



Приватний заклад вищої освіти
Одеський технологічний університет «ШАГ»
Кафедра інформаційних технологій та фундаментальної
підготовки

Випускна кваліфікаційна робота бакалавра

«СИСТЕМА КЕРУВАННЯ ПОТОКОВИМ КОНТЕНТОМ»

на здобуття бакалаврського рівня вищої освіти
зі спеціальності «122 Комп'ютерні науки»

Виконавці проєкту

№	П.І.Б.	Група
1	Байдуков М.М.	КН-П-181
2	Головань В.В.	КН-П-181
3	Чайковский А.В.	КН-Д-182

Автор звіту

_____ Байдуков Микола Миколайович

Ментори:

№ з/п	ПІБ, посада	Напрямок
1	Самойленко Д.М.	Розробка ПЗ
2	Судавная Т.О.	Графічний дизайн

АНОТАЦІЯ

УДК 004.9

Б-18

Байдуков М.М. Система керування поточковим контентом, що відповідає за функціонування його внутрішньої частини : Автореферат випускної кваліфікаційної роботи на здобуття бакалаврського рівня вищої освіти зі спеціальності «122 Комп'ютерні науки». - Одеса : ОТУШ, 2022 - 21 с.

Робота присвячена дослідженням, розвиненню та впровадженню інформаційної системи, головним призначенням якої є узгоджена взаємодія між користувачами з системою керування поточковим контентом, що відповідає за функціонування його внутрішньої частини. Додаткові функції системи включають реєстрацію, авторизацію та автентифікацію, засоби обмеження доступу до контенту, ведення журналів активності користувачів системи, послуги зберігання, доставки та показу.

Актуальність роботи визначається поширеністю задач, які ґрунтуються на оперативній фільтрації результатів за зберігання, доставки та відображення контенту. Особливу увагу звертає на себе той факт, що зазвичай складається з двох основних компонентів: додатки для керування контентом як зовнішній інтерфейс користувача, що дозволяє користувачеві додавати, змінювати і видаляти контент з веб-сайту без втручання веб-майстра, і додаток доставки контенту, яке компілює контент і оновлює веб-сайт.

Об'єктом дослідження у рамках даної роботи виступає алгоритм складання комплексної оцінки результатів за багатьма критеріями в умовах неповної визначеності вихідних умов. Предметом дослідження обрано фактор контенту користувача на момент формування запиту. У якості задач дослідження було встановлене наступне:

- провести огляд інформаційних систем, що використовується для забезпечення та організації спільного процесу створення, редагування та управління вмістом.
- визначити програмні засоби (інтерфейси API), які дозволяють отримувати дані про контент користувача, встановити їх особливості та ступінь вживаності для поставлених завдань.
- скласти програмне забезпечення, реалізувавши у ньому результати, досягнуті у попередніх задачах, провести випробування за різних умов та для різних пристроїв.
- впровадити у систему програмні засоби захисту даних для покращення показників інформаційної безпеки створеної системи.

Методи дослідження, що використані для досягнення поставлених задач, утворені суперпозицією методів системного аналізу, моделювання та когнітивної методології.

При складанні звіту використано 42 посилання на друковані та електронні джерела інформації.

ЗМІСТ

ВСТУП _____	3
Технічне завдання до проєкту _____	4
РОЗДІЛ 1. Загальна характеристика системи _____	6
РОЗДІЛ 2. Реалізація серверної частини _____	7
РОЗДІЛ 3. Результати випробувань _____	10
ВИСНОВКИ _____	14
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ _____	13

ВСТУП

Розвиток і поширення інформаційних технологій супроводжується перенесенням великої кількості послуг у цифровий простір. Стратегія розвитку інформаційного суспільства в Україні визначає, що «комунікаційні технології є необхідним інструментом соціально-економічного прогресу, одним з основних чинників інноваційного розвитку економіки».

