

РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ ТА ЇЇ РОЗГОРТАННЯ НА ХМАРНОМУ СЕРЕДОВИЩІ

Я.Л. Байтєльман¹, В.В. Поцєпаєв¹

¹ Department of Automation and Telecommunications, Donetsk National Technical University, Lutsk, Ukraine

E-mail: valerii.potsepaiev@donntu.edu.ua

Отримано 19.06.2024

Прийнято до публікації 01.08.2024

Опубліковано 01.11.2024

АНОТАЦІЯ

Метою даної роботи є створення простої та доступної для адаптації системи управління контентом, яка легко інтегрується у веб або мобільні застосунки, і може бути налаштована відповідно до специфічних потреб конкретного проекту. Однією з ключових переваг є її універсальність і гнучкість, що дозволяє розробникам швидко і з мінімальними зусиллями адаптувати програмне рішення до вимог конкретної галузі або бізнес-процесів.

Було проведено комплексну технічну розробку і створено програмне забезпечення, яке ґрунтується на використанні мови програмування PHP, реляційної бази даних MySQL і середовища для хмарного розгортання Amazon AWS. Така комбінація інструментів забезпечує високу продуктивність і надійність, що є критично важливим для сучасних застосунків. Робота містить детальні пояснення основних методів і функцій, які використовуються для обробки та збереження даних. Весь програмний код доступний у відкритому доступі.

Наукова новизна роботи полягає в тому, що матеріал викладено у доступній формі, що значно спрощує реалізацію завдань з управління контентом навіть для осіб, які не мають глибоких знань у програмуванні. Це відкриває нові можливості для міждисциплінарних досліджень і співпраці між різними галузями, зокрема в комп'ютерних науках та інших прикладних дисциплінах, таких як маркетинг, журналістика або освіта.

З точки зору практичної значимості наведені матеріали дозволяють за кілька годин розгорнути повноцінну систему управління контентом у хмарному середовищі Amazon AWS. Програмний код, із відповідними модифікаціями для реалізації різних бізнес-логік, був успішно впроваджений на декількох вебсайтах, які щоденно відвідують тисячі користувачів, що свідчить про ефективність і стабільність запропонованого рішення.

Матеріали роботи можуть бути корисними для навчання студентів технічних і економічних спеціальностей, а також школярів старших класів.

Ключові слова: управління контентом, вебсайт, PHP, MySQL, AWS

ВСТУП

Попри існування великого розмаїття рішень для управління контентом, включно із проектами відкритого коду (open source) в площині вирішення практичних завдань в різних галузях та індустріях часто виникає необхідність розробки компактного, простого для застосування і легкого в підтримці рішення. З урахуванням конкретних вимог до того або іншого проекту використання готових рішень передбачає значну адаптацію (кастомізацію), що вимагає, по-перше, детальне вивчення обраного рішення, а по-друге, внесення необхідних змін. Типовою є ситуація, коли взяте за основу рішення переробляється настільки суттєво, що фактично нічого не залишається від основи, або коли значний обсяг функціоналу взагалі не використовується, є зайвим, і таким чином суттєво ускладнюється підтримка і обслуговування, витрачаються пам'ять та обчислювальні потужності на виконання коду, який не є необхідним, а також є джерелом потенційних безпекових ризиків. Одним з прикладів таких універсальних систем управління контентом є WordPress, іншим – Joomla. Попри популярність цих систем і числені розроблені для них графічні теми та функціональні плагіни, добре відомими в індустрії є також скарги на роботу з ними. Окрім того, ці системи є "монолітними", що не дозволяє їх розгортання в безсерверному або мікросервісному середовищі, хоча саме безсерверні та мікросервісні рішення на сьогодні вважаються ефективними щодо вартості та обслуговування.

Автор мав власний досвід адаптації та використання WordPress і Joomla в сферах електронної комерції, бібліотечної справи та новин і в кожному проекті на тому чи іншому етапі стикався із економічно обґрунтованим рішенням відмовитись від цих систем на користь більш ефективної розробки.

АНАЛІЗ ЛІТЕРАТУРНИХ ДАНИХ ТА ПОСТАНОВКА ПРОБЛЕМИ

Мова програмування PHP [1] була вперше представлена в 1994 році у формі простої бібліотеки CGI, в перекладі українською – загального серверного інтерфейсу для виконання на сервері певного програмного коду, який викликається зверненням з боку клієнту через HTTP протокол [2]. Фактично технологія CGI забезпечила розвиток динамічних вебсайтів, на відміну від статичних документів, які просто завантажуються по запиту від клієнта. PHP була не першим втіленням CGI, проте швидко набуло популярності і не втрачає її завдяки своїй гнучкості і перед усім можливості як створювати

швидкі прості рішення на базі об'єднання в одному файлі коду HTML і JavaScript (які виконуються на клієнтській стороні) з кодом PHP (що виконується на стороні серверу), так і повноцінно підтримувати MVC-парадигму з чітким розподілом на модель, вигляд, контролер [3] і об'єктно-орієнтоване програмування [4]. Перше забезпечує розв'язання відносно простих задач на кшталт створення прототипів, перевірку гіпотез, швидке моделювання. Друге послугує потребам всього широкого спектру розробки програмних рішень різного рівня складності. Сервер бази даних MySQL є складовою частиною так званого набору інструментів LAMP, що складається з операційної системи Linux, веб-серверу Apache, серверу баз даних MySQL, мови програмування PHP, цей набір зберігає актуальність вже протягом багатьох років через низьку вартість і доступність [5]. В рамках даної роботи розглядається типова архітектура серверної сторони на основі LAMP. Покрокові інструкції з налаштування хмарного середовища Amazon AWS доступні на веб сайті з офіційною технічною документацією [6]. Варто зауважити, що аналогічне середовище можна розгорнути на потужностях будь-якого іншого хостінг-провайдера або навіть на власній апаратній базі, і запропоновані PHP-скрипти будуть виконуватися, проте пропонується використання деяких комерційних сервісів Amazon AWS, а саме CloudFront [7] для кешування і підтримки безпечного протоколу передачі гіпертексту, Route 53 [8] для налаштування DNS, S3 [9] для збереження статичних файлів, RDS [10] для розміщення бази даних та її автоматичного резервного копіювання, аналоги до яких можуть бути знайдені у інших провайдерів але їхнє використання вимагає внесення змін в технічний дизайн запропонованого рішення.

Формулювання проблеми: перед фахівцями різних сфер бізнесу, науки, державного чи місцевого врядування часто постає проблема організації управління контентом, при чому необхідно забезпечити віддалений доступ до такої системи через мережі загального користування (Інтернет), надійне зберігання інформації, включно із регулярним резервним копіюванням і гарантованим відновленням, а також забезпечити низку інших вимог з імплементації певної бізнес-логіки, що разом знижує привабливість використання готових рішень і натомість створює потребу в мінімалістичному наборі скриптів, які можна легко адаптувати під зазначені вище вимоги.

Мета: розробка програмного коду і моделей даних інструментами LAMP для розгортання на хмарному середовищі і з використанням спеціалізованих сервісів Amazon AWS.

Задачі:

1. Аналіз специфікацій сервісів Amazon AWS.

2. Розробка технічного дизайну.
3. Розробка і тестування програмного коду і моделей даних.
4. Огляд результатів роботи, підготовка пояснювальних матеріалів.

