технологій. Характерно, що CMS тієї епохи часто адаптувалися до конкретних потреб організації. Однак кілька корпоративних CMS передбачали ринок уніфікованих і професійно розроблених систем управління контентом. Прикладами корпоративних CMS того періоду є GeoCities, FileNet, StoryBuilder, Interwoven, Documentum, FatWire, FutureTense та інші [3].

Відбулася дедалі більша професіоналізація та спеціалізація систем керування контентом. У цей період почав з'являтися термін «WCMS», що означає «система керування веб-контентом». У той час як CMS є найширшим терміном, що позначає систему керування вмістом для внутрішньої мережі, архівів і бізнес-операцій, система керування веб-контентом спрямована на Інтернет [3].

З'явилися системи керування вмістом із відкритим кодом, такі як Drupal, WordPress і Joomla. Більшість таких систем містили backend і frontend та могли обробляти тексти, зображення та інші файли для зберігання, відображення та завантаження [3].

Завдяки розвитку технологій, які дозволяли динамічно доставляти контент, вебсайти переходили від статичних сторінок до інтерактивних, виникла потреба в більш частому оновленні вмісту та управлінні ним. Таким

чином, однією з центральних ролей CMS стало надання можливостей для різних ролей користувачів і дозволів для доставки вмісту [3].

Крім того, технологічні інновації призвели до постійного збільшення інших функцій CMS, включаючи попередній перегляд, обробку URL-адрес, RSS-канали, адаптивний дизайн, коментарі відвідувачів, системи відстеження активності користувачів, системи дозволів, візуальні редактори, шаблони, інтеграцію з електронною комерцією тощо. Тенденції популярності різних функцій CMS зображено на рис. 1.1 [3].

2000	2005	2010	2015
Static content	WYSIWYG authoring	Social media integration	Customer intelligence
Separate content from design	Dynamic content	WYSIWYG page design	Context-aware
Animated GIFs	Publishing workflows	Collaboration tools	Multi-device
	User generated content	Rich media integration	Service-enabled / APIs
	Modular architecture	Lead generation tools	Multi-site platform governance
	Syndication		PaaS/SaaS
	Search		

Рисунок 1.1 – Тенденції популярності функцій CMS

Наступним кроком розвитку стало Digital Experience Platform (DXP) – це інтегрований набір основних технологій, які підтримують створення,

управління, доставку та оптимізацію персоналізованого цифрового контенту. DXP може бути окремим продуктом або пакетом програмного забезпечення з кількома інтегрованими програмами. DXP включає функції класичної CMS та інструменти роботи із аналітикою, електронною комерцією, машинним навчанням, персоналізацією, SEO та API для доставки контенту на різні пристрої [4].

1.3 Архітектура CMS

Архітектура CMS визначає взаємозв'язок між інструментами та технологіями, які керують функціоналом інтерфейсу та серверу. Побудова системи керування контентом визначає універсальність, ефективність, гнучкість та масштабовання сервісів вебсайту.

У традиційній архітектурі CMS, яку часто називають монолітною (рис. 1.2), frontend і backend тісно пов'язані та визначені заздалегідь. Такий варіант підходить для простих випадків використання, таких як персональні або бізнес вебсайти для невеликих компаній з обмеженими ресурсами та бюджетами. Впровадження просте оскільки сервісна та інтерфейсна частини поєднані і не потребують окремих налаштувань. А розробка, впровадження та оновлення, найдешевша в порівнянні з іншими видами архітектури CMS. Накладаються обмеження для розповсюдження на різних платформах, таких як мобільні пристрої та IoT. Розширення функціоналу може виявитись досить складним, оскільки традиційна архітектура CMS принципово не створена для модифікації або інтеграції з сторонніми сервісами. Відсутність гнучкості уповільнює цикл DevOps, а зв'язаність backend і frontend породжує значні ризики безпеки додатку [5].

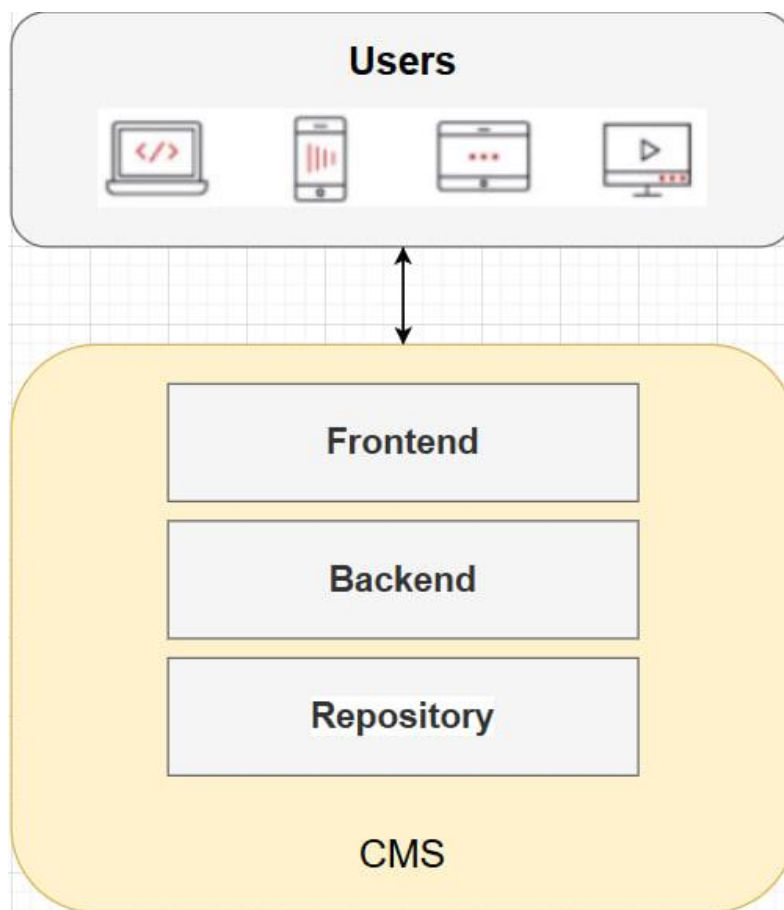


Рисунок 1.2 – Монолітна архітектура CMS

У роз'єднаній архітектурі CMS, що зображена на рис. 1.3, сервісна і інтерфейсна частина додатку є окремими системами. Backend керує створенням і зберіганням вмісту, тоді як frontend забезпечує представлення контенту користувачам через різні канали. Таким чином розробники можуть вносити зміни в сервісну частину без впливу або порушення роботи інтерфейсної. Окрім цього така архітектура менш вразлива до кібер атак. Збільшується швидкість публікації контенту, оскільки вміст надсилається безпосередньо до аудиторії, в обхід бек-енд серверу. Що в свою чергу підвищує SEO критерії, оскільки швидкість завантаження сторінки є фактором ранжування пошуковими системами. Впровадження дорожче та складніше, ніж традиційна архітектура CMS. Більше архітектурних рівнів означає додаткову роботу над розробкою, синхронізацією та вищу технічну експертизу працівників [5].

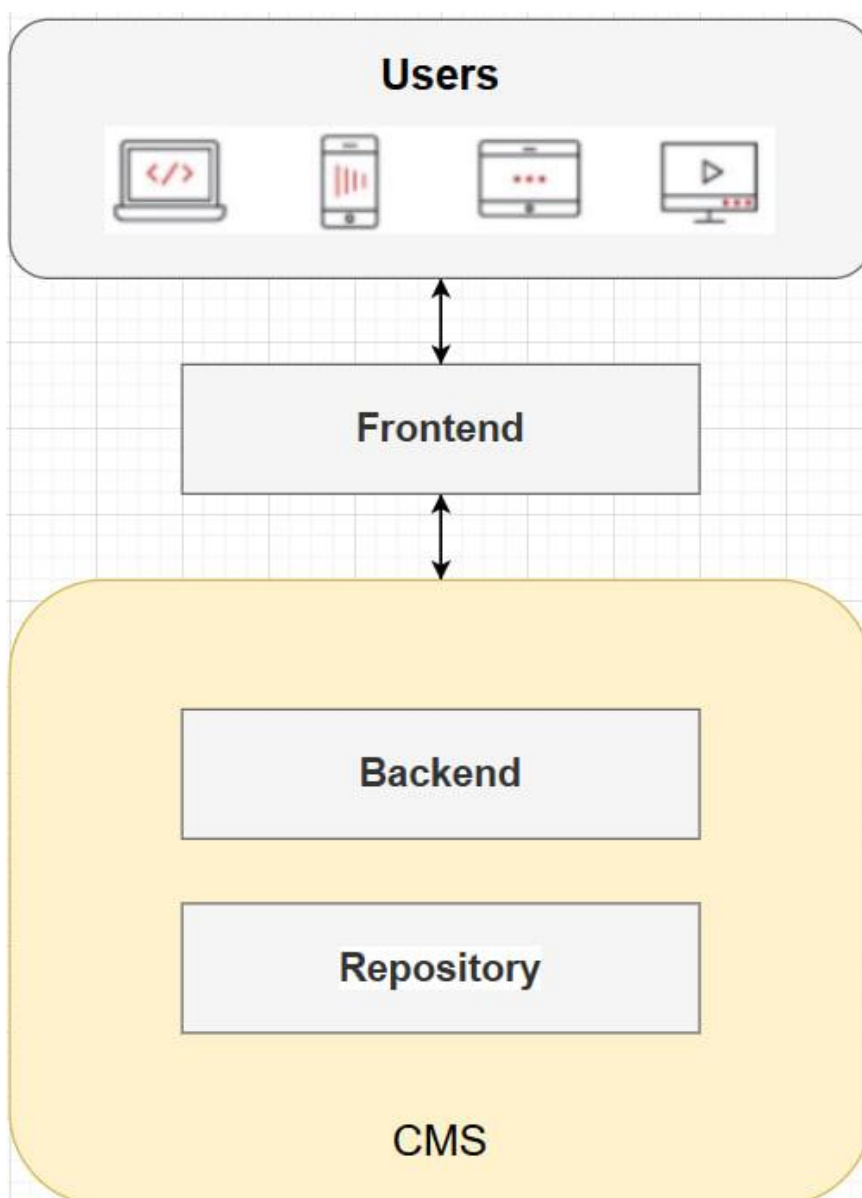


Рисунок 1.3 – Роз’єднана архітектура CMS

В безголовій архітектурі CMS, що зображена на рис. 1.4, немає попередньо визначеного frontend, який використовується для представлення контенту. Натомість backend передає вміст на різні пристрої та канали і дозволяє кожному з них керувати тим, як цей вміст має бути представлений. Завдяки безголовій архітектурі CMS розробники мають повну свободу та гнучкість вибору frontend технологій. Для маркетологів це дає змогу створювати багатоканальні кампанії – важливу можливість для корпоративного бізнесу. Така система є захищеною від атак, але натомість стає складною та дорогою в впровадженні та управлінні. Окрім цього неможливо запуснути деякі функції аналітики контенту, персоналізації, оскільки дані про взаємодію з

клієнтом не передаються між сервісною та інтерфейсною частинами у реальному часі [5].

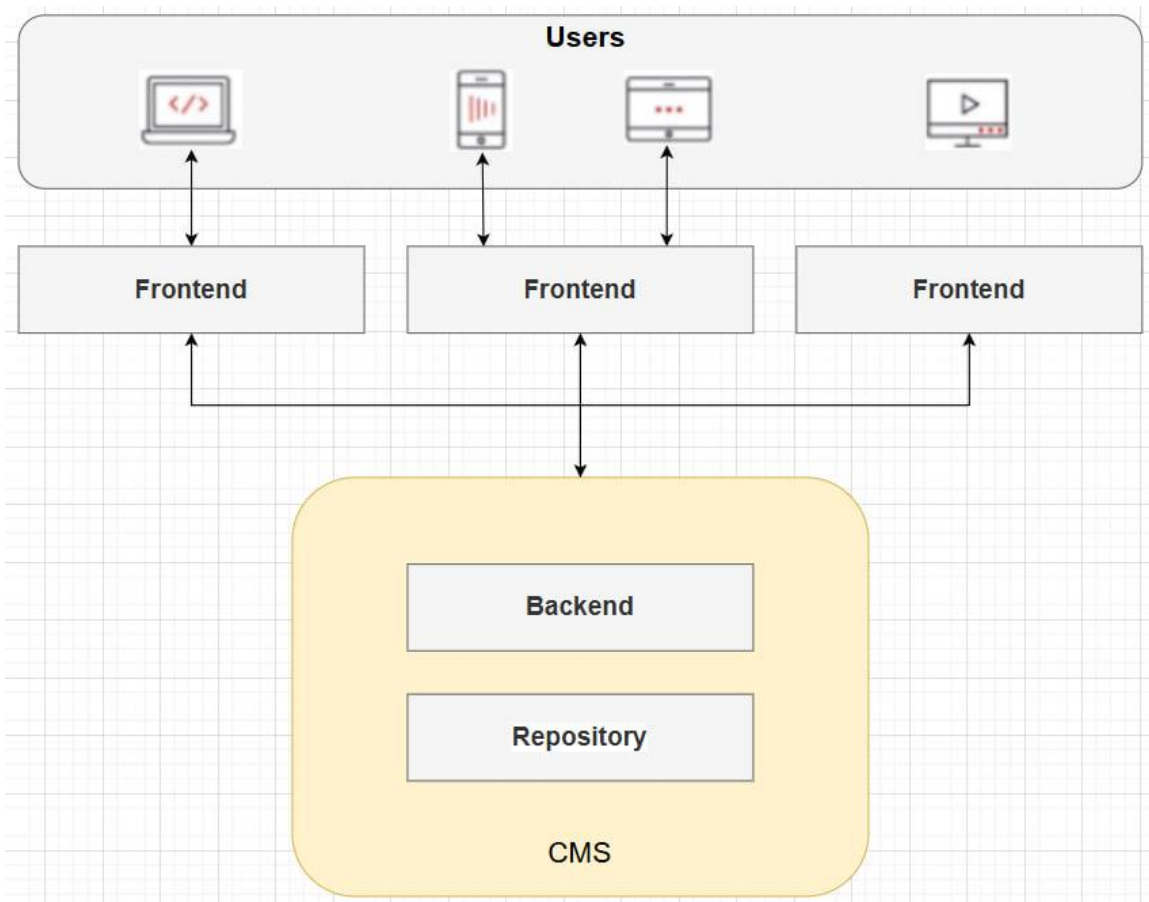


Рисунок 1.4 – Безголова архітектура CMS

Вибір CMS та відповідної архітектури є стратегічним рішенням, для цього організації повинні враховувати свої поточні та майбутні потреби, розмір бюджету, а також функціональні можливості та обмеження обраної платформи.