Сучасні мобільні пристрої, - смартфони, планшети, ноутбуки, - практично всі обладнані пристроями визначення геолокації, що значно покращує ефективність прикладного програмного забезпечення, яке здатне врахувати у своїй роботі дані щодо розміщення користувача. Подібні технології широко проникають у всі галузі діяльності, створюючи окремі напрями наукового дослідження.

У той же час, широкий вжиток інформаційних систем, особливо з реалізованими засобами коригування контентом, потребує покращення інформаційної безпеки як щодо захисту персональних даних користувача, так і щодо відомостей про його місцезнаходження. Це актуалізує додаткові дослідження і розробки в зазначеній області.

В якості предмету даної роботи було обрано досить поширену систему узгодження взаємодії користувачів до функцій якої належать розміщення контенту, що надходять з боку одних користувачів, та пошук відповідних пропозицій з боку інших. При пошуку має враховуватись обмеження, щодо вікової межі користувачів.

Також функції переважної більшості подібних системи мають передбачати можливість збереження історії розміщення, пошуку пропозицій, а також фактів їх любові чи неприязнь. У перспективі - засобів оплати. Відповідно, це передбачає засоби авторизації та автентифікації, а також убезпечення персональних даних від несанкціонованого доступу інших користувачів.

Детальне технічне завдання до проєкту наводиться далі. У рамках даного звіту розглядається серверної частина, що являє собою web-додаток, створений для користувачів пристроїв з операційною системою «Windows». Звіт складається з трьох розділів у яких описується загальна характеристика системи, деталі створення серверної частини та результати її розгортання та тестування. Висновки відбивають результати, що досягнуті у рамках даної частини проєкту, загальний висновок є сукупністю звітів усіх учасників проєкту.

ТЕХНІЧНЕ ЗАВДАННЯ ДО ПРОЄКТУ

Загальний опис.

Програмне забезпечення (ПЗ) призначене для створення розподіленої системи управління контентом, який має потоковий зміст. Головна особливість системи полягає у тому, що наявний контент має бути забезпечений засобами відтворення (перегляду або прослуховування) без необхідності завантаження. Функції системи мають забезпечити повний життєвий цикл управління контентом: створення, актуалізація, модифікація та знищення.

Призначення:

ПЗ орієнтується на створення узагальненої концепції контейнера для поточкових даних. Головний акцент робиться на засобах групування, пошуку та розподілу доступу до контенту. Функції відтворення контенту не є безпосереднім предметом завдання і можуть бути запозичені у третіх сторін. ПЗ створюється за прикладом сервісу відеохостингу, що надає послуги розміщення відеоматеріалів, у той же час має складатись за принципом універсальності, мати змогу змінити цільове призначення.

Вимоги до функціоналу. Обов'язкова частина

Головна сторінка

Стартовий інтерфейс ПЗ (головна сторінка або активність) має за мету популяризацію системи. На ньому має бути забезпечено відображення популярних показників як-то загальна кількість користувачів та інформаційних одиниць, вибірка з найбільш популярного контенту та нових дописів, зовнішня реклама. Посилання на інші сторінки (засоби) системи, а також на сторінки партнерів.

Реєстрація користувачів:

Користувач ПЗ має змогу зареєструватись як за допомогою авторизаційних даних (логіну/паролю), так і за допомогою облікових записів популярних соціальних мереж (Google, Facebook, тощо).

При реєстрації обов'язково зазначається засіб оперативного контактування (ЗОК) - телефон та електронна пошта. На тестовому етапі, за умови ускладнення замовлення послуг телефонії чи СМС, можливе використання лише електронної пошти. Засіб оперативного контактування при реєстрації підлягає верифікації через відповідь на тестовий виклик чи надісланий лист, або введення коду, пересланого через повідомлення довільного типу.

Рекомендується використовувати ЗОК у якості авторизаційного логіну користувача.