МАТЕРІАЛИ ТА МЕТОДИ ДОСЛІДЖЕНЬ

В роботі використано методи структурного і порівняльного аналізу, з інформаційним і аналітичним підходом розглянуто наукову та методичну літературу, а також онлайн ресурси, застосовано експериментальні методи для перевірки програмного коду.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Під терміном “контент” мається на увазі будь-які текстові або графічні дані, наприклад, матеріали для вебсайту служби новин, або історія пацієнтів для медичної інформаційної системи, або будь-яка інша бібліотека документів. В даній роботі розглядається система з вільним доступом до даних з боку кінцевих користувачів, що відповідає прикладу сайту новин.

З огляду на типи користувачів системи та їхні функції виокремлюються 3 типи:

1. Адміністратори, які мають найбільші повноваження в системі, можуть створювати та редагувати інших адміністраторів та редакторів контенту, надавати їм доступ до системи або блокувати його, також можуть редагувати контент.
2. Редактори контенту, мають права створювати та змінювати контент.
3. Кінцеві користувачі, мають права перегляду контенту.

Система управління контентом включає наступні компоненти:

1. Адміністративна панель, через яку здійснюється створення і редагування контенту адміністраторами системи; складається з скриптами з кодом HTML і JavaScript для виконання на клієнтській стороні та скриптами з кодом PHP для виконання на серверній стороні.
2. База даних, в якій зберігається текстова складова контенту і метадані (дата створення, інформація про автора, дозвіл на виведення у відповідь на запит з боку клієнтського інтерфейсу, належність до групи або розділу, тощо).
3. Файлове сховище, для зберігання файлів зображень (з можливим розширенням до будь-яких файлів, документів, відео і таке інше). Файлове сховище реалізується окремо від серверу, на якому зберігаються

скрипти серверного виконання з метою, за необхідності, перенести скрипти серверного виконання до безсерверного середовища.

4. Скрипти серверного виконання, які обслуговують запити від клієнтського веб або мобільного застосунку, повертаючи контент для відображення на боці користувача.
5. Система кешування та екранування доступу до файлів та скриптів з метою покращення їхнього захисту.
6. Веб або мобільний застосунок користувача (в даній роботі не наводиться програмний код користувацького застосунку, але визначається формат команд для звернення до відповідних скриптів серверного виконання).

Компонента діаграма наведена на Рис. 1.

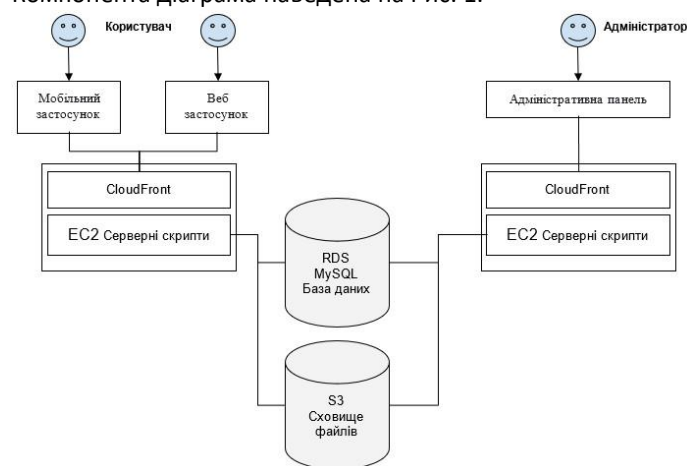


Рис. 1. Компонента діаграма

ОГЛЯД СЕРВІСІВ AWS

На сервісі EC2 (Рис. 2) створюється віртуальний сервер з потрібним обсягом оперативної пам'яті, процесорної потужності, розміром жорсткого диску. Дозволяється вибір операційної системи з списку доступних. Здійснюються налаштування безпеки та моніторингу. З практичного досвіду, для сайту з відвідуванням близько 1000 користувачів на добу, рекомендується обирати тип серверу “t2-micro”. В рамках умов даної роботи обирається операційна система Amazon Linux 2023. Базові конфігурації здійснюються через адміністративну веб-панель AWS EC2. Для встановлення веб серверу Apache та завантаженню файлів доступ до EC2 здійснюється через SSH. Покрокові інструкції наведено в офіційній документації [6]. В адміністративній веб-панелі AWS EC2 рекомендується налаштувати:

- моніторинг навантаження на сервер і відповідно надсилання повідомлення електронною поштою адміністратору в разі перевищення навантаження на процесор більше 50%;
- автоматичне резервне копіювання вмісту жорсткого диску серверу один раз на добу і зберігання копій щонайменше протягом 14 днів, що дасть можливість легкого і швидкого повернення по попередньої версії в разі необхідності;

- відкрити лише порти 443 (HTTPS), 80 (HTTP), 22 (SSH);
- асоціацію даного серверу з публічною IP адресою, яка буде використовуватися для підключення до нього через SSH. Як буде пояснено нижче, ця адреса не використовуватиметься в налаштуваннях DNS, тож вона потрібна лише для зручності адміністрування.