2 АНАЛІЗ СУЧАСНИХ CMS

2.1 Тенденції ринку CMS

Ринок систем управління контентом на сьогодні є дуже різноманітним, пропонуються рішення для різних потреб – від простих особистих сайтів до великих корпоративних порталів. Розвиток технологій та зміни в споживанні контенту призвели до того, що вибір CMS більше не обмежується кількома стандартними платформами. На сучасному ринку існує безліч варіантів, кожен з яких націлений на конкретні завдання та аудиторію.

На рис. 2.1 зображена популярність різних CMS протягом 2016 – 2023 років.

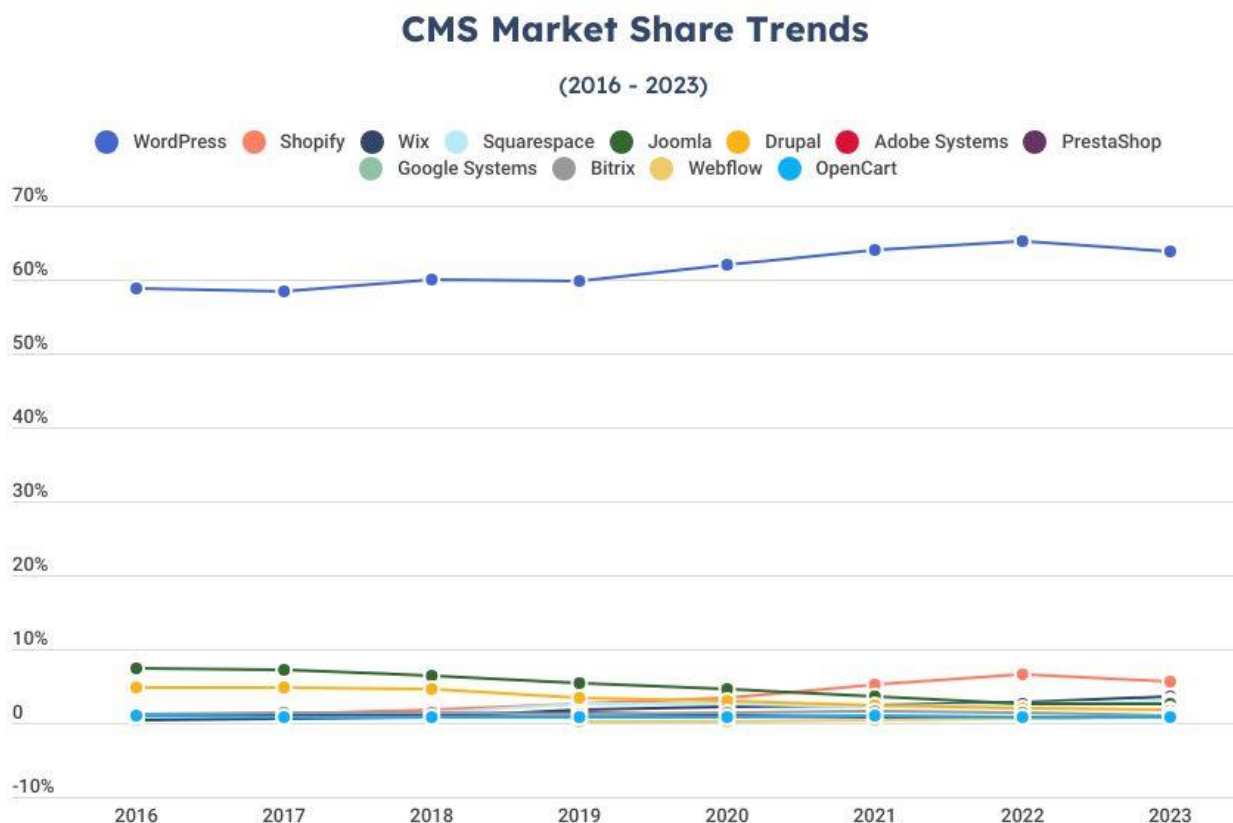


Рисунок 2.1 – Тенденції ринку CMS

Прості CMS, на кшталт Wix залишаються популярними завдяки своїй доступності та можливості створювати вебсайти без глибоких знань програмування. Вони підтримують широкий спектр плагінів, що дозволяє користувачам налаштувати зовнішній вигляд і функціонал сайту відповідно до своїх потреб. Проте ці платформи мають певні обмеження, коли йдеться про масштабування сайту або інтеграцію з більш складними бізнес-процесами.

Водночас зростає попит на хмарні CMS, які забезпечують більшу гнучкість, масштабованість та зручність для сучасних бізнесів. Такі системи дозволяють компаніям швидко розгорнути вебсайти, не витрачаючи ресурси на налаштування інфраструктури. Завдяки інтеграції з іншими сервісами, хмарні CMS забезпечують безперервність бізнес-процесів, полегшують управління контентом та є швидко адаптивними під потреби кінцевих користувачів.

Також популярні спеціалізовані e-commerce CMS. Вони пропонують інструменти для управління товарами, замовленнями, платежами та логістикою. Це особливо зручно для бізнесів, що спеціалізуються на онлайн-продажах.

Таким чином, ринок систем управління контентом представлений широким набором продуктів – від простих платформ до масштабованих хмарних і спеціалізованих систем. Вибір CMS залежить від специфіки завдань і вимог користувачів.

2.2 Система управління контентом WordPress

WordPress – це система управління контентом з відкритим вихідним кодом. Це означає, що це безкоштовне і гнучке програмне забезпечення, яке дозволяє керувати вмістом вебсайту, не потребуючи глибоких знань з програмування. На сьогодні WordPress найпопулярніша і найпоширеніша система управління контентом. Спочатку запущений у 2003 році як платформа для ведення блогів, WordPress значно еволюціонував. Сьогодні на ньому працюють понад 43% усіх веб-сайтів в Інтернеті, що робить його найпопулярнішою CMS у світі [6]. Вона відома своїм зручним інтерфейсом, широкою архітектурою плагінів та великою і активною спільнотою. WordPress написаний на PHP і використовує базу даних MySQL для зберігання контенту [6].

Створення вебсайту на WordPress передбачає вибір доменного імені, пошук хостинг-провайдера та встановлення WordPress. Після встановлення WordPress та створення акаунту користувач отримує доступ до інтерфейсу програми [7], зображеного на рис. 2.2.

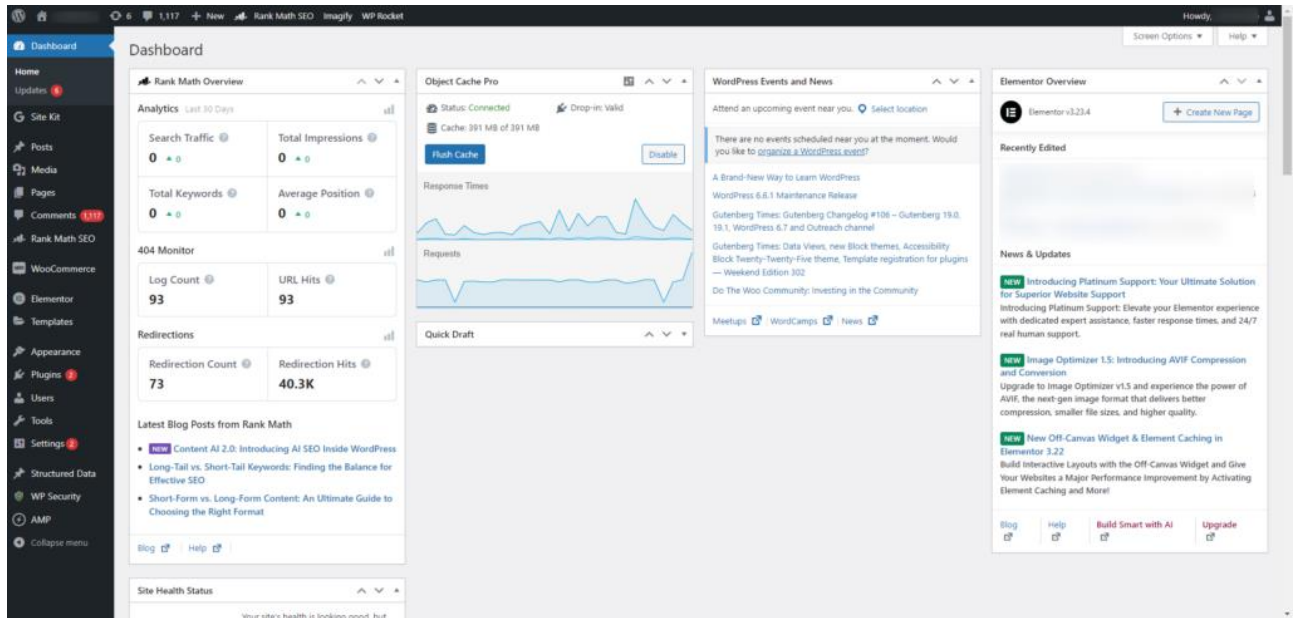


Рисунок 2.2 – Панель управління WordPress

Основні функції на панелі управління це створення сторінки та допису.

Функція створення сторінки дозволяє швидко створювати вебсторінки. WordPress має вбудований редактор, який можна використовувати для швидкого форматування тексту і мультимедійних ресурсів. Редактор контенту зображений на рис. 2.3.

Допис має такий самий редактор, як і розділ сторінка за винятком того, що в цій категорії контенту створюються дописи в блогах. Основна відмінність між сторінками полягає в обсязі. Щоб керувати великою кількістю публікацій в блозі WordPress має широкі можливості категоризації та управління ієрархією. Пости можна легко організувати за категоріями [7].

Пости і сторінки є вбудованими типами контенту але розробники можуть розширювати перелік за потреби бізнесу.

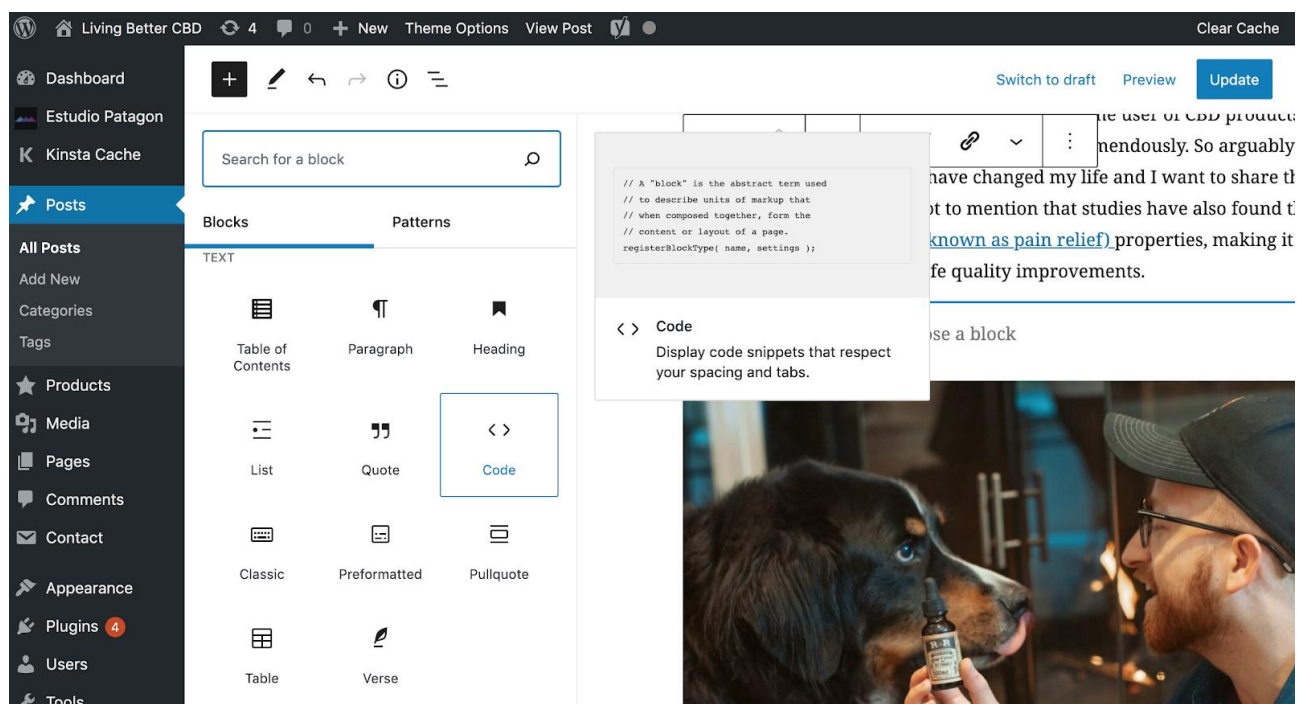


Рисунок 2.3 – Редактор контенту WordPress

Наступною ключовою особливістю WordPress є можливість налаштувати тему для веб сайту. Тема – це набір шаблонів і таблиць стилів, які визначають зовнішній вигляд і функціональність сайту. Існують тисячі доступних безкоштовних та платних тем з різноманітними дизайнами та опціями [7].