За умови що користувач забув пароль, ЗОК має бути використаний для його відновлення. Схема відновлення паролю жорстко не регламентується.

Також передбачається наявність у системі користувачів з додатковими правами: модераторів та адміністраторів. Реєстрація кореневого адміністратора системи має бути реалізоване при інсталяції системи, наступним користувачам змінюють права наявні адміністратори.

Управління власним контентом:

Основний ресурс зареєстрованого користувача ПЗ. Повинен забезпечити усі необхідні засоби для

- Створення даних та метаданих (розміщення відео та його характеристик). Серед рекомендованих характеристик: опис, режим доступу (загальний/приватний), категорія, теги, вікові обмеження, дозвіл/заборона коментування.
- Встановлення, зміна та вилучення метаданих у власному контенті (редагування)
- Групування контенту (за принципом плейлистів)
- Перегляд поточної активності: історії переглядів, перелік відзначеного контенту (вибране), контактні особи (друзі), коментарі до контенту, тощо
- Пошук контенту інших користувачів за метаданими, групування та фільтрування результатів за різними ознаками, наявними у знайденому контенті. Для об'ємних результатів має бути забезпечена пагінація.
- Контакткування з іншими користувачами, що розміщують контент, за допомогою ЗОК або через внутрішні засоби системи (чат або)

Також засобами особистого кабінету має бути передбачено можливість зміни персональних даних. У разі зміни ЗОК необхідна його актуалізація за тією ж схемою, що й при реєстрації.

Зворотній зв'язок щодо контенту

Користувач має змогу встановити власну оцінку (лайк або бал) для довільного контенту. Якщо дозволено коментування контенту, користувач має змогу залишити коментар. Дозволяється встановлювати обмеження на кількість коментарів від одного користувача. Засоби пошуку, фільтрування та групування мають забезпечити врахування рейтингу контенту, у т.ч. наявність чи кількість коментарів, наприклад, виводити результати у порядку за якого першими йдуть найбільш коментовані одиниці.

Вимоги до функціоналу. Опціональна частина

Модерація

Для забезпечення дотримання загальноприйнятих норм спілкування, а також з метою недопущення публікування контенту, який містить заборонені компоненти (ненормативні висловлювання, заклики до протиправних дій, приниження чи дискримінація, тощо), усі інформаційні одиниці, а також створені користувачами метадані (описи, теги), а також коментарі мають можливість бути заблокованими. Право блокування належить кореневому адміністратору системи, а також модераторам. При погодженні інформаційна одиниця має зберігати ідентифікатор користувача, який встановив блокування. Також має бути зворотна можливість розблокування заблокованого контенту.

Привілейовані функції

З метою покращення економічної ефективності проєкту передбачається наявність додаткових функцій системи, що доступні лише за умови платної підписки (або внутрішніх досягнень) користувачів. До таких функцій можна віднести відображення ЗОК інших користувачів (для прямого контактування), розширені засоби фільтрації та групування. Також можливе встановлення обмеження щодо кількості опублікованих одиниць контенту, коментарів, груп (плейлистів) тощо.

Архітектура

ПЗ має бути спроектоване за архітектурою PWA (Progressive Web Application) з чітко відокремленими розподіленими вузлами системи:

Москит - візуальна модель інтерфейсу користувача із зазначенням зв'язків і переходів між різними частинами інтерфейсу.

Backend - серверне програмне забезпечення, що забезпечує збереження та оброблення поточних даних, взаємодію з базою даних. Надає програмний інтерфейс взаємодії (API) з іншими вузлами.

Web App - додаток, доступний з веб-простору (сайт), що забезпечує візуальний інтерфейс користувачів усіх категорій

Mobile App (опціонально) - додаток, що встановлюється на мобільних пристроях (смартфон, планшет). Дана частина виконується за наявності у команді проєкту трьох і більше програмістів.

.