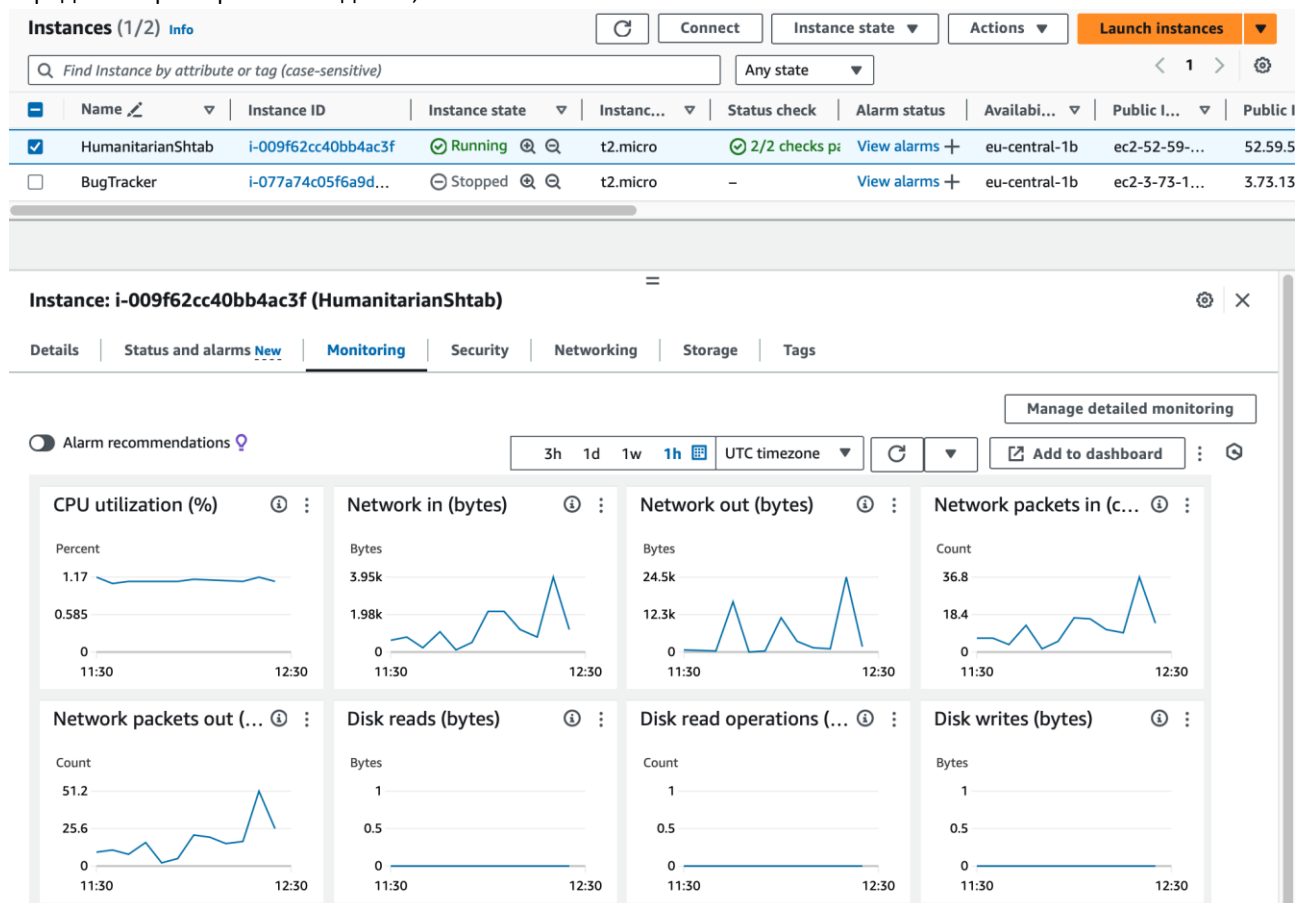


Рис.2. Налаштування EC2

За бажанням замість EC2 можна розміщувати скрипти серверного виконання на безсерверних сервісах, наприклад Lambda [11]. Альтернативно, для обслуговування великої кількості одночасних запитів, можна встановити режим автоматичного збільшення ресурсів серверу EC2, або розподіл запитів через балансир навантаження на декілька EC2 серверів, хоча це виходить за межі даної роботи, автор вважає за потрібне вказати на наявність таких можливостей для втілення різних підходів до динамічних змін потужності в залежності від навантаження та вимог конкретного проекту. Так само важливо зазначити, що AWS підтримує різні географічні регіони, тобто для обслуговування

клієнтів в Північній Америці варто обирати американські географічні регіони, для європейських клієнтів – відповідно європейські, для Азії – свої регіони. Існує можливість синхронізації серверів, розташованих в різних зонах.

CloudFront – це сервіс AWS, що виконує декілька функцій, важливих для захисту і оптимізації доступу до даних. По-перше, він приховує публічну IP адресу EC2 серверу, по-друге здійснює кешування даних, що дозволяє зменшити навантаження на EC2, наприклад, відповідь на запит клієнта на каталог розділів веб сайту можна кешувати, так як така інформація не змінюється часто, тож тривалість життя кешу можна встановити на

рівні кількох годин. Крім того, через CloudFront безкоштовно замовляється SSL сертифікат для безпечного HTTPS з'єднання з сервером. Таким чином всі ресурси в запропонованій системі будуть доступні тільки через

HTTPS, доступ через HTTP буде заблокований, а запити HTTP будуть автоматично переадресовані на HTTPS (Рис. 3). Деталі налаштування CloudFront наведено в

The screenshot displays the AWS CloudFront console for distribution E17UFYFE9RV2NV. It is divided into two main sections: 'Details' and 'Settings'. The 'Details' section includes fields for 'Distribution domain name' (d3s8cq5yvwq4yp.cloudfront.net), 'ARN' (arn:aws:cloudfront::963776958110:distribution/E17UFYFE9RV2NV), and 'Last modified' (April 2, 2022 at 5:56:50 PM UTC). The 'Settings' section includes 'Description' (humanitaryishtab.org), 'Price class', 'Use all edge locations (best performance)', 'Supported HTTP versions' (HTTP/2, HTTP/1.1, HTTP/1.0), 'Alternate domain names' (www.humanitaryishtab.org, *.humanitaryishtab.org), 'Custom SSL certificate' (humanitaryishtab.org), 'Security policy' (TLSv1.2_2021), and 'Standard logging' (Off), 'Cookie logging' (Off), and 'Default root object' (-).

Рис.3. Налаштування CloudFront

документації до цього сервісу[7]. Важливо зауважити, що замовлення SSL потребує налаштування адресації в сервісі домених імен Route 53 (наведено нижче). Крім того, використання CloudFront певною мірою відіграє роль захисту від можливих DDoS атак.

Сервіс RDS дозволяє створювати бази даних без необхідності витрат зусиль на підтримку і обслуговування серверу баз даних (Рис. 4), при чому в залежності від необхідності існує можливість автоматичного збільшення чи зменшення потужності обраного RDS середовища в залежності від навантаження. Для даної роботи обрано найменше доступне середовище RDS з підтримкою MySQL, а саме db.4tg.micro, із загальним обсягом 50GB і автоматичним розширенням до 100GB за умови виникненні необхідності, налаштовано автоматичне щоденне резервне копіювання із зберіганням останніх 7 копій. Докладно про налаштування RDS можна прочитати в технічній документації сервісу [10].

Важливо пам'ятати, що винесення бази даних в окремий сервіс на RDS зменшує навантаження на сервер EC2, проте користування кожним з сервісів AWS не є безкоштовним. Утримання найменшого серверу RDS

коштує приблизно \$19 щомісяця, а найменшого EC2 – приблизно \$16 за місяць, в залежності від трафіку ці цифри можуть зростати. Amazon пропонує програму підтримки стартапів і кожному новому клієнтові надає певний безкоштовний ліміт ресурсів на перші 12 місяців, що є зручним для новачків і дає можливість практично ознайомитися з різними сервісами AWS, але з таким підходом новачки воліють до нехтування вартістю сервісів і через рік стикаються з неприємним сюрпризом, коли отримують перший рахунок. З іншого боку, винесення бази даних в RDS зменшує ризики злому та витоку даних, так як на сервері EC2 не виконується програмне забезпечення серверу баз даних і відповідно його потенційні помилки і безпекові недоліки є повністю виключеними. Те саме вірно і для веб серверу та операційної системи EC2: якщо скрипти серверного виконання розмістити в безсерверному середовищі, наприклад Lambda, тоді буде повністю відсутнім програмне забезпечення Apache і Linux разом з їхніми можливими недоліками і ризиками, проте вартість Lambda може перевищувати вартість серверу EC2. Загальне правило таке: чим більше навантаження, чим більше

користувачів, тим вигіднішими стають безсерверні рішення.