Для більшого розширення існуючої функціональності та відкриття нових можливостей використовуються плагіни. Плагіни дають змогу інтегрувати додаткові сервіси, покращити продуктивність або автоматизувати певні завдання без необхідності писати код. Також створювати власні плагіни для вирішення специфічних завдань бізнесу [7].

Отже хоча WordPress починався як простий інструмент для блогу, він виріс далеко за межі цих завдань [7]. WordPress є чудовим варіантом для малого та середнього бізнесу, особливо якщо це особистий блог чи магазин електронної комерції.

2.3 Система управління контентом Shopify

Shopify – це зручна система управління контентом, яка розвивається з 2004 року і розроблена спеціально для інтернет-магазинів. З його допомогою

величезна кількість компаній продають товари, в тому числі цифрові, по всьому світу. На даний момент це одне з найбільш затребуваних рішень для самостійного запуску проекту в сфері електронної комерції. Ключовою перевагою платформи Shopify є можливість запустити інтернет магазин не маючи досвіду програмування. Система дуже проста, але це не означає, що вона не має жодного необхідного підприємцю функціоналу [8].

Найвідомішою функцією системи Shopify є редактор візуальних тем. Використовуючи шаблони сторінок за допомогою редактора тем користувач може налаштувати дизайн сайту. Shopify пропонує різноманітний вибір тем, як безкоштовних так і платних. Також користувачі можуть самостійно створити шаблони використовуючи HTML, CSS, Liquid та завантажити на платформу.

Іншою важливою особливістю є власний платіжний процесор Shopify Payments, а також інтеграція з сторонніми платіжними шлюзами, наприклад PayPal, Stripe та Amazon Pay. Підтримка різноманітних платіжних шлюзів дозволяє оплату кредитними і дебетовими картками, за допомогою цифрових гаманців і навіть криптовалюти. Крім того, Shopify має вбудовані інструменти для управління мультивалютними транзакціями. Ця функція дозволяє магазинам відображати ціни в місцевій валюті та автоматично конвертувати їх на основі місцезнаходження клієнта.

Наступною функцією пов'язаною з продажами є POS. Shopify POS дозволяє компаніям продавати офлайн та онлайн в єдиній системі, допомагаючи синхронізувати товар, клієнтську базу та звітність.

Shopify має магазин застосунків, який пропонує тисячі інтеграцій зі сторонніми сервісами та плагінів для розширення основних можливостей платформи. За рахунок високої активності спільноти застосунки часто покращуються та адаптуються до нових технологій.

Також система має власні інструменти SEO та маркетингу. Вони дозволяють користувачам оптимізувати свої вебсторінки для пошукових систем, редагуючи метатеги, заголовки та URL. Окрім цього платформа підтримує інтеграцію з Google Analytics для аналізу поведінки користувачів і ефективності продажів.

Завдяки простій в використанні та розвиненій функціональності для ведення бізнесів пов'язаних з офлайн та онлайн продажами Shopify є надзвичайно популярним в сфері електронної комерції.

2.4 Система управління контентом Wix

Wix – це хмарна система управління контентом, що дозволяє легко створити функціональний вебсайт користувачу без знань дизайну та програмування. Як популярна і затребувана платформа вона пропонує сучасні та унікальні інструменти.

Wix пропонує використання штучного інтелекту для створення сайтів. Wix ADI створює повністю функціональний сайт, спираючись на відповіді користувача на кілька запитань про сайт. Він автоматично обирає стиль, макет, кольори, текстовий контент та зображення, що відповідають зазначеним цілям. Цей підхід дозволяє швидко отримати базову версію сайту, яку в подальшому легко доналаштовувати. Також ШІ може допомогти в створенні текстів, дизайні логотипів, генерації відео, редагуванні фото, та навіть створенні чат боту [9].

Якщо немає чіткого бачення майбутнього сайту можна скористатись готовими шаблонами. Wix пропонує придбати сотні шаблонів для різних галузей та цілей. Платформа також дозволяє створити та завантажити свій шаблон за допомогою Wix Studio [9].

Після вибору шаблону користувач може адаптувати його за допомогою Wix Editor. Цей інструмент підтримує технологію drag-and-drop, дозволяючи легко додати та налаштувати будь які типи контенту [9].

В Wix Editor інтегрований Wix Video Maker. Wix Video Maker – безкоштовний онлайн-редактор відео, створений у партнерстві з Vimeo. Цей інструмент підходить для створення та редагування відео для соціальних мереж, презентацій продуктів, оголошення акцій чи підсумків подій [9].

Звісно схожий функціонал доступний і для фото. Wix Photo Studio дозволяє редагувати зображення безпосередньо на платформі. Можливості включають обрізку, зміну розміру, регулювання яскравості та контрасту, застосування фільтрів, видалення фону, додавання тексту тощо. Також в редактор вбудована інтеграція з Photoshop для забезпечення більш складних редагувань [9].

Окрім цього є цікава можливість найняти фрілансера для допомоги з розробкою вебсайту через платформу Wix Marketplace. Спеціалісти можуть

допомогти з окремим завданням наприклад розробка дизайну, налаштування SEO чи створити вебсайт повністю відповідно наданих вимог [9].

Wix добре адаптована для розробки рішень з електронної комерції. Платформа має свій платіжний інструмент Wix Payments. Це рішення приймає всі види платежів в Інтернеті, включаючи функцію розсрочки та оплату в кредит. Користувач можете налаштувати одноразові платежі за покупку товару або періодичні платежі за підписку. Для США та Канаді, Wix Payments інтегроване з POS Wix. Ця інтеграція допоможе синхронізувати керування каталогом, транзакціями, аналітикою та платежами для онлайн і офлайн магазинів [9].

SEO інструменти від Wix платформи відрізняються від конкурентів використанням інтеграції Wix ADI та Google Search Console. Такий дует не тільки допоможе швидко налаштувати метадані, URL адреси та теги, але й сгенерує звітність моніторингу ефективності веб сайту.

3 ДОСЛІДЖЕННЯ СИСТЕМИ АЕМ

3.1 Технологічний стек

Система управління контентом АЕМ є надійною платформою, побудованою на перевірених, масштабованих та гнучких технологіях.

Він написаний на основі Java, це надає перевагами надійності, простоти використання, об'єктно-орієнтованих концепцій програмування та широкої підтримки спільноти [10].

Для обробки HTTP запитів використовується REST фреймворк Apache Sling. Sling співвідносить URL адреси HTTP запитів із ресурсами контенту на основі шляху, розширення та селекторів запиту. Далі запити обробляються скриптами та сервлетами, які динамічно вибираються залежно від поточного ресурсу.

Щоб забезпечити модульність, АЕМ користується Apache Felix фреймворком, що є реалізацією інфраструктури OSGi. Це модульна система, що дозволяє АЕМ функціонувати як набір невеликих багаторазових компонентів, що знаходяться в OSGi контейнері [10]. Кожен контейнер можна запускати, зупиняти або налаштовувати незалежно.

В якості репозиторію використовується CRX. CRX – це реалізація об'єктної бази JCR від Adobe. Вона забезпечує структурований доступ до забережених даних та додає функції повнотекстового пошуку, керування версіями та транзакції [10].

Для спрощення інтеграції серверної логіки з дизайном, використовується HTL. Він був розроблений компанією Adobe як заміна JSP, пропонуючи більш безпечну, ефективну та зручну для розробників альтернативу для створення веб-сторінок.

У ролі інструмента для управління навантаженням та кешування використовується Dispatcher. Це допомагає покращити продуктивність системи, а також захищає сервер від атак [10].

Кожна з цих технологій є невід'ємною частиною платформи АЕМ, що забезпечує стабільну роботу та адаптивність до різноманітних потреб.

3.2 Інфраструктура АЕМ

Інфраструктура АЕМ складається з декількох ключових компонентів, які працюють узгоджено для забезпечення високої продуктивності та зручного керування ресурсами. Серед них основними є Author, Publish, Dispatcher.

Author екземпляри розміщується на віддаленому сервері і працюють через налаштований порт. Віддалений сервер налаштований за внутрішнім брандмауером і зазвичай вимагає доступу через VPN. Екземпляр автора є середовищем, де користувачі створюють чи редагують сторінки, завантажують медіаресурси, налаштовують компоненти та виконують будь-які інші дії, які не повинні бачити кінцеві користувачі веб-сайту [11].

Publish екземпляр – місце, де знаходиться опублікований контент. Ресурси на екземплярі публікацій загальнодоступні. [11].

Для перенесення контенту використовуються агенти реплікації. Агенти реплікації налаштовуються за допомогою URL адреси та облікових даних до Publish екземплярів. Екземпляр автора надсилає контент для публікації на URL-адресу, а екземпляр публікації отримує вміст і відповідно оновлює чи створює ресурси [11].

Dispatcher – це модуль Apache, встановлений на сервері, який дозволяє кешувати певні ресурси. Кешування прискорює доставку контенту кінцевому користувачу. Правила кешування налаштовуються в спеціальних конфігураціях диспетчера на сервері, зображених на рис. 3.1. Також за допомогою цих конфігурацій можна заборонити доступ до певних сторінок, перенаправити користувача на інші ресурси якщо це необхідно з міркувань безпеки або якщо на цих сторінках міститься персоналізований контент. Dispatcher також виступає в ролі балансувальника навантаження для різних Publish екземплярів [11].

```

# The load will be balanced among these render instances
/renderers
{
  /rend01
  {
    # Hostname or IP of the render
    /hostname "pub01.core.com"
    # Port of the render
    /port "4503"
    # Connect timeout in milliseconds, 0 to wait indefinitely
    # /timeout "0"
  }
  /rend02
  {
    /hostname "pub02.core.com"
    /port "4503"
  }
  /rend03
  {
    /hostname "pub03.core.com"
    /port "4503"
  }
}

```

Рисунок 3.1 – Файл конфігурації Dispatcher

Кількість різних екземплярів АЕМ може відрізнятися залежно від потреб та розміру сайту. На рис. 3.2 показано один з можливих варіантів. Кожен з екземплярів відіграє важливу роль у контролі та доступі до контенту, створеного користувачами.

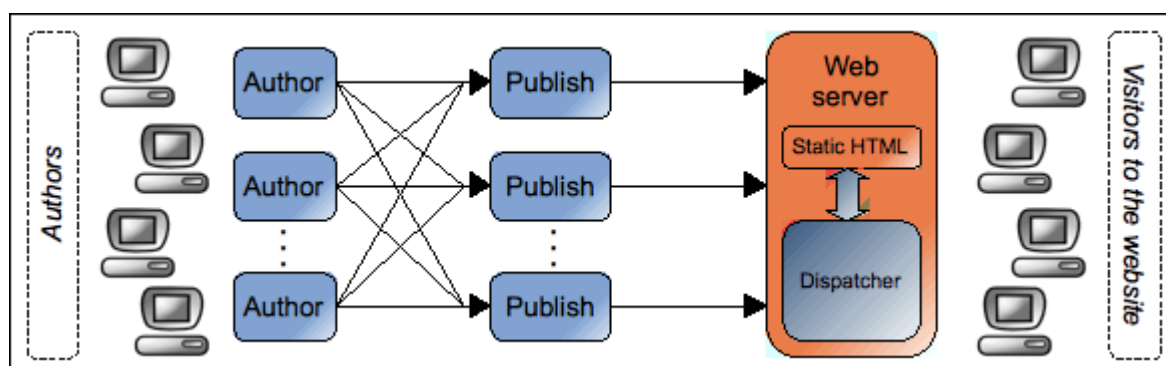


Рисунок 3.2 – Схема конфігурації АЕМ інфраструктури

3.3 Створення та управління контентом

Для створення контенту в АЕМ найчастіше використовуються компоненти. Компоненти слугують будівельними блоками для створення сторінок. Кожен компонент призначений для виконання певної функції, починаючи від базового відображення тексту та зображень і закінчуючи більш складними функціями, такими як динамічні каруселі, навігаційні меню або інтерактивні форми. Ці компоненти можна використовувати багаторазово і конфігурувати, що надає авторам значну гнучкість. Наприклад, при створенні допису в блозі автор може вибрати компонент заголовка, текстовий компонент для основного контенту і компонент зображення для додавання відповідних візуальних ефектів. Окрім цього компоненти підтримують інтерфейс WYSIWYG. Це означає що створення, редагування блоків на сторінці не потребує оновлення сторінки [12].