Усі модулі ПЗ (окрім візуальної моделі Москит) повинні обов'язково включати до свого складу засоби

- тестування функціональності та модульні тести,
- розгортання на новому пристрої (інсталяції) із встановленням початкових таблиць у БД та кореневого користувача-адміністратора

Технології

Зберігання вихідних кодів ПЗ повинно здійснюватись за допомогою системи контролю версій (вибір конкретної системи узгоджується групою проєкту).

Структура файлів та каталогів повинна відбивати архітектуру ПЗ, на зразок:

- root
 - Backend
 - Web
 - Mobile
 - Москит (вихідні файли або посилання на зовнішній проєкт)

Для кожного з модулів рекомендується (з можливістю часткової зміни) наступні технології створення:

Backend - PHP технології із виділеним або віртуальним хостингом,

Web - JS, HTML, CSS, або JS фреймворки, зокрема ReactJS, AngularJS

Mobile - Java, або Kotlin, або Flutter

Москит - визначається ментором з дизайну

Команда проєкту

Для виконання поставлених завдань передбачається робота команди розробників окремо за кожним з модулів проєкту.

Мінімальна команда проєкту - два учасники, у такому складі один учасник виконує модуль Backend та програмну частину Web, другий - модель Mockup та розміткову частину Web.

Рекомендована команда проєкту - три учасники, у такій команді кожен з учасників відповідає за одну з частин: Mockup, Backend, Web.

Дозволяється включення до команди більшої кількості учасників, у такому разі модуль Mobile App набуває обов'язкової реалізації. Також модуль Mockup може бути поділений на окремі моделі для Web та Mobile частин.

Звітність

За результатами виконання завдань із створення ПЗ кожен з учасників команди проєкту складає власний (індивідуальний) звіт, що складається з практичної частини та автореферату. Практична частина є прямим результатом роботи і подається у вигляді посилання на репозиторій (чи інший URL) або у вигляді архіву на довільному носіїві даних. Вихідний код повинен оформлятися у відповідності до загальноприйнятих стандартів, на зразок стандартів GNU <http://www.gnu.org/prep/standards/>, у тому числі з дотриманням вимог до коментування та документування коду.

Автореферат містить описову частину, у якій розкриваються основні концепти проєкту, деталізуються завдання до виконавця, встановлюється актуальність роботи, її об'єкт та предмет, обґрунтовується вибір технологій та відзначається шлях виконання та ступінь досягнення поставлених задач, посилання на джерела інформації.

РОЗДІЛ 1

ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМИ

У відповідності до технічного завдання (ТЗ) система була спроектована за архітектурою «Progressive Web Application» (PWA). У складі системи було умовно відокремлено наступні архітектурні частини:

- «Mockup»,
- «Backend»,
- «Frontend».

Частина «Mockup» являє собою набір моделей та описів поведінки користувача. Модель подає інтерфейси взаємодії користувачів різного типу з інформаційною системою, траєкторії рухів та переходів між різними інтерфейсами. При створенні інтерфейсів застосовано принципи розумної навігації, включно із підходами UI/UX, висновки колористики та ергономіки для різних видів пристроїв: смартфони, планшети, лептопи, настільні комп'ютери. Макети цієї роботи можна переглянути за даним посиланням: <https://www.figma.com/file/NWgcgIIItsZgB1ugP0pxNNm/Design-FunTab?node-id=0%3A1>.

Частина «Backend» реалізована як централізований вузол збереження та оброблення даних, що створюються та модифікуються протягом життєвого циклу інформаційної системи. Дана частина включає у себе базу даних проєкту та програмний інтерфейс (API) доступу до основних її функцій, зокрема, керування контентом, авторизації, узгодження профілів користувачів різного типу та/або різних пристроїв. Частина «Backend» виконана на базі програмного забезпечення PHP та СУБД «MySQL».