Сервіс зберігання статичних файлів S3 є безсерверним середовищем, із ключовим елементом “bucket”

(українською – “відро”). Його слід розуміти як безрозмірну директорію, де можна зберігати необмежену кількість файлів будь-якого типу і розміру, в ній також можна створювати необмежену кількість

Instance		
Configuration	Instance class	Storage
DB instance ID database-1	Instance class db.t4g.micro	Encryption Enabled
Engine version 8.0.33	vCPU 2	AWS KMS key aws/rds
DB name -	RAM 1 GB	Storage type General Purpose SSD (gp2)
License model General Public License	Availability	Storage 50 GiB
Option groups default:mysql-8-0 In sync	Master username root	Provisioned IOPS -
Amazon Resource Name (ARN) arn:aws:rds:eu-west-1:963776958110:db:database-1	Master password *****	Storage throughput -
Resource ID	IAM DB authentication Not enabled	Storage autoscaling Enabled

Рис. 4. Налаштування RDS

файлів будь-якого типу і розміру, в ній також можна створювати необмежену кількість вкладених директорій, так як це можливо у файлової системі на комп'ютері. Імена файлів в директорії мають бути унікальними. Доступ до файлів і директорій можна налаштовувати з будь-яким ступенем відкритості, від повної доступності для будь-кого, навіть через веб, і таким чином bucket можна перетворити на статичний веб сайт, до приватності кожного окремого файлу чи директорії, так щоб доступ до них був лише у власника, і відповідно, bucket може бути приватним архівом. Вартість збереження файлів на S3 є найменшою серед усіх сервісів AWS, меншою за збереження на жорстких дисках EC2 і значно меншою за збереження в базі даних RDS. Саме тому є сенс виносити зберігання статичних файлів в S3. Крім того S3 також має механізми резервного копіювання, в тому числі через збереження файлів в “дзеркальний” bucket. Файли і директорії в S3 зуться об'єктами (Рис. 5). Як і для інших

сервісів Amazon пропонує повну технічну документацію для S3 [9].

Важливо зауважити наступне:

- файли на S3 не можуть виконуватись, на відміну від файлів на жорсткому диску комп'ютера чи сервера, розташований на S3 скрипт з програмним кодом або скомпільований бінарний файл не може бути викликаним для виконання, так як в середовищі S3 немає оперативної пам'яті і немає процесору;
- існують бібліотеки (SDK) для різних мов, в тому числі для PHP, і в цій роботі така бібліотека використовується, для програмного завантаження файлів на S3, їхнього видалення, копіювання, тощо. За принципом дії це схоже до функцій роботи з файлами, тільки у випадку з S3 файли знаходяться на віддаленому сховищі.

Сервіс конфігурації доменних імен Route 53 використовується для налаштувань адресації доступу до фронт-енд скриптів клієнтського застосунку та бек-енду,

причому у відповідності до Рис. 1 і Табл. 1, маємо підключити:

- адресу доменного імені – до дистрибутиву CloudFront, що вказує на розташування HTML, CSS та JavaScript

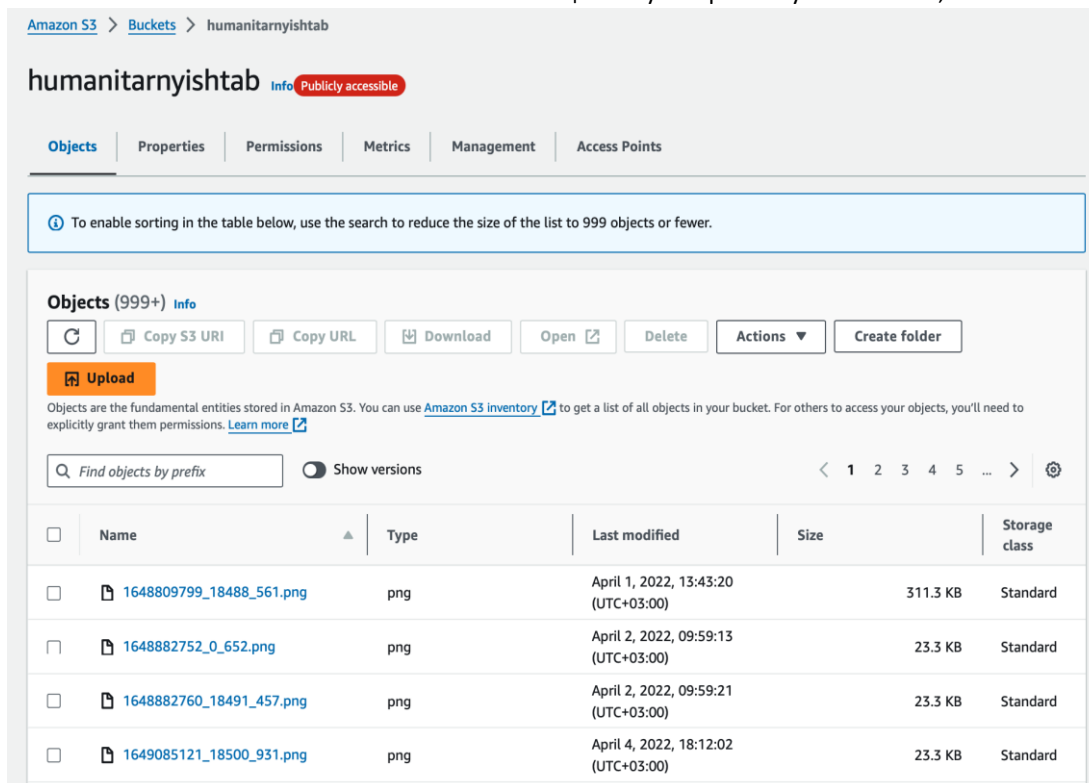


Рис.5. Файли в S3

файлів фронт-енд веб застосунку на S3;

- адресу піддомену скриптів API для обслуговування запитів від фронт-енду – до дистрибутиву CloudFront, що вказує на розташування цих скриптів на EC2;
- адресу піддомену адміністративної панелі – до дистрибутиву CloudFront, що вказує на розташування файлів PHP на EC2;

- адресу піддомену статичного контенту – до дистрибутиву CloudFront, що вказує на розташування статичних файлів, таких як графічні зображення, які завантажуються через адміністративну панель систему управління контентом, на S3.

Табл. 1. Відповідність адрес, дистрибутивів та ресурсів

Адреса	Дистрибутив CloudFront	Сервіс AWS	Призначення
myproj.com.ua	XXX1.cloudfront.net *	S3	Файли HTML, JS, CSS та інші клієнтського веб застосунку
services.myproj.com.ua	XXX2.cloudfront.net *	EC2	Файли бек-енд скриптів для клієнтського застосунку
admin.myproj.com.ua	XXX3.cloudfront.net *	EC2	Файли PHP адміністративної панелі
resources.myproj.com.ua	XXX4.cloudfront.net *	S3	Статичний контент (зображення, тощо)

* XXX1, XXX2 і так далі представляють собою значення, згенеровані сервісом CloudFront.

Таким чином приховуються реальні IP-адреси сервісів (утіліта `ping` буде повертати IP-адреси серверів CloudFront, а не ресурсів, на яких реально розташовані ті чи інші файли), також доступ до кожного сервісу

опиняється під SSL. Route 53 можна використовувати для налаштувань адресації поштових сервісів, проте це виходить в межі даної роботи (Рис. 6).

The screenshot shows the AWS Route 53 console interface for the hosted zone 'humanitaryishtab.org'. At the top, there are navigation links for 'Route 53', 'Hosted zones', and the specific zone name. Below this, there are buttons for 'Delete zone', 'Test record', and 'Configure query logging'. A section titled 'Hosted zone details' includes an 'Edit hosted zone' button. Below that, there are tabs for 'Records (10)', 'DNSSEC signing', and 'Hosted zone tags (0)'. The 'Records (10)' tab is active, showing a search bar and buttons for 'Delete record', 'Import zone file', and 'Create record'. A table lists the records:

Record name	Type	Routin...	Differe...	Alias	Value/Route traffic to
humanitaryishtab.org	A	Simple	-	Yes	d3s8cq5yvq4yp.cloudfront.net.
humanitaryishtab.org	MX	Simple	-	No	10 inbound-smtp.eu-west-1.amazonaws.com
humanitaryishtab.org	NS	Simple	-	No	ns-813.awsdns-37.net. ns-1776.awsdns-30.co.uk. ns-159.awsdns-19.com. ns-1359.awsdns-41.org.