Розробники мають можливість розширювати бібліотеку вбудованих компонентів, створюючи свої на основі існуючих або з нуля, пристосовані до конкретних потреб бізнесу, що дозволяє створювати унікальні елементи дизайну та взаємодії з клієнтом, зберігаючи при цьому модульну структуру системи. Розширення відбувається за допомогою додавання XML файлів що описують функції взаємодії з компонентом, Java та JavaScript файлів, що описують функціональні особливості та HTML, CSS файлів, які відповідають за дизайн компонента.

Трапляються випадки коли постійно використовується блок з певних компонентів, для уникнення повторення такі компоненти можна з'єднати в Experience Fragment. Перевага фрагментів полягає в тому, що вони дозволяють керувати структурованим контентом без обмежень, пов'язаних з дизайном. На відміну від компонентів, які прив'язані до макета сторінки, фрагменти контенту складаються з багаторазового редакційного вмісту, який можна динамічно вставляти в різні контексти. Ця функція особливо корисна для організацій, які працюють з контентом у великих масштабах на різних платформах. Наприклад, компанія, що займається електронною комерцією, може розробити фрагмент контенту для опису продукту, включивши в нього поля для назв продуктів, специфікацій, цін і рекламної інформації. Посилаючись на цей фрагмент на сторінках з детальним описом продукту, в каталогах і рекламних банерах,

компанія може підтримувати узгодженість контенту, одночасно спрощуючи його оновлення. Якщо необхідно скоригувати ціну, зміну можна внести один раз в межах фрагмента, і вона автоматично відобразиться в усіх випадках, де він використовується.

Для ефективного управління контентом також використовують шаблони сторінок. Вони визначають дозволені чи заборонені компоненти, дозволяють попередньо визначити розташування контенту. Це допомагає підтримувати узгодженість дизайну, дозволяючи при цьому кастомізувати вміст сторінки у встановлених межах.

Інтеграція компонентів з DAM дозволяє працювати з мультимедійними ресурсами. Система управління цифровими ресурсами надає можливість користувачам зберігати, керувати та обмінюватися зображеннями, відео, аудіофайлами, документами та іншими ресурсами [12]. Для зберігання використовується централізоване сховище. Таке рішення гарантує, що користувачі працюють з найактуальнішими версіями ресурсів.

DAM має потужні пошукові можливості, які дозволяють швидко знаходити ресурси за допомогою метаданих, смарт-тегів та ключових слів. Підключення Adobe Sensei розширює ці пошукові функції, автоматично генеруючи смарт-теги за допомогою машинного навчання.

Завдяки інструментам редагування DAM спрощується підготовка контенту для різних цифрових каналів. Користувачі можуть обрізати, змінювати розмір та застосувувати фільтри для фото і відео матеріалів безпосередньо в платформі. Також доступна інтеграція з застосунками Adobe Creative Cloud, такими як Photoshop та InDesign.

3.4 Керування вебсайтом

Наступний рівень абстракції після компонентів для керування контентом є сторінки та мікросайти. MSM дозволяє налаштувати зв'язки між сторінками та сайтами таким чином щоб змінений вміст ресурсу автоматично переносився на зв'язані ресурси. Це дозволяє швидко і легко підтримувати багато сайтів без необхідності мануального втручання.

Для створення зв'язку користувач повинен обрати Blueprint. Blueprint діє як головна версія контенту, з якої інші сайти чи сторінки беруть свою

структуру та вміст. Будь-які зміни, внесені в Blueprint, можуть бути поширені на пов'язані з ним ресурси за допомогою визначених механізмів синхронізації.

Версія оригінального контенту, створена на основі Blueprint називається Live Copy. Її можна певною мірою налаштувати, але основне призначення полягає в тому, щоб успадковувати контент і структуру від шаблону. Ступінь успадкування та кастомізації регулюється політикою синхронізації, що забезпечує баланс адаптивності [13].

Наступний важливий елемент MSM це Language Copy. Вона підтримує багатомовне управління сайтом, дозволяючи створювати версії контенту різними мовами. Language Copy відрізняється від Live Copy тим, що зосереджується на перекладі та локалізації контенту, а не на успадкуванні загальних оновлень з Blueprint. Мовні копії спрощують процеси перекладу, зберігаючи при цьому чіткий зв'язок зі структурою оригінального контенту [13].

АЕМ має потужні перекладацькі інструменти, зокрема інтеграцію зі службами машинного перекладу, словниками перекладів і можливостями перекладення вручну. Ці інструменти спрощують процес створення багатомовного контенту для різних мовних і культурних аудиторій.

Процес передачі оновленого контенту з Blueprint до пов'язаних з ним Live Copy або Language Copy називається Rollout. Конфігурації розгортання визначають, як і коли оновлення надсилатимуться до залежних ресурсів. АЕМ пропонує попередньо визначені конфігурації розгортання, такі як «Push on Modify» або «Activate on Blueprint activation», які визначають час і спосіб поширення оновлень. Однак розробники мають можливість створювати власні конфігурації розгортання. Такі конфігурації можуть встановлювати детальні правила синхронізації, тригери та дії. Адаптивність дозволяє створити унікальні стратегії управління контентом, що водночас відповідають глобальній узгодженості, та локальним вимогам кастомізації.

3.5 Автоматизація управління контентом

Workflow в АЕМ є інструментом автоматизації процесів створення, затвердження, перегляду та публікації контенту. Впроваджуючи робочі процеси значно підвищується ефективність управління контентом.

В основі роботи інструменту Workflow лежать моделі робочих процесів. Моделі містять послідовність і логіку кроків, необхідних для завершення робочого процесу. За замовчуванням користувачам надається кілька готових моделей робочих процесів з простою логікою затвердження чи публікації сторінок. Розробники можуть створювати власні моделі для задоволення користувацьких потреб. Конфігурація моделей робочих процесів здійснюється за допомогою візуального редактора, який дозволяє легко налаштовувати кроки та умови за допомогою перетягування відповідних блоків в робочу область редактора.

Види кроків в моделі Workflow заздалегіть визначені.

Process Step – на цьому кроці під час виконання запускається програмний код або ЕСМА-скрипт. Він використовується, коли певна логіка має бути виконана програмно. Наприклад, зміна розміру зображення або генерація метаданих [14].

Participant Step – цей крок призначає завдання певному користувачеві або групі. Він використовується для завдань, які вимагають участі людини, наприклад, перегляд або затвердження контенту. Призначений учасник отримує сповіщення і робочий процес чекає виконання цього завдання [14].

Dynamic Participant Step – на цьому кроці завдання для виконання користувачу призначається динамічно за допомогою програмного коду або ЕСМА-скрипту. Він пропонує більшу гнучкість альтернативу Participant Step, спрямовуючи завдання користувачам на основі програмних умов або значень метаданих [14].

Decision Step – цей крок відповідає за розгалуження дій в робочому процесі. Він підтримує операції І та АБО для керування подальшими діями робочого процесу. На цьому кроці завдання затвердження контенту може бути направлено різним менеджерам залежно від типу контенту або пріоритету [14].

Workflow може бути запущено вручну або за допомогою Workflow Launcher. Цей інструмент відстежує зміни контенту в репозиторії та запускають відповідний робочий процес без будь-якого ручного введення. Наприклад, лаунчер можна налаштувати так, щоб він запускав робочий процес затвердження контенту щоразу, коли новий ресурс завантажується до певної папки.

Отже, автоматизація процесів управління контентом зменшує витрати на ручну роботу, захищає від помилок та забезпечує своєчасне виконання бізнес задач.

3.6 Політики формування вартості

АЕМ працює переважно на основі моделі передплати, і вартість цих ліцензій може значно відрізнятись залежно від конкретних потреб організації [16].

Як платформа корпоративного рівня, вона починається від 100000 доларів США за ліцензію [15]. Немає безкоштовної чи пробної версії, але після купівлі надається доступ до online підтримки.

Фактори які впливають на загальну вартість ліцензії:

- кількість користувачів, які матимуть доступ до системи,
- обсяг цифрового контенту в системі,
- конкретні функції чи модулі.

Підприємства повинні бути готові до широкого діапазону ліцензійних платежів, які зазвичай коливаються від десятків тисяч до сотень тисяч доларів на рік. Для АЕМ, як хмарної служби (АЕМaaS), витрати можуть бути особливо високими, понад 5000 доларів США на користувача системою в місяць [16].

Окрім ліцензування, впровадження АЕМ також пов'язане витратами. Вони включають витрати пов'язані з консультаціями та плануванням, інтеграцією системи, міграцією даних із існуючих платформ, а також початковою конфігурацією та налаштуванням системи. [16]

Також треба зважати, що розробка та підтримка роботи платформи, інтеграція АЕМ зі сторонніми системами, такими як CRM або ERP-платформи збільшують загальну вартість [16].

До того ж організаціям можуть бути потрібні витрати на навчання команди ефективному використанню АЕМ. Adobe пропонує ряд навчальних програм і сертифікаційних курсів, адаптованих до різних ролей користувачів і рівнів знань. Хоча ці програми є безцінними для максимізації ваших інвестицій, але це також збільшить витрати [16].

Таким чином, вартість використання АЕМ не обмежується лише ліцензуванням. Проте інвестиції в АЕМ виправдані для великих організацій, де необхідна централізована система управління контентом із широкими можливостями персоналізації, масштабованості та інтеграції з різноманітними бізнес-платформами.

4 ПРОЄКТ ВДОСКОНАЛЕННЯ СИСТЕМИ АЕМ ДЛЯ РОЗРОБКИ ВЕБСАЙТІВ ТЕЛЕКОМУНІКАЦІЙНИХ КОМПАНІЙ

4.1 Структура проєкту

Початок роботи з Adobe Experience Manager передбачає створення і налаштування екземпляра АЕМ і комерційного проєкту в рамках якого буде відбуватись модифікація платформи.

Для створення екземпляру АЕМ спершу треба купити підписку на сайті Adobe. Після цього користувачу буде надано доступ до особистого профілю з ліцензією та jar файлом з CMS застосунком. Далі необхідно встановити та налаштувати застосунок згідно інструкцій від Adobe. Стандартне налаштування для професійного використання включає створення author, publish та підключення dispatcher. Для використання АЕМ в рамках дипломної роботи достатньо встановлення author екземпляра.

Структура модулів в АЕМ проєктах ґрунтується на відповідності структури JCR. Отже створення проєкту можливо вручну чітко дотримуючись структури або за допомогою скрипту для генерації. Скрипт розроблений компанією Adobe та оновлюється відповідно до версії та архитипу АЕМ. Скрипт зображений на рис. 4.1.

```
mvn archetype:generate \  
  -DarchetypeGroupId=com.adobe.granite.archetypes \  
  -DarchetypeArtifactId=aem-project-archetype \  
  -DarchetypeVersion=latest \  
  -DgroupId=com.yourcompany.project \  
  -DartifactId=my-aem-project \  
  -Dversion=1.0-SNAPSHOT \  
  -Dpackage=com.yourcompany.project
```

Рисунок 4.1 – Скрипт для генерації АЕМ проєкта

Проєкт містить наступну структуру:

- all – модуль необхідний для зборки єдиного файлу та розгортання його в хмарній інфраструктурі;
- core – модуль, що зберігає програмний код, необхідний для впровадження нової функціональності;
- dispatcher – модуль, який зберігає всі конфігурації пов'язані з налаштуванням dispatcher екземпляра;
- it.test – модуль, що зберігає код для інтеграційного тестування;
- ui.apps – модуль, який зберігає xml структуру АЕМ компонентів;
- ui.apps.structure – модуль для зберігання і управління конфігураціями компонентів;
- ui.config – модуль, що необхідний для зберігання конфігурацій програмного коду;
- ui.content – модуль, що зберігає статичний контент сайту;
- ui.frontend – модуль для фронтенд інтеграції;
- ui.tests – модуль, що зберігає код для тестування.

Структура проєкту зображена на рис. 4.2.

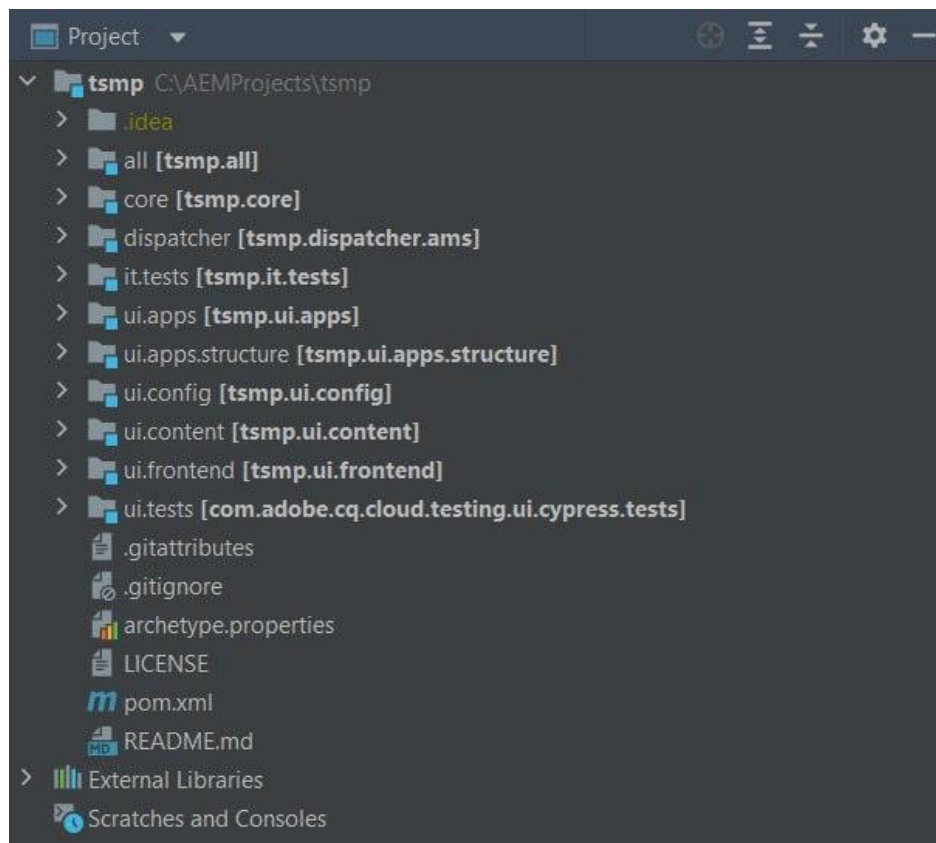


Рисунок 4.2 – Структура проєкту

В якості інструмента для управління залежностями та збірки проекту в дипломній роботі використан Maven. Maven – це інструмент автоматизації збірки та управління проектами з відкритим вихідним кодом. Він автоматизує компіляцію вихідного коду та керування залежностями, збирає двійкові коди в пакети та виконує тестові сценарії. Maven створений на основі концепції POM [17]. Об'єктна модель проекту це тип XML-файлу, який містить дані з вашого проекту та деталі конфігурації.