Клієнтська частина системи «Front-end», призначена для надання користувачеві візуального інтерфейсу через який він отримує доступ до функцій керування потоковим контентом розробленої інформаційної системи. Також до функцій даної частини належить збереження локальних даних для можливості використання системи при повторному вході без повторної авторизації. Вихідні коди та ресурси даної частини можна переглянути в репозиторії за даним посиланням: <https://VladislavaVG@bitbucket.org/VladislavaVG/funtab.git>.

РОЗДІЛ 2

РЕАЛІЗАЦІЯ СЕРВЕРНОЇ ЧАСТИНИ

Серверна частина інформаційної системи реалізована з використанням наступних технологій:

- Засоби розроблення: мова програмування PHP
- Додаткове ПЗ та засоби: Google Cloud Directory Sync для засобів синхронізування акаунтів (з використанням токен OAuth)
- У складі серверної частини (desktop додатку) передбачено та реалізовано наступні функції:
 - Авторизація користувача, реєстрація нових користувачів
 - Подання на оновлення даних
 - Керування особистими та поточними даними (кабінетом)
 - Активація і перевірка даних через власну поштову скриньку

Початок роботи користувача із додатком супроводжується його реєстрацією як учасника інформаційної системи. При реєстрації користувач обов'язково надає відомості про електронну пошту, через яку буде здійснюватись оповіщення користувача щодо актуальних питань, а також про зміну активності у його особистому просторі. Попередньо до реєстрації введені дані проходять перевірку встановлена змінна значенням, відмінним від null та на пустоту.

```
if( isset( $date->firstname ) &&  
    isset( $date->lastname ) &&  
    isset( $date->username ) &&  
    isset( $date->email ) &&  
    isset( $date->password )) {  
}  
  
if ( !empty( $date->firstname ) &&  
    !empty( $date->lastname ) &&  
    !empty( $date->username ) &&  
    !empty( $date->email ) &&  
    !empty( $date->password )){  
}
```

Для більшого захисту системі від даних з пробелами та символами на початку чи в кінці, тегів HTML та PHP, екранування символів чи спеціальних символів

```
function clean_var( $value ){  
    $value = trim( $value ) ;
```

```
$value = strip_tags( $value ) ;  
$value = stripslashes( $value ) ;  
$value = htmlspecialchars( $value ) ;  
return $value ;  
}
```

Авторизація користувача здійснюється за допомогою персонального секрету (логіну та паролю), введеного на етапі реєстрації.

При реєстрації чи авторизації здійснюється комунікація з частиною «Backend» у результаті чого авторизація підтверджується або спростовується.

Обмін даними з серверною частиною реалізовано за допомогою окремих програмних засобів.

```
function is_get_request(): bool {  
    return strtoupper( $_SERVER[ 'REQUEST_METHOD' ] ) === 'GET' ;  
}  
  
function is_post_request(): bool {  
    return strtoupper( $_SERVER[ 'REQUEST_METHOD' ] ) === 'POST' ;  
}  
  
function is_put_request(): bool {  
    return strtoupper( $_SERVER[ 'REQUEST_METHOD' ] ) === 'PUT' ;  
}  
  
function is_delete_request(): bool {  
    return strtoupper( $_SERVER[ 'REQUEST_METHOD' ] ) === 'DELETE' ;  
}  
  
function is_option_request(): bool {  
    return strtoupper( $_SERVER[ 'REQUEST_METHOD' ] ) === 'OPTION' ;  
}
```

Також було додано функцію налаштування CORS механізму що додає можливість користувачу отримувати дозволи на доступ до вибраних ресурсів із сервера

```
function is_get_request(): bool {  
function HTTPHeader( $methods ) {
```

```

if (isset($_SERVER['HTTP_ORIGIN'])) {
    $temp = $_SERVER['HTTP_ORIGIN'] ? : "192.168.9.108:3000" ;
    header("Access-Control-Allow-Origin: {$temp}");
    header('Access-Control-Allow-Credentials: true');
    header('Access-Control-Max-Age: 86400');
}
if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS') {
    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))
        header("Access-Control-Allow-Methods: {$methods} ");
    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))
        header("Access-Control-Allow-Headers:
{$_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");
    exit(0);
}
}