Рис. 6. Записи в Route 53

ДІАГРАМИ ПОСЛІДОВНОСТІ ВИКОНАННЯ ЗВЕРНЕНЬ

Діаграма послідовності виконання звернень при відображенні контенту за запитом від клієнтського веб застосунку наведена на Рис. 7. Після завантаження коду HTML, JavaScript, CSS відбувається його інтерпретація та виконання в браузері і відповідно до його функцій може виконуватися завантаження додаткових статичних елементів, наприклад, зображень, а також здійснюватися запити на виконання серверних скриптів, наприклад, коду PHP, що знаходиться на сервері EC2 і виконується на ньому, звертається до бази даних для отримання або збереження інформації. Запити до бази даних здійснюються мовою MySQL і виконуються на сервері RDS.

Діаграма послідовності звернень при відображенні контенту за запитом від клієнтського мобільного застосунку наведена на Рис. 8. Мобільний застосунок звертається до API бек-енду та завантажує статичні файли аналогічно до дій веб застосунку. Єдина відмінність полягає в тому, що мобільний застосунок не потребує завантаження власного коду з серверу кожного разу, коли користувач запускає застосунок на виконання.

Діаграма послідовності звернень при збереженні контенту за запитом від адміністративної панелі наведена на Рис. 9.

СТРУКТУРА БАЗИ ДАНИХ

Система управління контентом пропонує організацію даних у формі статей, статті можуть належати до однієї чи більше категорій, до статті можуть бути додані графічні файли. Доступ до редукування здійснюється адміністраторами. Зазначена вище інформація зберігаються в базі даних в наступних таблицях.

Таблиця `admin_users` (Таб. 2) містить дані про користувачів адміністративної панелі. Відповідно до безпекових рекомендацій, паролі зберігаються у хеш-форматі, який не дозволяє зворотної трансформації для отримання значення паролю в його первинній текстовій формі.

Таблиця `categories` (Таб. 3) містить дані про категорії статей. Кожна стаття може належати до однієї категорії. Кожна категорія має асоційовану з нею піктограму з бібліотеки Font Awesome.