Maven конфігурації дуже важливі для правильного розгортання проекту в АЕМ середовищі. В ньому описані всі модулі проекту, технічні деталі, нахшталт версій та інструментів платформи, плагіни необхідні для збірки проекту, дані доступу до платформи. Основний файл конфігурації показана на рис. 4.3.

```

38
39 <properties>
40   <aem.host>localhost</aem.host>
41   <aem.port>4502</aem.port>
42   <aem.publish.host>localhost</aem.publish.host>
43   <aem.publish.port>4503</aem.publish.port>
44   <sling.user>admin</sling.user>
45   <sling.password>admin</sling.password>
46   <vault.user>admin</vault.user>
47   <vault.password>admin</vault.password>
48   <frontend-maven-plugin.version>1.12.0</frontend-maven-plugin.version>
49   <core.wcm.components.version>2.23.4</core.wcm.components.version>
50   <java.version>11</java.version>
51   <bnd.version>5.1.2</bnd.version>
52   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
53   <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
54   <componentGroupName>TSMP</componentGroupName>
55 </properties>
56
57 <build>
58   <plugins>
59     <!-- Maven Release Plugin -->
60     <plugin>
61       <groupId>org.apache.maven.plugins</groupId>
62       <artifactId>maven-release-plugin</artifactId>
63       <version>2.5.3</version>
64       <configuration>
65         <scmCommentPrefix>[maven-scm] :</scmCommentPrefix>
66         <preparationGoals>clean install</preparationGoals>
67         <goals>install</goals>
68         <releaseProfiles>release</releaseProfiles>
69       </configuration>
70     </plugin>
71     <!-- Maven Source Plugin -->
72     <plugin>
73       <groupId>org.apache.maven.plugins</groupId>
74       <artifactId>maven-source-plugin</artifactId>
75       <version>3.0.1</version>
76       <inherited>true</inherited>
77     </plugin>

```

Рисунок 4.3 – Основна maven конфігурація

4.2 Архітектура проєкту

Основне завдання перед архітектурою проєкту по модифікації платформи АЕМ для телекомунікаційних компаній полягає в створенні легко масштабованої, ефективною та зручної системи для розробників і адміністраторів.

Більшість коду створено з використанням Java Generic підходу.

Це спеціальний підхід до опису даних та алгоритмів, що дозволяє працювати з різними типами даних, не змінюючи код для їх використання. Окрім цього весь код навмисно було зосереджено в одному OSGI контейнері для подальшої збірки в єдиний jar файл. Це необхідно для забезпечення портативності. Також всі необхідні залежності це проєкти з відкритим кодом. Ціль такого підходу до побудови архітектури це створення легкого для інтеграції та корисного для телекомунікаційних організацій застосунку.

Зазвичай при описанні архітектури розділяють Frontend (Презентаційний рівень) та Backend (Функціональний) рівень.

Презентаційни рівень створеного застосунка представляє собою інструменти АЕМ платформи та XML, HTML, HTL, CSS файли спеціалізованих компонентів. Цей рівень відповідає за відображення контенту та забезпечення інтерактивної взаємодії з користувачами. Його завдання – забезпечити зручний, швидкий і адаптивний інтерфейс. На рис. 4.4 показано необхідні файли для відображення компонента Tariff Comparator. Структура є аналогічною для всіх доданих компонентів.

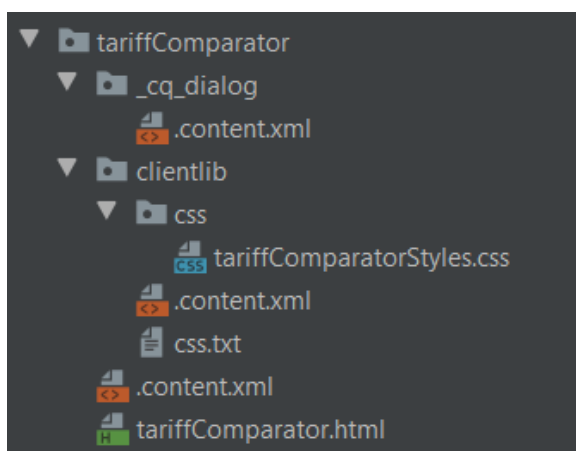
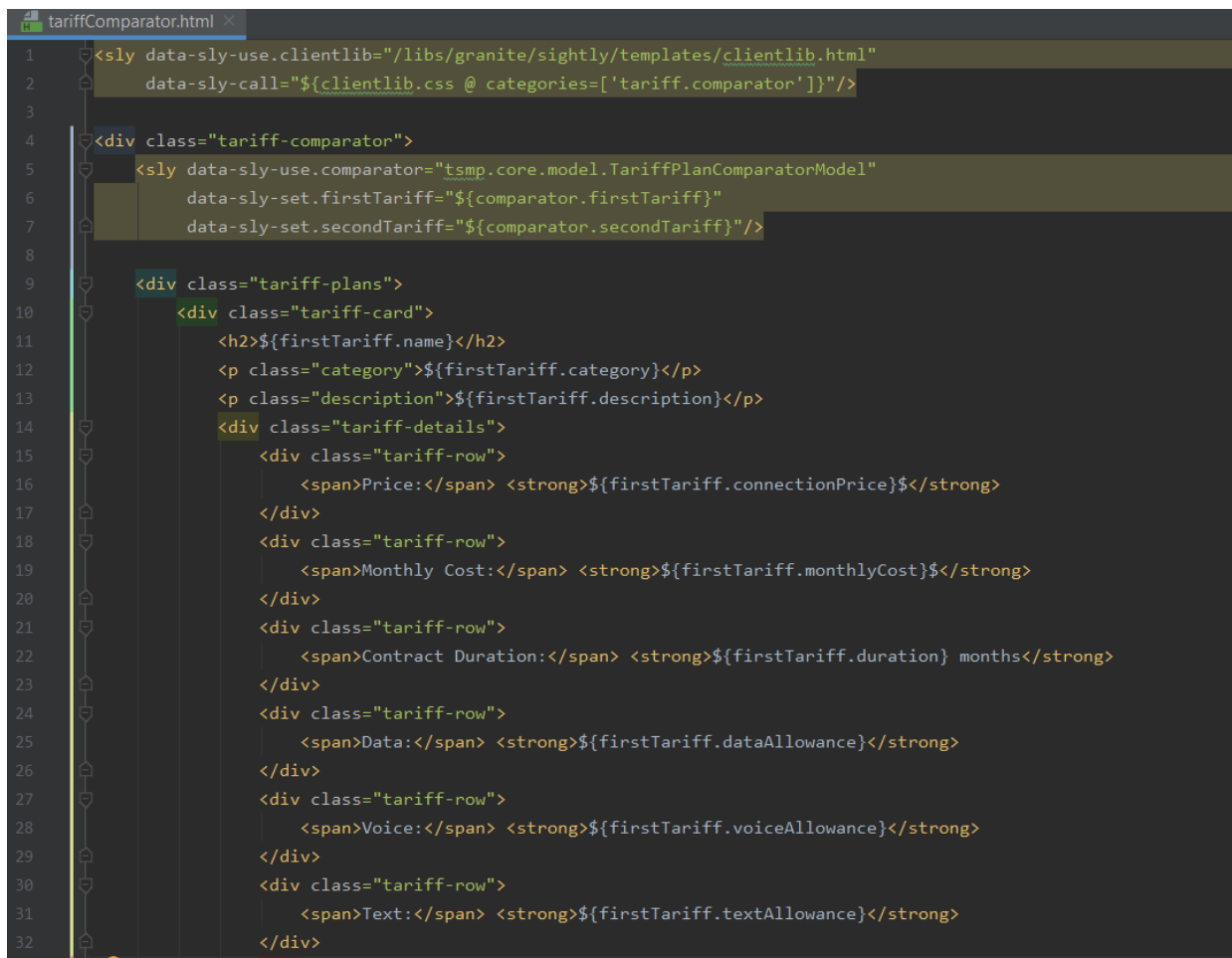


Рисунок 4.4 – Структура файлів для відображення компонента Tariff Comparator

Характерна особливість презентаційного рівня написаного на АЕМ, без використання сторонніх фреймворків – відсутність HTTP запитів між Frontend та Backend. Натомість взаємодія виконується за рахунок використання референсів на Sling Model з HTML за допомогою HTL інструментів. Sling Model – це частина функціонального рівня, Java-класи, які використовуються для виконання логіки компонентів. HTML файл та Sling Model для компонента Tariff Comparator зображені на рис. 4.5 та рис. 4.6 відповідно.



```
1 <sly data-sly-use.clientlib="/libs/granite/sightly/templates/clientlib.html"
2 data-sly-call="${clientlib.css @ categories=['tariff.comparator']}"/>
3
4 <div class="tariff-comparator">
5 <sly data-sly-use.comparator="tsmp.core.model.TariffPlanComparatorModel"
6 data-sly-set.firstTariff="${comparator.firstTariff}"
7 data-sly-set.secondTariff="${comparator.secondTariff}"/>
8
9 <div class="tariff-plans">
10 <div class="tariff-card">
11 <h2>${firstTariff.name}</h2>
12 <p class="category">${firstTariff.category}</p>
13 <p class="description">${firstTariff.description}</p>
14 <div class="tariff-details">
15 <div class="tariff-row">
16 <span>Price:</span> <strong>${firstTariff.connectionPrice}</strong>
17 </div>
18 <div class="tariff-row">
19 <span>Monthly Cost:</span> <strong>${firstTariff.monthlyCost}</strong>
20 </div>
21 <div class="tariff-row">
22 <span>Contract Duration:</span> <strong>${firstTariff.duration} months</strong>
23 </div>
24 <div class="tariff-row">
25 <span>Data:</span> <strong>${firstTariff.dataAllowance}</strong>
26 </div>
27 <div class="tariff-row">
28 <span>Voice:</span> <strong>${firstTariff.voiceAllowance}</strong>
29 </div>
30 <div class="tariff-row">
31 <span>Text:</span> <strong>${firstTariff.textAllowance}</strong>
32 </div>
```

Рисунок 4.5 – HTML файл компонента Tariff Comparator

```

@Model(adaptables = Resource.class, defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL)
public class TariffPlanComparatorModel {

    @Inject
    private OffersHolderService offersHolderService;

    @Optional
    @ValueMapValue
    private String firstTariffId;

    @Optional
    @ValueMapValue
    private String secondTariffId;

    private TariffPlanDto firstTariff;

    private TariffPlanDto secondTariff;

    @PostConstruct
    protected void init() {
        List<TariffPlanDto> tariffPlans = offersHolderService.getTariffPlans();

        firstTariff = tariffPlans.stream()
            .filter(tariffPlan -> firstTariffId.equals(tariffPlan.getId()))
            .findAny().orElse( other: null);

        secondTariff = tariffPlans.stream()
            .filter(tariffPlan -> secondTariffId.equals(tariffPlan.getId()))
            .findAny().orElse( other: null);
    }
}

```

Рисунок 4.6 – Sling Model компонента Tariff Comparator

Разом з тим використання HTTP запитів не заборонено та може допомогти для динамічного відображення даних в діалогових вікнах компонентів. Наприклад Tariff Comparator компонент має XML конфігурацію, що при відкритті діалогового вікна надсилає HTTP запит на отримання необхідних для відображення даних. Обробкою HTTP запитів займається Servlet, зображений на рис. 4.7. Servlet – це також частина функціонального рівня, Java-класи, що забезпечують взаємодію між двома рівнями за рахунок обміну HTTP запитами.

```

@Component(
    service = Servlet.class,
    property = {
        Constants.SERVICE_DESCRIPTION + "= Dynamically Populate Tariff Dropdown Servlet",
        "sling.servlet.resourceTypes=" + "/apps/tariffDropDown"
    })
public class DynamicallyPopulateTariffDropDownServlet extends SlingSafeMethodsServlet {

    private static final String VALUE = "value";
    private static final String TEXT = "text";

    @Reference
    private OffersHolderService offersHolderService;

    @Override
    protected void doGet(SlingHttpServletRequest request, SlingHttpServletResponse response) {
        ResourceResolver resourceResolver = request.getResourceResolver();
        List<TariffPlanDto> tariffPlanDtos = offersHolderService.getTariffPlans();

        DataSource ds =
            new SimpleDataSource(
                new TransformIterator(
                    tariffPlanDtos.iterator(),
                    input -> {
                        TariffPlanDto model = (TariffPlanDto) input;
                        ValueMap vm = new ValueMapDecorator(new HashMap<>());
                        vm.put(VALUE, model.getId());
                        vm.put(TEXT, model.getName());
                        return new ValueMapResource(
                            resourceResolver, new ResourceMetadata(),
                            JcrConstants.NT_UNSTRUCTURED, vm);
                    });
            );
        request.setAttribute(DataSource.class.getName(), ds);
    }
}

```

Рисунок 4.6 – Servlet для динамічного відображення даних в діалоговому вікні компонента Tariff Comparator

Окрім описаних частин функціональний рівень розробленого застосунку містить Service клас, Util класи та Workflow крок.