```

У якості прикладу далі наведено приклад передавання даних методом POST, що широко вживається у різних частинах проєкту

```

<?php
include_once '../header/HTTPheaders.php' ;
HTTPHeader("POST") ;

include_once '../config/database.php' ;
include_once '../object/user.php' ;
include_once '../object/channel.php' ;
include_once '../object/tagsStandart.php' ;

include_once '../channel/create_default_channel.php' ;

include_once '../libs/http_requests.php' ;
include_once '../libs/salt.php' ;
include_once '../libs/default_icon.php' ;
include_once '../send/send_email.php' ;

include_once '../geo/SxGeo.php';

include_once "../config/core.php" ;
include_once "../libs/php-jwt-master/src/BeforeValidException.php" ;
include_once "../libs/php-jwt-master/src/ExpiredException.php" ;
include_once "../libs/php-jwt-master/src/SignatureInvalidException.php" ;

```

```

include_once "../libs/php-jwt-master/src/JWT.php" ;
use \Firebase\JWT\JWT ;

$databse = new Database() ;
$db = $databse->getConnection() ;

$user = new User( $db ) ;
$channel = new Channel( $db ) ;
$tagStandart = new TagStandart( $db ) ;
$jwtClass = new JWT() ;
$SxGeo = new SxGeo( './geo/SxGeoCity.dat', SXGEO_BATCH |
SXGEO_MEMORY ) ;

$data = json_decode( file_get_contents( "php://input" ) ) ;

$geo = $SxGeo->getCityFull($_SERVER['REMOTE_ADDR']) ;

if( is_post_request() && isset( $data->email ) && isset( $data->userName ) ) {

    $user->email = $data->email ;
    $email_exists = $user->emailExists() ;

    if( $email_exists && $user->active_code == true ){
        http_response_code(412) ;
        echo json_encode( array( "code" => 412 ),
JSON_UNESCAPED_UNICODE) ;
    }else{
        if( $email_exists )
            $user->delete() ;

        if( isset( $data->password ) || isset( $data->googleId ) ) {

            $user->user_name = $data->userName ;
            $user->password = isset( $data->googleId ) ? null : $data->password ;
            $user->first_name = isset($data->firstName) ?$data->firstName: null ;
            $user->last_name = isset($data->lastName) ?$data->lastName: null ;
            $user->google_id = isset($data->googleId) ?$data->googleId: null ;
            $user->country = $geo["country"]["name_en"] ;
            $user->date_create = date("Y-m-d") ;

            if( isset($data->imageUrl) )

```

```

        $user->image_url = $data->imageUrl ;
    else
        $user->icon = $data_icon[ rand( 0, count( $data_icon ) - 1 ) ] ;

    if ( !empty( $data->userName ) && !empty( $data->email ) &&
$user->create() ) {

        if( isset( $data->googleId ) ) {
            $user->read() ;

            $arr_data = crateDefaultData( $iss, $aud, $iat, $nbf, $key, $user,
$channel, $tagStandart, $jwtClass );

            echo json_encode($arr_data) ;
            http_response_code( 200 ) ;
            echo json_encode( array( "code" => 200 ),
JSON_UNESCAPED_UNICODE);
        }else{
            sendCodeEmail($user->email, $user->user_name, $user->code);

            http_response_code(200);
            echo json_encode(array("code" => 200),
JSON_UNESCAPED_UNICODE);
        }
    }
    else {
        http_response_code(503);
        echo json_encode( array( "code" => 503),
JSON_UNESCAPED_UNICODE);
    }
    }else{
        http_response_code(400);
        echo json_encode( array( "code" => 400 ),
JSON_UNESCAPED_UNICODE);
    }
}
}
else {
    http_response_code(400);
    echo json_encode( array( "code" => 400 ), JSON_UNESCAPED_UNICODE);
}
}
?>

```

Основну частину взаємодії користувача із додатком реалізовано за допомогою роботи з контентом. Користувач має змогу визначити потребу своєї цікавості, що становить його інтерес, за допомогою категорії тегу або назви.