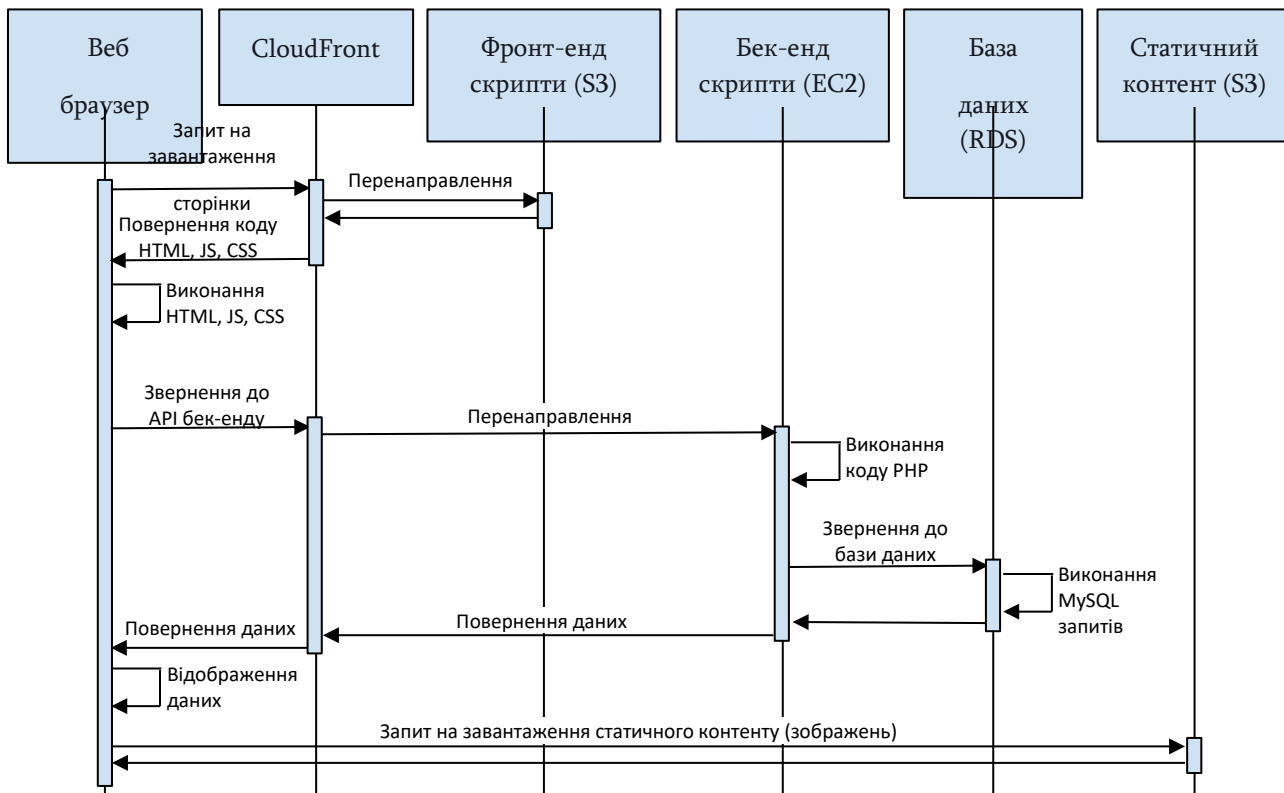


Рис. 7. Запит контенту від вебзастосунку

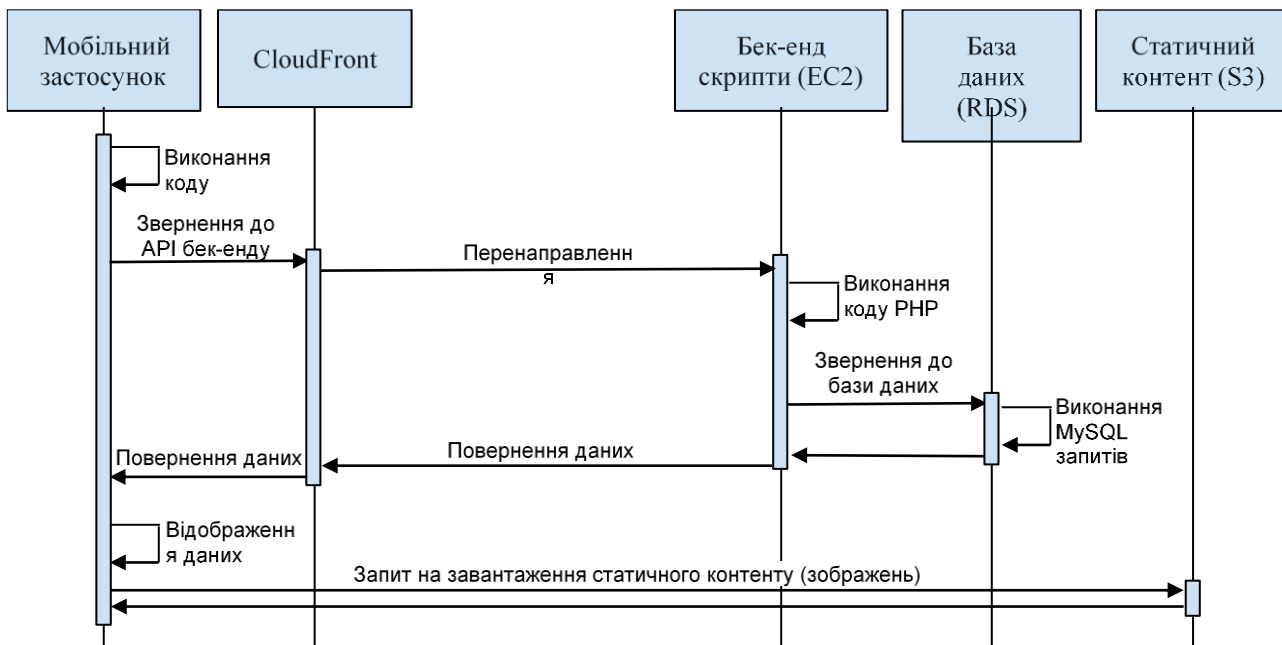


Рис. 8. Запит контенту від мобільного застосунку

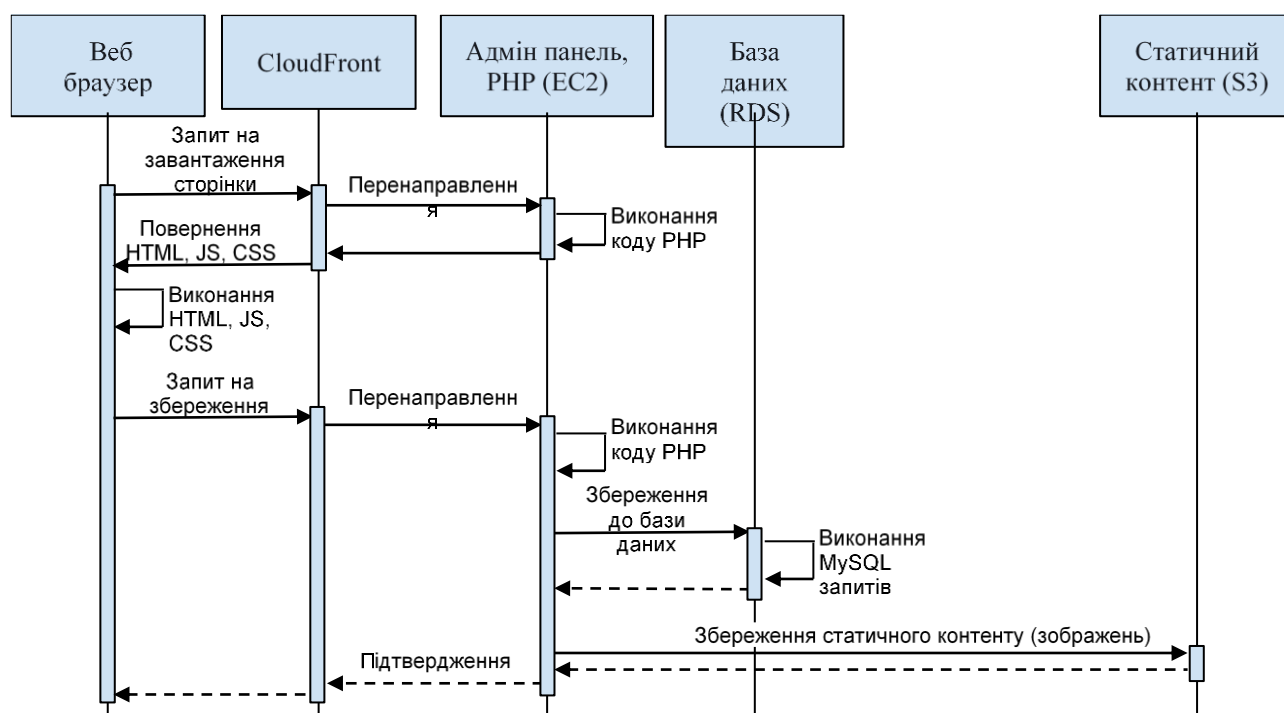


Рис. 9. Збереження контенту

Табл. 2. Структура таблиці admin_users

Поле	Тип	Примітки
id	int (11)	Унікальний ідентифікатор, первинний ключ, автоінкремент
login	varchar (50)	Ім'я користувача
password	varchar (50)	Пароль

Табл. 3. Структура таблиці categories

Поле	Тип	Примітки
id	int (11)	Унікальний ідентифікатор, первинний ключ, автоінкремент.
title	varchar (200)	Назва категорії
icon	varchar (100)	Піктограма
color	varchar (20)	Колір піктограми
bg_color	varchar (20)	Колір фону
t_color	varchar (20)	Колір тексту

Таблиця content (Таб. 4) містить самі статті.

Табл. 4. Структура таблиці content

Поле	Тип	Примітки
id	int (11)	Унікальний ідентифікатор, первинний ключ, автоінкремент.
title	varchar (255)	Назва статті. Індекс.
introtext	mediumtext	Текст вступу. Індекс.
full_text	longtext	Повний текст статті
created	int (11)	Дата створення (unixtime)
created_by	int (11)	Автор. Id з таблиці admin_users
images	text	Інформація про зображення
category_id	int (11)	Id з таблиці categories
is_gall_vertical	int (11)	Тип галереї зображень

API ДЛЯ ОБСЛУГОВУВАННЯ ЗАПИТІВ

API для обслуговування запитів від клієнтського веб або мобільного застосунків включає 2 інтерфейси. Кожний підтримує звернення методами get та post. Відповідь повертається у форматі JSON.

Для отримання списку категорій викликається інтерфейс categories.

URL:

<https://services.myproj.com.ua/api/categories.php>

Параметри:

q – критерій пошуку. Не обов'язковий. Якщо не заданий, повертаються всі категорії.

page – номер сторінки. Не обов'язковий. Значення за замовчуванням 0.

count – кількість елементів на сторінці. Не обов'язковий. Значення за замовчуванням 50.

Приклад:

<https://services.myproj.com.ua/api/categories.php?q=про>

Відповідь:

```
{
```

```
"page":0,
"count":50,
"total":"1",
"items":
[
{
"id":"4",
"title":"\u041f\u0440\u043e\u0434\u0443\u043a\u0443\u0430\u0442\u0442\u0438\u0445\u0430\u0444\u0447\u0448\u0445\u0430\u0440\u0440\u0447\u0443\u0442\u0434\u0434\u0444",
"icon":"<i class=\"fa-solid fa-plate-wheat\"> </i>",
"color":"black",
"bg_color":"#E6E6E6",
"T_color":"black"
}
]
```

Важливо:

1. Відповідь приходить у не форматованому вигляді, тобто виступи у прикладі вище зроблені для наочності.