Service в АЕМ – це Java-класи які виконують специфічну функціональність або бізнес-логіку та використовуються іншими частинами програми, такими як компоненти, сервлети. В програмі використовується лише один сервіс OffersHolderServiceImpl, зображений на рис. 4.7. Він потрібен для зберігання закешованих даних з репозиторію.

Workflow крок представляє собою java клас, який додає серверну логіку для автоматичної обробки excel файлів завантажених в АЕМ репозиторій. Він зображений на рис. 4.8.

```

@Component(service = OffersHolderService.class)
public class OffersHolderServiceImpl implements OffersHolderService {
    @Reference
    private ResourceResolverFactory resourceResolverFactory;

    private List<NetworkHardwareDto> hardwareProducts;
    private List<TariffPlanDto> tariffPlans;

    @Activate
    public void activate() {
        try (ResourceResolver resourceResolver = CQUtils.getResourceResolver(resourceResolverFactory)) {
            hardwareProducts = (List<NetworkHardwareDto>) initOffer(resourceResolver, AssetType.NETWORK_HARDWARE);
            tariffPlans = (List<TariffPlanDto>) initOffer(resourceResolver, AssetType.TARIFF_PLAN);
        } catch (LoginException e) {
            LOGGER.error("Failed to get activate service");
        }
    }

    private List<?> initOffer(ResourceResolver resourceResolver, AssetType assetType) {
        String repositoryPath = assetType.getRepositoryPath();
        Resource datasourceResource = resourceResolver.getResource(repositoryPath);

        if (Objects.isNull(datasourceResource)) {
            LOGGER.error("Failed to get resource by {} path", repositoryPath);
            return Collections.emptyList();
        }
    }
}

```

Рисунок 4.7 – Service для зберігання закешованих даних з репозиторію

```

@Component(service = WorkflowProcess.class, immediate = true, property = {
    "process.label=" + "Parse XML Asset"
})
public class ParseXMLAssetProcess implements WorkflowProcess {
    @Reference
    private ResourceResolverFactory resourceResolverFactory;

    @Reference
    private OffersHolderService offersHolderService;

    @Override
    public void execute(WorkItem workItem, WorkflowSession workflowSession, MetaDataMap metaDataMap) throws WorkflowException {
        try (ResourceResolver resourceResolver = CQUtils.getResourceResolver(workflowSession.getSession(), resourceResolverFactory)) {
            String payloadPath = workItem.getWorkflowData().getPayload().toString();
            Resource payloadResource = resourceResolver.getResource(payloadPath);

            if (Objects.isNull(payloadResource)) {
                LOGGER.error("Failed to get payload resource by {} path.", payloadPath);
                return;
            }

            Resource payloadMetadata = payloadResource.getChild("jcr:content/metadata");

            if (Objects.isNull(payloadMetadata)) {
                LOGGER.error("Failed to get payload metadata for resource {}.", payloadPath);
                return;
            }
        }
    }
}

```

Рисунок 4.8 – Workflow крок для автоматичної обробки excel файлів

Util класи –це java класи, що містять набір статичних методів і констант, які забезпечують допоміжні функції для спрощення розробки. В розробленій програмі утіліти займаються здебільшого трансформацією даних з одного формату в інший. На рис. 4.9 та 4.10 зображені util клас ExcelDataExporter, що експортує дані з excel документа та JsonDataTransformer, що трансформує дані в json формат. Ці утіліти використовують сторонні бібліотеки для опрацювання даних.

```
public class ExcelDataExporter {
    private ExcelDataExporter() {
    }
    public static <T> List<T> exportExcelData(InputStream stream, Class<T> type) {
        return Poiji.fromExcel(stream, PoijiExcelType.XLSX, type);
    }
}
```

Рисунок 4.9 – util клас ExcelDataExporter

```
public class JsonDataTransformer {
    private JsonDataTransformer() {
    }
    public static <T> List<T> json2Collection(String json, Class<T> type) {
        return new Gson().fromJson(json, TypeToken.getParameterized(List.class, type).getType());
    }
    public static String collection2Json(List<?> collection) { return new Gson().toJson(collection); }
}
```

Рисунок 4.10 – util клас JsonDataTransformer

Таким чином презентаційний та сервісний рівні взаємодіють між собою і утворюють цілісну архітектуру, яка забезпечує ефективну обробку даних, динамічне відображення контенту та зручний інтерфейс для користувачів.

4.3 Додана функціональність для розробки вебсайтів телекомунікаційних компаній

Модифікація АЕМ для телекомунікаційних компаній полягає в розширенні можливостей управління контентом шляхом створення інтеграції автоматизованого імпорту даних з excel документів та їх відображення через спеціалізовані компоненти.

Для структурованого описання доданої функціональності використовуються діаграми діяльності. Діаграми діяльності є невід'ємною частиною уніфікованої мови моделювання (UML), яка допомагає візуалізувати робочі процеси або види діяльності в системі. Вони показують, як різні дії пов'язані між собою і як система переходить з одного стану в інший, полегшуючи розуміння взаємодії елементів у системі [18].

Додану функціональність логічно розділити на дві окремі діаграми. Перша зображена на рис. 4.11 та описує процес завантаження, обробки та збереження даних з excel документа.

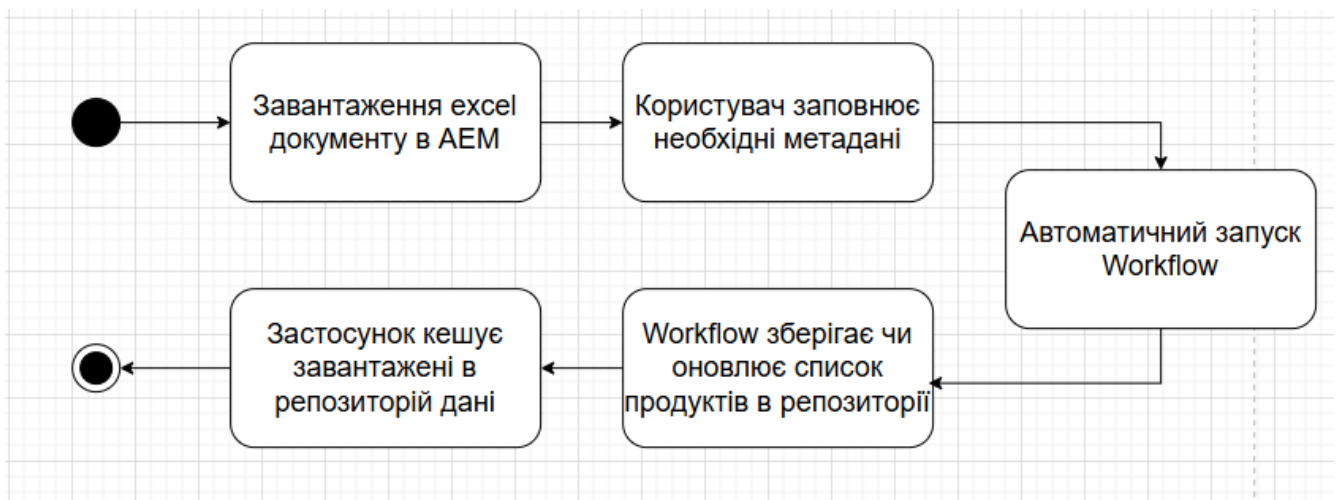


Рисунок 4.11 – Діаграма діяльності процесу завантаження та обробки excel документа

Передумовою до початку описаної функціональності є створений excel документ. Наразі існує 2 види документів доступних для обробки - це перелік мережного обладнання та телекомунікаційних тарифів. Вони зображені на рис. 4.12 та 4.13 відповідно.

Product ID	Product Name	Category	Description	Price	Speed/Capacity	Ports	Network Type	Features	Availability
201	Router 3000X	Router	High-speed router for home and small offices.	150	1Gbps	4	Wi-Fi 6, Wired	Dual-band, VPN, QoS, MU-MIMO	Available
202	Switch 24-Port Giga	Switch	24-port gigabit switch for large offices.	200	10Gbps	24	Wired	Layer 2, VLAN support, rack mountable	Available
203	Wi-Fi Router 5000 Pro	Router	Enterprise-grade router with advanced security.	350	2.5Gbps	6	Wi-Fi 6	WPA3, QoS, Beamforming, VPN Support	Available
204	Hub 8-Port	Hub	Simple 8-port hub for small office networks.	30	100Mbps	8	Wired	Basic data transfer, Plug and play	Available
205	ADSL Modem 2001	Modem	ADSL modem for high-speed broadband connections.	50	24Mbps	1	ADSL	Built-in router, 2x2 MIMO, WPA2	Available
206	Firewall 5000	Firewall	Next-gen firewall with advanced threat protection.	600	10Gbps	6	Wired, Wi-Fi	Intrusion detection, URL filtering	Limited
207	Outdoor Access Point	Access Point	Outdoor Wi-Fi access point for large areas.	120	1Gbps	2	Wi-Fi 5	Weatherproof, Dual-band, PoE	Available
208	Managed Switch 48-Port	Switch	Managed 48-port gigabit switch with advanced controls.	400	10Gbps	48	Wired	Layer 3, VLAN, SNMP, PoE	Available

Рисунок 4.12 – Файл переліку мережного обладнання

Product ID	Product Name	Category	Description	Price	Monthly Cost	Contract Duration	Data Allowance	Voice Allowance	Text Allowance	Network Coverage	Availability
101	Unlimited Mobile Plan	Mobile Plan	Best for heavy mobile data users.	50	50	12	Unlimited	Unlimited	Unlimited	Nationwide	Available
102	Basic Mobile Plan	Mobile Plan	Suitable for light users, affordable and simple.	20	20		12 5 GB	200 mins	100 texts	Nationwide	Available
103	High-Speed Fiber	Home Internet	Ultra-fast speeds for heavy internet users.	70	70		24 500 GB	N/A	N/A	Nationwide	Limited
104	Basic Broadband	Home Internet	Affordable home broadband plan.	30	30		12 50 GB	N/A	N/A	City Only	Available
105	Business Mobile Plan	Business Plan	Ideal for business users with higher data needs.	90	90		24 100 GB	1000 mins	500 texts	Nationwide	Available
106	International Roaming Plan	International	Great for frequent travelers, includes roaming.	100	100		12 10 GB	500 mins	500 texts	Global	Limited
107	Unlimited 5G Business Plan	Business Plan	High-speed 5G and unlimited data for businesses.	120	120		24 Unlimited	Unlimited	Unlimited	Major Cities	Available
108	Prepaid Mobile Plan	Mobile Plan	Pay-as-you-go for maximum flexibility.	10	10		0 2 GB	100 mins	100 texts	Nationwide	Available

Рисунок 4.13 – Файл переліку телекомунікаційних тарифів

Процес починається після завантаження excel документа в АЕМ. Це можна зробити вибравши розділ Asset на домашній сторінці платформи та перейшовши в спеціально створену папку Upload Asset Box. Далі потрібно перетягти потрібний документ в робочу область або натиснути кнопку Create і вибрати документ в файловій системі комп'ютера. Після цього файл опиниться в робочій області з червоною стрічкою, що показує необхідність заповнення метаданих. Робоча панель після завантаження excel документа зображена на рис. 4.14.

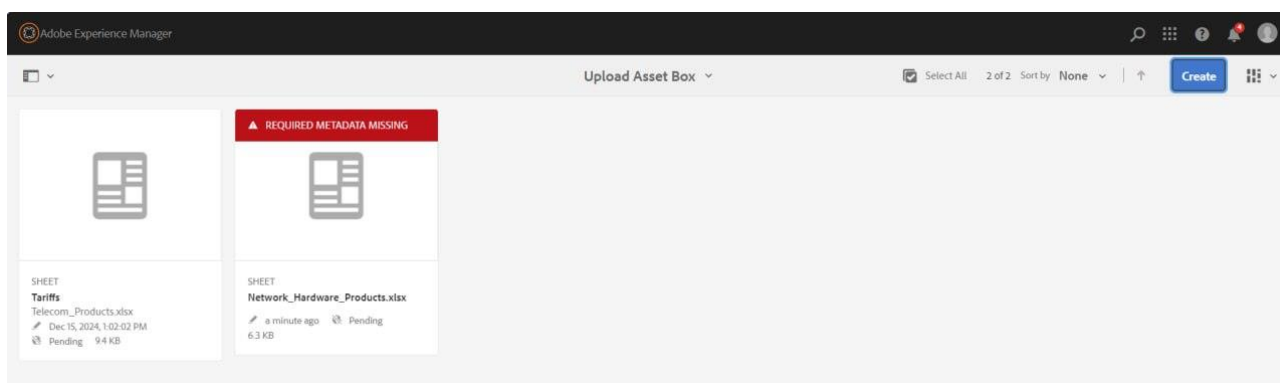


Рисунок 4.14 – Робоча панель після завантаження excel документа

Наступний крок це заповнення метаданих. Натиснувши на створений документ відкриється сторінка деталей про документ, в ній потрібно заповнити необхідні рядки. Важливо вибрати XML Type Tag на основі нього програма буде розрізняти вид документа та правильно його опрацьовувати. Заповнена метадата зображена на рис. 4.15.