Для забезпечення правильності та наявності поштової скриньки генерується випадковий код розміром у 8 числових знаків

```
function generateCode( $length = 8 ) {  
    $code = "" ;  
  
    for($i = 0; $i < $length; $i++) {  
        if( $i == 0 )  
            $code .= chr(rand(49, 57));  
        else  
            $code .= chr(rand(48, 57));  
    }  
    return $code;  
}
```

Зберігання паролів у базі даних здійснюється за допомогою хешування SHA512 з додаванням до нього солі

```
function generateSalt( $length = 10 ) {  
    $salt = "" ;  
    for($i = 0; $i < $length; $i++) {  
        $salt .= chr(rand(33, 126));  
    }  
    return $salt;  
}  
  
$salt = generateSalt() ;  
$password = hash( 'sha512', $salt.trim( $password ) ) ;
```

Який відправляється на поштову скриньку користувача яку він вказав при реєстрації за допомогою метода sendCodeEmail() який приймає параметри пошти користувача яку ввів користувач, його ім'я, прізвище та згенерований код

```
function sendCodeEmail($to_, $lastname_, $firstname_, $code_){  
    $to = $to_;
```

```

$subject = 'Your confirmation code for logging into your Funtab account';
$message = 'body message' ;

$headers = 'MIME-Version: 1.0' . "\r\n";
$headers .= 'Content-type: text/html; charset=UTF-8' . "\r\n";
$headers .= "From: FunTab <funtab@gmail.com>" . "\r\n";
mail($to, $subject, $message, $headers);
}

```

Також для аутентифікації та передачі даних був використаний JWT Token. JWT Token створюється сервером, підписується секретним ключем і передаються клієнту, який надалі використовує цей токен для підтвердження своєї особистості.

```

$token = array(
    "iss" => $iss,
    "aud" => $aud,
    "iat" => $iat,
    "nbf" => $nbf,
    "data" => array(
        "id" => $user->id,
        "firstName" => $user->first_name,
        "lastName" => $user->last_name,
        "userName" => $user->user_name,
        "email" => $user->email,
        "country" => $user->country,
        "imageUrl" => $user->image_url,
        "icon" => $user->icon,
        "googleId" => $user->google_id,
        "googleAccount" => $user->google_account,
        "dateCreate" => $user->date_create
    )
);

$jwt = $jwtClass::encode( $token, $key, "HS256" );

```

Із розвитком Web-технологій і спеціального програмного інструментарію для обслуговування Web-сайтів широкого розповсюдження набув клас масштабних Інтернет-ресурсів, які називають Web-порталами. Web-портал – це Web-сайт, що надає велику кількість послуг та/або надає доступ до великої кількості інформаційних ресурсів широкій аудиторії користувачів. До

Web-порталів відносяться корпоративні, державні, портали новин, розважальні та ін.