2. Кириличні символи, тобто будь-який текст українською або іншою мовою, що використовує не ASCII символи, будуть закодовані, а всім лапкам будуть передувати символи \, щоб лапки, що в тексті, відрізнялись від лапок, що обмежують назву ключа чи

відправляє назад лише дані (і далі виконання PHP припиняється, весь HTML код не відправляється знову); JavaScript, отримавши відповідь, виводить результати в певні елементи HTML. Таким чином, все що потрібно, наприклад, для функціоналу збереження та редагування публікації, знаходиться в одному PHP файлі, це є зручним, коли над проектом працює одна особа, проте, при масштабуванні на певному етапі настає потреба забезпечити розподіл обов'язків в команді, і тоді має сенс розділити HTML і JavaScript та PHP частини, перше стає відповідальністю інженера-розробника клієнтської частини, друге – програміста серверної частини. Також, можливо замінити HTML і JavaScript на, наприклад, ReactJS.

inc/db.php містить функції підключення до бази даних, а також ім'я користувача, пароль, адресу серверу бази даних. Крім того, тут містяться деякі спільні функції, які використовуються в інших файлах, наприклад функція створення паролю. Даний файл не викликається безпосередньо, а включається в інші файли.

index.php – скрипт, з якого починається взаємодія з адміністративною панеллю, тут відбувається авторизація

користувачів.

[Панель Керування](#)

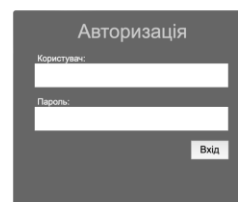


Рис. 11. Index.php

inc/session.php – скрипт для перевірки, чи є користувач авторизованим. Включається до скриптів всіх “внутрішніх” сторінок, доступних лише для авторизованих користувачів (Рис. 11).

inc/menu.php – скрипт з путнками меню, вони однакові для “внутрішніх” сторінок.

articles.php виводить список всього контенту (публікацій), з можливістю посторінкового перегляду та пошуку (Рис. 12).

Гуманітарний штаб: Публікації Користувачі Категорії Різні Вихід									
Публікації									
Пошук <input type="text"/> Категорія <input type="text"/> Область <input type="text"/> <input type="button" value="Шукати"/> <input type="button" value="Нова публікація"/>									
ID	Категорія	Область	Місто	Назва	Кількість	Автор	Дата	Actions	
19901				Звіт допомоги за тиждень	0	Anastasia	26-02-2024	🔗	
19900		Вінницька	Літин	#ДітиМаютьПравоНаДитинство	0	Anastasia	22-02-2024	🔗	
19899	Військова амуніція			18 тепловізорів, 12 прожекторів, 34 зарядні станції Bandera Po...	0	Anastasia	22-02-2024	🔗	
19898	Військова амуніція			П'ять безпілотників Mavic 3T для бійців бригади "Червона ...	0	Anastasia	19-02-2024	🔗	
19897	Продукти харчування			Продукти харчування, вода, медикаменти та окопні свічк...	0	Anastasia	19-02-2024	🔗	
19896	Військова			Mavic 3T для бійців 14-ї штурмової бригади	0	Anastasia	16-02-2024	🔗	

Рис. 12. articles.php

article_edit.php – створення та редагування статті (Рис. 13, 14).

categories.php – перегляд категорій публікацій (Рис. 15).

Публікація

Категорія: Якщо категорію не обрати, це буде просто новина (в стрічці новин) і не буде враховуватися в статистиці.

Додаткові категорії: Якщо відправлення містить предмети із різних категорій.

Область:

Місто:

Дата:

Назва:

Кількість:

Одиниці виміру:

Вступ:

Текст:

B I U Normal

Потреба у "залізних пташках" на полі бою залишається стабільно високою. Бо кожен безпілотною — це і збережені життя наших захисників, і втрати ворога у живій силі та техніці.
Поки наші хлопці роблять там усе можливе задля Перемоги, ми маємо робити те саме тут. Тому об'єднуємо зусилля та працюємо згуртовано далі!

Рис. 13. article_edit.php (редагування тексту)

Потреба у "залізних пташках" на полі бою залишається стабільно високою. Бо кожен безпілотною — це і збережені життя наших захисників, і втрати ворога у живій силі та техніці.
Поки наші хлопці роблять там усе можливе задля Перемоги, ми маємо робити те саме тут. Тому об'єднуємо зусилля та працюємо згуртовано далі!

Розташування зображень:

Зображення:

no file selected




Рис. 14. article_edit.php (редагування зображення)

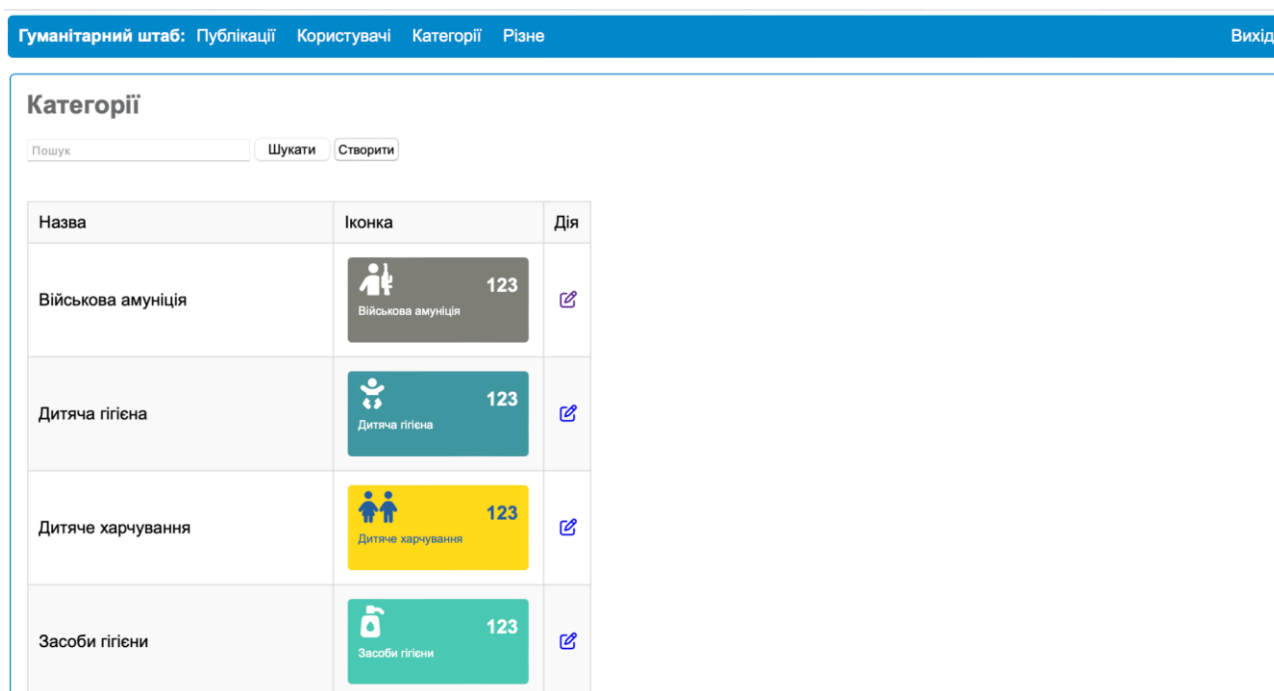


Рис. 15. categories.php

category_edit.php – створення та редагування категорії (Рис. 16).

users.php – перегляд списку користувачів (адміністраторів, Рис. 17). В системі є 2 рівня користувачів – адміністратори з повним доступом та користувачі, які мають доступ лише до редагування контенту.