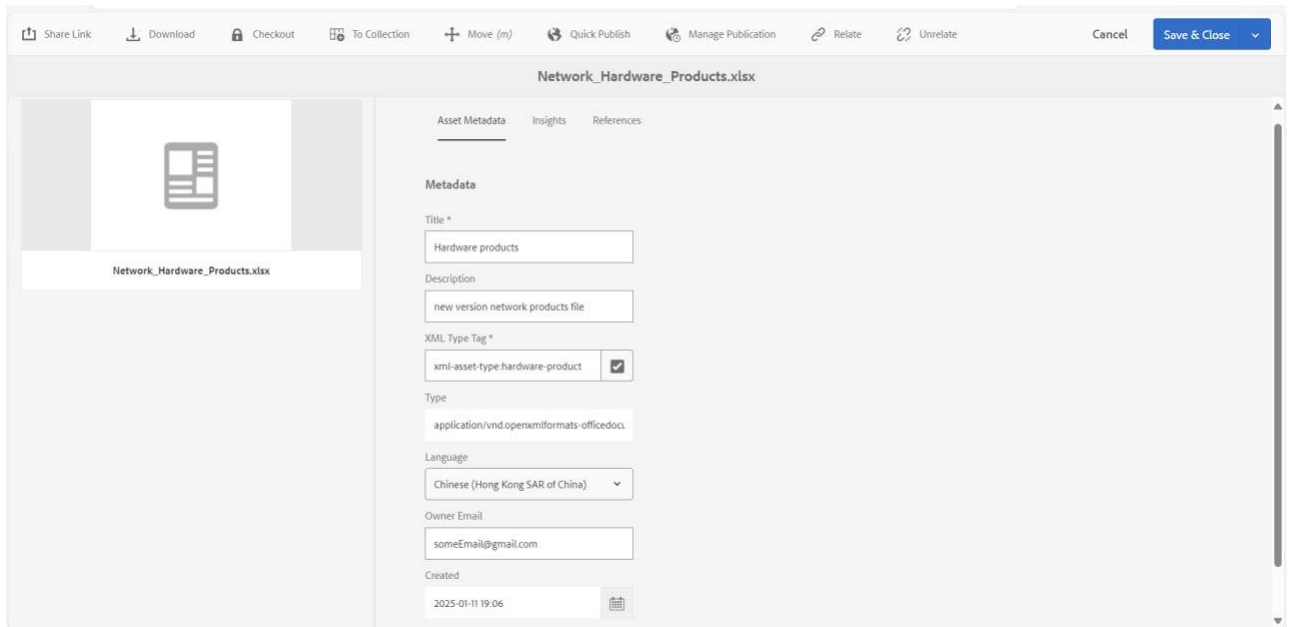


Рисунок 4.15 – Заповнена метадата завантаженого документа

Далі Workflow Launcher автоматично запустить Upload XML Asset Workflow для цього файлу. Після закінчення робочого процесу в профіль AEM надійде повідомлення та дані з документа будуть збережені в репозиторії. В репозиторії дані зберігаються в окремому вузлі, зображеному на рис. 4.16 в форматі JSON. Окрім цього збережені дані будуть кешовані в системі для швидкого доступу при відображенні в компонентах.

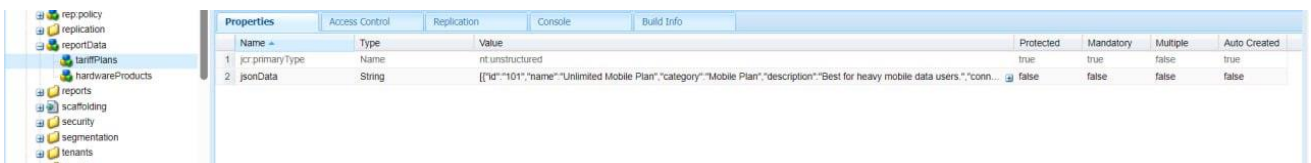


Рисунок 4.16 – Збережені дані з excel файлу в AEM репозиторії

Наступна діаграма діяльності зображена на рис. 4.17 та описує створення компонентів.

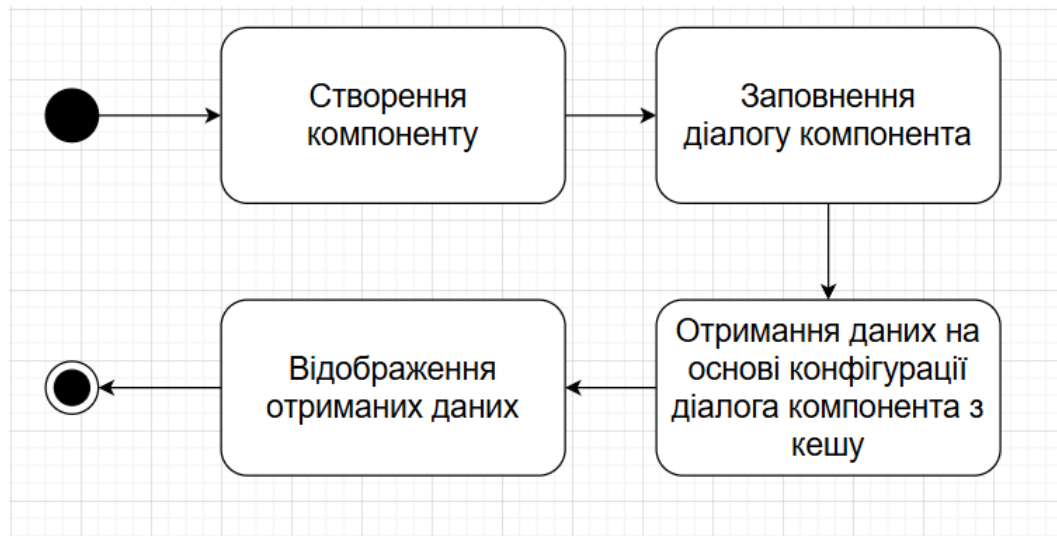


Рисунок 4.17– Діаграма діяльності процесу створення компонентів.

Процес створення компонентів складається з вибору відповідного компонента з списку компонентів на сторінці та заповнення діалогу компонента. Програма обробить дані зазначені при конфігурації компонента та відобразить результат.

Одним з таких компонентів є Tariff Comparator, зображений на рис. 4.18. Його призначення надати користувачам можливість порівнювати тарифні плани за ключовими параметрами для вибору найкращої пропозиції. Параметри включають вартість, обсяг даних, швидкість інтернету, доступність підключення тощо.

Компонент конфігурується за допомогою діалогового вікна, що відкривається при натисканні на нього, та дозволяє обрати два тарифа з випадаючого списку. Випадаючий список містить назви тарифів з завантаженого excel файлу. Діалог компонента зображений на рис. 4.19.

Наступним компонентом є Product Slider, зображений на рис. 4.20. Він створений для зручного відображення мережного обладнання у зручному форматі каруселі. Такий формат добре підходить для надання ключової інформації про групу продуктів. Він дозволяє акцентувати увагу на важливих характеристиках і перевагах без перевантаження сторінки надмірною кількістю тексту.

Basic Mobile Plan	High-Speed Fiber
Mobile Plan	Home Internet
<i>Suitable for light users, affordable and simple.</i>	<i>Ultra-fast speeds for heavy internet users.</i>
Price: 20\$	Price: 70\$
Monthly Cost: 20\$	Monthly Cost: 70\$
Contract Duration: 12 months	Contract Duration: 24 months
Data: 5 GB	Data: 500 GB
Voice: 200 mins	Voice: N/A
Text: 100 texts	Text: N/A
Network: Nationwide	Network: Nationwide
Availability: Available	Availability: Limited

Рисунок 4.18 – Компонент Tariff Comparator

Tariff Comparator

Properties

First Tariff

Basic Mobile Plan

Second Tariff

High-Speed Fiber

Рисунок 4.19 – Діалог компонента Tariff Comparator

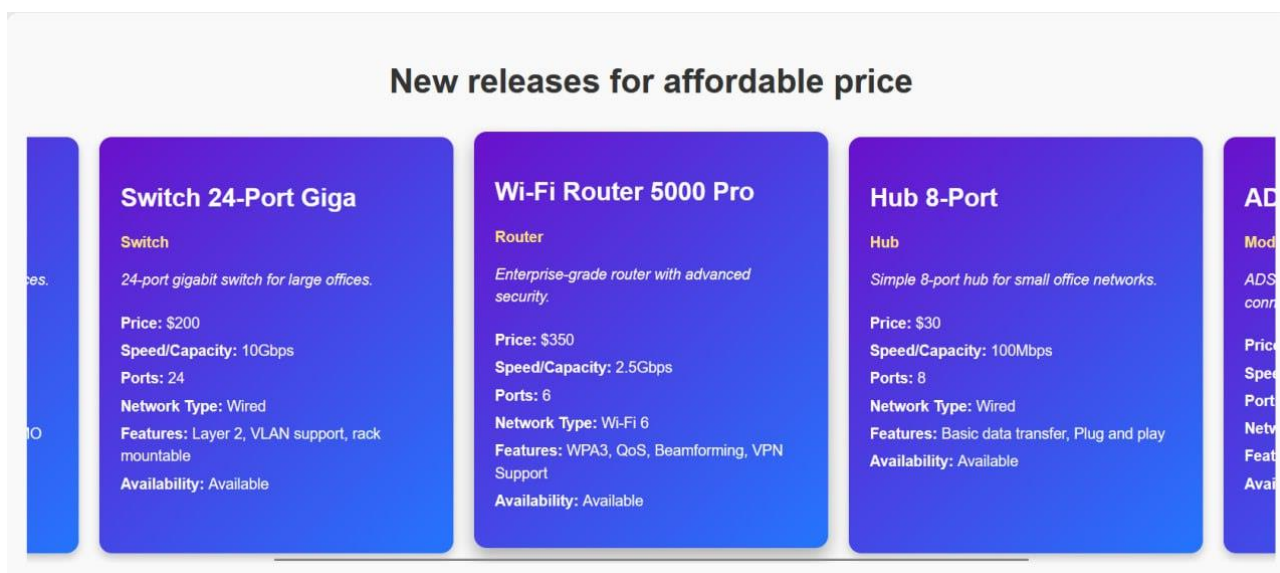


Рисунок 4.20 – Компонент Product Slider

Діалог компонента зображений на рис. 4.21. В ньому користувач створює набір продуктів додаючи їх персональний ідентифікатор в multifield. Також можливо задати заголовок групі обраних товарів.

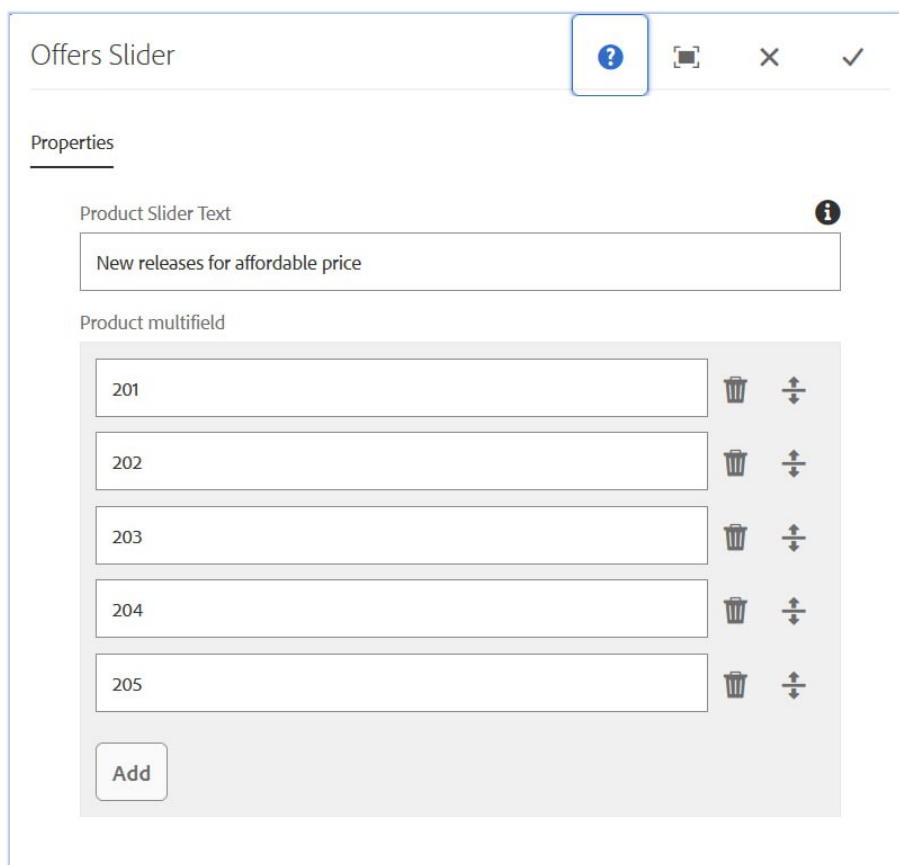


Рисунок 4.21 – Діалог компонента Product Slider

Останім розробленим компонентом є Offer, зображений на рис. 4.22 та рис. 4.23. Він необхідний для відображення детальної інформації про один конкретний товар або пропозицію. Цей компонент допомагає створювати контент з акцентом на індивідуальних товарах або сервісах. Крім цього, його зручно використовувати в рекламі або промо-акціях для привернення уваги до спеціальних пропозицій.