Назва:

Іконка:

Колір:

Колір фону:

Колір лічильника:

Рис. 16. category_edit.php

ID	Логін	Ел. пошта	Тип	Статус	Дія
15	Anastasia	dzi...@gmail.com	Користувач	Активний	
14	yelena	yel...@mail.com	Адміністратор	Активний	

Рис. 17. users.php

user_edit.php – створення та редагування користувачів. Важливо зауважити, паролі не зберігаються у відкритому вигляді, а шифруються незворотнім чином [14] (Рис. 18). Якщо користувач втрачає пароль, відновити його не є можливим, тому буде створений новий пароль. Також слід додати, що авторизація користувачів за одним фактором (паролем) не вважається надійною, тому варто

додати другий фактор, наприклад, одноразовий пароль, що надсилатиметься користувачу по електронній пошті або SMS.

logout.php – очищає дані в сесії та виконує перенаправлення на index.php.

Рис. 18. user_edit.php

ПРИКЛАДИ ВЕБ-САЙТІВ

В Табл. 5 перелічені проекти, де було застосовано дану систему управління контентом.

Табл. 5. Приклади веб сайтів

URL	Розширення базового функціоналу
humanitarnyishtab.org	Додаткові поля в публікаціях, належність публікації до кількох категорій.
vmas.info	2-факторна авторизація, підтримка 3 мов
catholic.co.il	2-факторна авторизація, підтримка 5 мов, розширений каталог категорій
daily-gospel.net	2-факторна авторизація, змінена структура контенту

ВИСНОВКИ

В ході проведення дослідження отримані наступні результати:

1. Проведено аналіз специфікацій сервісів Amazon AWS та PHP.
2. Розроблено архітектуру системи управління контентом.
3. Розроблено і протестовано структуру бази даних та програмний код.
4. Наведено пояснення щодо реалізації програмного коду у вигляді коментарів в кодї та пояснювальних матеріалів в тексті даної роботи.

ДОСТУПНІСТЬ ДАНИХ

Програмний код викладено для вільного доступу за посиланням: <https://btlr.com/code/cms/>.

ЛІТЕРАТУРА

- [1] PHP documentation. [Онлайн]. URL: <https://www.php.net>. Дата звернення: 17.10.2024.
- [2] Understanding HTTP using Browsers. [Онлайн]. URL: <http://surl.li/vwkvym>. Дата звернення: 17.10.2024.
- [3] MVC. [Онлайн]. URL: <http://surl.li/ziawqy>. Дата звернення: 17.10.2024.
- [4] Introduction of Object Oriented Programming. [Онлайн]. URL: <http://surl.li/slyzju>. Дата звернення: 17.10.2024.
- [5] What is a LAMP stack? [Онлайн]. URL: <http://surl.li/gmesgm>. Дата звернення: 17.10.2024.
- [6] Tutorial: Install a LAMP server on AL2023. [Онлайн]. URL: <http://surl.li/pkutti>. Дата звернення: 17.10.2024.
- [7] Amazon CloudFront. [Онлайн]. URL: <http://surl.li/oqwnkq>. Дата звернення: 17.10.2024.
- [8] Amazon Route 53. [Онлайн]. URL: <http://surl.li/tusvxf>. Дата звернення: 17.10.2024.
- [9] Amazon S3. [Онлайн]. URL: <http://surl.li/qbziyg>. Дата звернення: 17.10.2024.
- [10] Amazon Relational Database Service. [Онлайн]. URL: <http://surl.li/awliinq>. Дата звернення: 17.10.2024.
- [11] Amazon Lambda. [Онлайн]. URL: <http://surl.li/brdjuw>. Дата звернення: 17.10.2024.
- [12] PHP documentation json_encode. [Онлайн]. URL: <http://surl.li/ejpfyb>. Дата звернення: 17.10.2024.
- [13] JSON Decode Online. [Онлайн]. URL: <http://surl.li/dsbdou>. Дата звернення: 17.10.2024.
- [14] PHP documentation password_hash. [Онлайн]. URL: <http://surl.li/vjrdlf>. Дата звернення: 17.10.2024.

DEVELOPMENT OF A CONTENT MANAGEMENT SYSTEM AND ITS CLOUD DEPLOYMENT

Yakiv (Jacob) Baytelman, Valerii Potsepaiev

The aim of this work is to create a simple and easily adaptable content management system that can be seamlessly integrated into web or mobile applications and customized according to the specific needs of a particular project. One of its key advantages is its versatility and flexibility, allowing developers to quickly and effortlessly tailor the software solution to the requirements of specific industries or business processes.

A comprehensive technical development process was carried out, resulting in the creation of software based on the use of the PHP programming language, the MySQL relational database, and the Amazon AWS cloud deployment environment. This combination of tools ensures high performance and reliability, which are critically important for modern applications. The work provides detailed explanations of the core methods and functions used for data processing and storage. All of the software code is available as open source.

The scientific novelty of this work lies in the presentation of the material in an accessible format, significantly simplifying the implementation of content management tasks even for individuals without deep programming knowledge. This opens up new possibilities for interdisciplinary research and collaboration across various fields, particularly in computer science and other applied disciplines such as marketing, journalism, or education.

From a practical standpoint, the presented materials allow for the deployment of a full-fledged content management system in the Amazon AWS cloud environment within a few hours. The software code, with appropriate modifications to implement different business logic, has been successfully integrated into several websites that are visited daily by thousands of users, demonstrating the effectiveness and stability of the proposed solution.

The materials of this work can be useful for educating students of technical and economic specialties, as well as senior high school students.

Keywords: content management, website, PHP, MySQL, AWS.

REFERENCES

- [1] PHP documentation. [Online]. URL: <https://www.php.net>. Accessed: 17.10.2024.
- [2] Understanding HTTP using Browsers. [Online]. URL: <http://surl.li/vwkvym>. Accessed: 17.10.2024.
- [3] MVC. [Online]. URL: <http://surl.li/ziawqy>. Accessed: 17.10.2024.
- [4] Introduction of Object Oriented Programming. [Online]. URL: <http://surl.li/slyzju>. Accessed: 17.10.2024.

- [5] What is a LAMP stack? [Online]. URL: <http://surl.li/gmesgm>. Accessed: 17.10.2024.
- [6] Tutorial: Install a LAMP server on AL2023. [Online]. URL: <http://surl.li/pkutti>. Accessed: 17.10.2024.
- [7] Amazon CloudFront. [Online]. URL: <http://surl.li/oqwnkq>. Accessed: 17.10.2024.
- [8] Amazon Route 53. [Online]. URL: <http://surl.li/tusvxf>. Accessed: 17.10.2024.
- [9] Amazon S3. [Online]. URL: <http://surl.li/qbziyg>. Accessed: 17.10.2024.
- [10] Amazon Relational Database Service. [Online]. URL: <http://surl.li/awling>. Accessed: 17.10.2024.
- [11] Amazon Lambda. [Online]. URL: <http://surl.li/brdjuw>. Accessed: 17.10.2024.
- [12] PHP documentation json_encode. [Online]. URL: <http://surl.li/ejpfyb>. Accessed: 17.10.2024.
- [13] JSON Decode Online. [Online]. URL: <http://surl.li/dsbdou>. Accessed: 17.10.2024.
- [14] PHP documentation password_hash. [Online]. URL: <http://surl.li/vjrdlf>. Accessed: 17.10.2024.