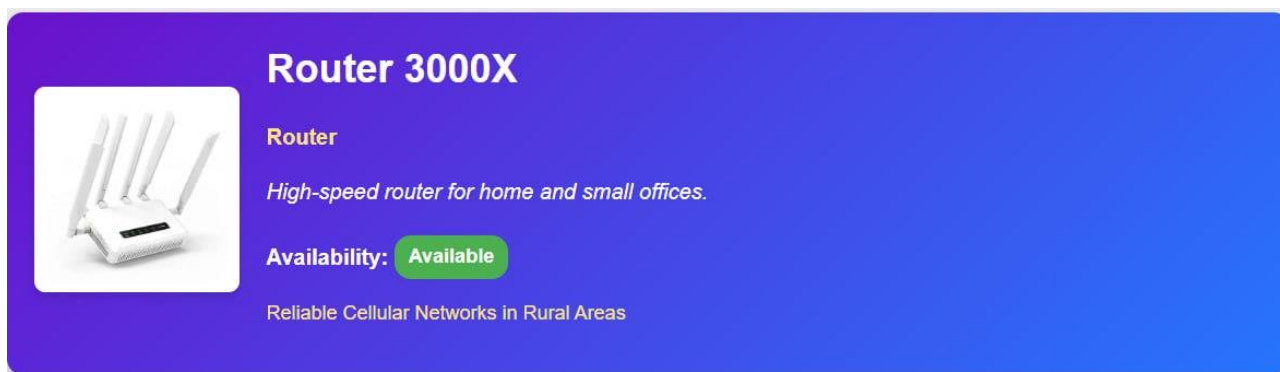


Рисунок 4.22 – Компонент Offer з конфігурацією мережного обладнання

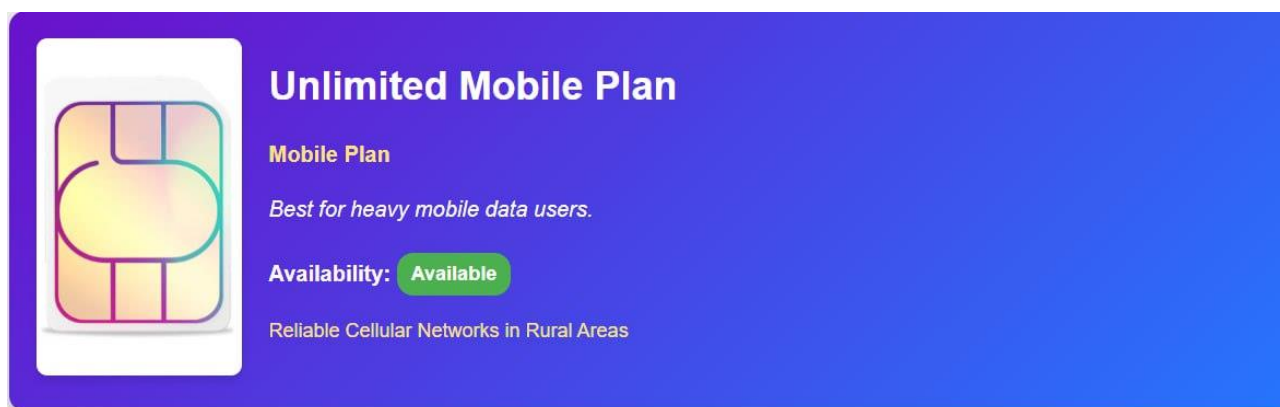
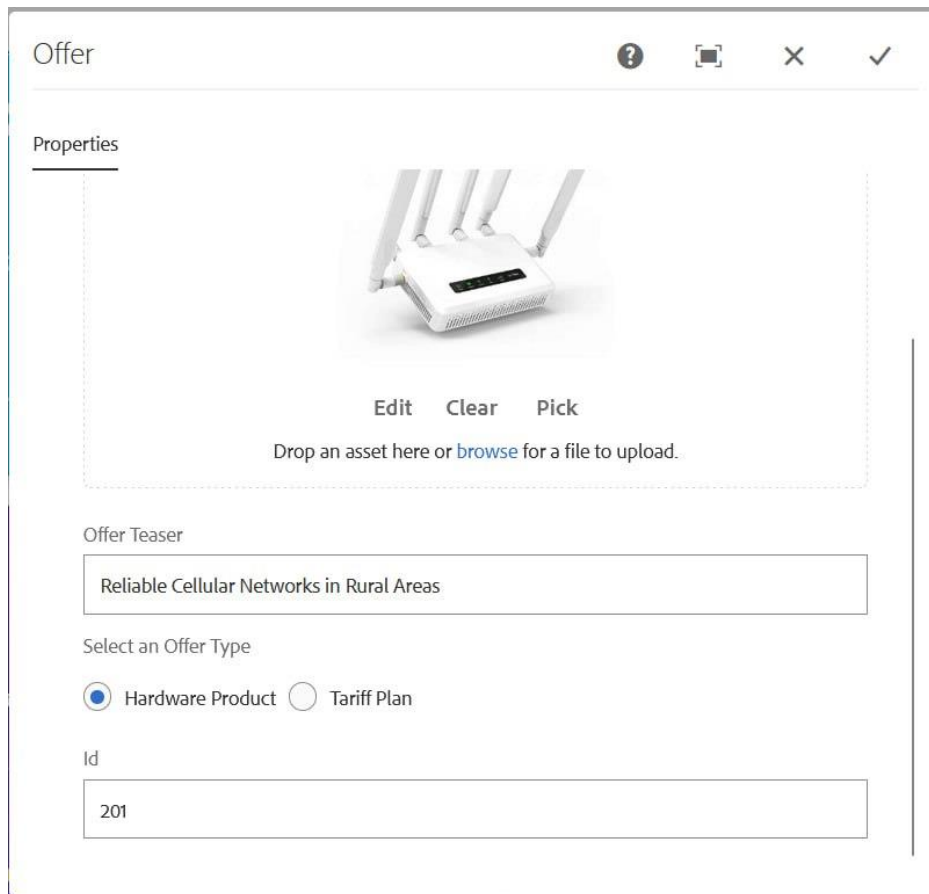


Рисунок 4.23 – Компонент Offer з конфігурацією тарифного плану

Компонент має діалог зображений на рис. 4.24. Він містить фото пропозиції, короткий опис, радіо кнопку для вибору типу пропозиції та її персональний ідентифікатор. Цей діалог є універсальним та дозволяє створити опис мережного продукту чи тарифного плану.



The screenshot shows a dialog box titled "Offer" with a standard window header containing a help icon, a maximize icon, a close icon, and a checkmark icon. Below the title bar, there is a "Properties" section. This section features a central image of a white wireless router with four antennas. Below the image are three buttons: "Edit", "Clear", and "Pick". Underneath these buttons is a dashed border indicating a drop zone, with the text "Drop an asset here or [browse](#) for a file to upload." Below the "Properties" section is an "Offer Teaser" field containing the text "Reliable Cellular Networks in Rural Areas". This is followed by a "Select an Offer Type" section with two radio buttons: "Hardware Product" (which is selected) and "Tariff Plan". At the bottom, there is an "Id" field containing the number "201".

Рисунок 4.24 – Діалог компонента Offer

Отже, модифікація АЕМ для телекомунікаційних компаній дозволяє інтегрувати в АЕМ платформу автоматизований імпорт даних з excel документів. Впроваджені компоненти, такі як Tariff Comparator, Product Slider та Offer, покращують користувацький досвід, дозволяючи зручно відображати та порівнювати пропозиції й продукти.

ВИСНОВКИ

У сучасному світі стрімкого розвитку цифрових технологій системи управління контентом стали невід'ємною частиною успішного функціонування вебсайтів. Вони прискорюють процес створення контенту, спрощують та автоматизують його управління. Це дозволяє швидко адаптуватись до тенденцій ринку та вподобань користувачів.

В першому розділі кваліфікаційної роботи визначено сутність терміну CMS та описані основні компоненти системи управління контентом. Також представлений розвиток та архітектурні можливості таких систем.

У другому розділі проаналізовано тенденції ринку CMS на прикладі популярних платформ WordPress, Shopify, Wix. Аналіз показав, що всі з них мають схожий базовий функціонал, який дозволяє легко створювати та управляти контентом, а також підтримують плагіни і модулі розширення. Проте ключові відмінності між платформами обумовлені наявністю специфічних функцій, які орієнтовані на потреби окремих галузей.

Третій розділ присвячено дослідженню системи Adobe Experience Manager. Виконано загальний огляд технологічного стеку, інфраструктури платформи, основних можливостей та політик формування вартості програмного продукту.

В четвертому розділі виконано і описано розробку проєкту вдосконалення платформи АЕМ для розробки вебсайтів телекомунікаційних компаній. В результаті була створена функціональність для автоматизованого імпорту даних з excel документів та відображення їх через спеціалізовані компоненти, описана архітектура проєкту, створені діаграми діяльності.

В роботі показана необхідність використання CMS систем, зокрема АЕМ для ефективного створення та управління контентом. А також розроблений сучасний масштабований та портативний застосунок вдосконалення платформи АЕМ для розробки вебсайтів телекомунікаційних компаній.

Деякі результати роботи було представлено на дванадцятій міжнародній науково-технічній конференції «Проблеми інформатизації» та опубліковано тези доповідей [19] за тематикою кваліфікаційної роботи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. TechTarget: What is a content management system (CMS)? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techtarget.com/searchcontentmanagement/definition/content-management-system-CMS>
2. Scandiweb blog: What is CMS: Your Essential Guide to Content Management Systems [Електронний ресурс] – Режим доступу до ресурсу: <https://scandiweb.com/blog/what-is-cms-guide-to-content-management-systems/>
3. Medium: The history of CMS — what has happened? [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@cuneax/the-history-of-cms-what-has-happened-c00266873be6>.
4. Adobe Experience Cloud Blog: The digital experience platform (DXP) — what it is, why you need it, and how to get started [Електронний ресурс] – Режим доступу до ресурсу: <https://business.adobe.com/blog/basics/what-is-digital-experience-platform-dxp>.
5. Sitecore: What is CMS architecture? [Електронний ресурс] – Режим доступу: <https://www.sitecore.com/resources/insights/development/what-is-cms-architecture>.
6. GeeksForGeeks: What is WordPress? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/what-is-wordpress/>
7. FreeCodeCamp: What is WordPress? The Blog and Website Tool Explained [Електронний ресурс] – Режим доступу до ресурсу: <https://www.freecodecamp.org/news/what-is-wordpress/>
8. AdwService Digital marketing tem: Shopify CMS Review: Platform Pros and Cons [Електронний ресурс] – Режим доступу до ресурсу: <https://adwservice.com.ua/en/shopify-cms-review>
9. WixBlog: What is Wix, you ask? Let us show you [Електронний ресурс] – Режим доступу до ресурсу: <https://www.wix.com/blog/what-is-wix>
10. Article: AEM Technology Stack [Електронний ресурс] – Режим доступу до ресурсу: <https://www.linkedin.com/pulse/aem-technology-stack-suresh-dhulipudi-msdre/>
11. Bounteous: AEM Infrastructure [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bounteous.com/insights/2018/02/20/aem-infrastructure/>

12. HurixDigital: Key Features and Capabilities of Adobe Experience Manager [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hurix.com/key-features-and-capabilities-of-adobe-experience-manager/#content>

13. Medium: AEM – Multi Site Management. Language copy Vs Live copy [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@manumathew28.94/aem-multi-site-management-523519bf633b>

14. Medium: Understanding AEM Workflow [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@manumathew28.94/understanding-aem-workflow-b017f0519c44>

15. WebJump: Adobe Experience Manager (AEM) guide: what it is, how it works, pricing guidelines, and more [Електронний ресурс] – Режим доступу до ресурсу: <https://webjump.ai/en/what-is-adobe-experience-manager/>

16. Multidots: What Influences the Cost of Adobe Experience Manager? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.multidots.com/blog/cost-of-adobe-experience-manager/>

17. Apache Maven Project: Welcome to Apache Maven [Електронний ресурс] – Режим доступу до ресурсу: <https://maven.apache.org/>

18 Activity Diagrams – Unified Modeling Language (UML) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/>

19. Чистюк Д.С. Вдосконалення системи управління контентом АЕМ для розробки вебсайтів телекомунікаційних компаній / Д.С. Чистюк, Д.В. Чеботарьова // Тези доповідей дванадцятої міжнародної науково-технічної конференції «Проблеми інформатизації», 21 – 22 листопада 2024 р., Баку – Харків – Бельсько-Бяла. 2024. – Том 1. – С. 68.