На кодї нижче наведено приклад створення та завантаження контенту та всіма його налаштуваннями та лімітами.

```
include_once '../header/HTTPheaders.php' ;
HTTPHeader("POST") ;

include_once "../libs/http_requests.php" ;
include_once "../libs/uploads.php" ;
include_once "../libs/information_volume.php" ;

$__DIR__ = "/home/am455914/funtab.com.ua/www/ServerData" ;
$uploadDir = "/Videos/" ;
$name_param = 'file' ;

if( is_post_request() ) {
    uploadFile($FILES[$name_param], getAmountByte( 300, 'mb' ), $__DIR__ .
$uploadDir ) ;
}

function getAmountByte( $num, $unit = 'byte') {
    switch ( mb_strtolower( $unit ) ) {
        case 'byte': case 'Byte' : return $num ; break ;
        case 'kb' : case 'Kb' : return pow(2,10)*$num ; break ;
        case 'mb' : case 'Mb' : return pow(2,20)*$num ; break ;
        case 'gb' : case 'Gb' : return pow(2,30)*$num ; break ;
        case 'tb' : case 'Tb' : return pow(2,40)*$num ; break ;
        case 'pt' : case 'Pb' : return pow(2,50)*$num ; break ;
        case 'et' : case 'Et' : return pow(2,60)*$num ; break ;
        case 'zt' : case 'Zt' : return pow(2,70)*$num ; break ;
        case 'yt' : case 'Yt' : return pow(2,80)*$num ; break ;
        default : echo 'there is no such unit of measure.' ; break ;
    }
}

function uploadFile( $FILES, $limit, $__DIR__ ){

    $extension = strstr($FILES['name'], '.');
    $error_code = $FILES['error'] ;
    $file_tmp_name = $FILES['tmp_name'] ;
    $type = $FILES['type'] ;
```

```

#echo json_encode( $FILES ) ;

if( $error_code != null || !is_uploaded_file( $file_tmp_name ) ) {

    $error_messages = [
        UPLOAD_ERR_INI_SIZE => 'The file size has exceeded the
upload_max_filesize value in the PHP configuration.',
        UPLOAD_ERR_FORM_SIZE => 'The size of the uploaded file has
exceeded the MAX_FILE_SIZE value in the HTML form.',
        UPLOAD_ERR_PARTIAL => 'The download file was only partially
received.',
        UPLOAD_ERR_NO_FILE => 'The file was not loaded.',
        UPLOAD_ERR_NO_TMP_DIR => 'The temporary folder is missing.',
        UPLOAD_ERR_CANT_WRITE => 'Failed to write file to disk.',
        UPLOAD_ERR_EXTENSION => 'The PHP extension has stopped
downloading the file.',
    ];
    $unknown_message = 'An unknown error occurred while uploading the file.' ;

    $output_message = isset( $error_messages[ $error_code ] ) ?
$error_messages[ $error_code ] : $unknown_message ;

    die( $output_message ) ;
}

$mime = stristr( $type , '/' , true ) ;
#echo $mime . "\n" ;
$limit_bytes = $limit ;

if( filesize( $file_tmp_name ) > $limit_bytes )
    echo json_encode( array( "code" => 205 ) ) ;

$uniqid_t = uniqid( " , true ) ;
$unimd_name = md5( $uniqid_t ) . $extension ;

if ( move_uploaded_file( $file_tmp_name, $__DIR__ . $unimd_name ) ) {
    if( $mime == 'image' )
        return $unimd_name ;
    echo json_encode(
        array(
            "code" => 200,
            "name" => $unimd_name

```

```

        ));
    } else {
        echo json_encode(
            array(
                "code" => 400
            ));
    }
}

```

Та додавання інформації про цей контент

```

<?php
include_once '../header/HTTPheaders.php' ;
HTTPHeader("POST") ;

include_once '../config/database.php' ;
include_once '../object/video.php' ;
include_once '../object/tag.php' ;
include_once '../object/videoTags.php' ;
include_once '../object/category.php' ;

include_once '../libs/http_requests.php' ;
include_once "../libs/getID3-master/getid3/getid3.php" ;

include_once "../libs/uploads.php" ;
include_once "../libs/information_volume.php" ;

$__DIR__ = "/home/am455914/funtab.com.ua/www/ServerData" ;
$uploadDirPoster = "/Posters/" ;
$uploadDirVideo = "/Videos/" ;
$name_param = 'poster' ;

$dbase = new Database() ;
$db = $dbase->getConnection() ;

$video = new Video( $db ) ;
$tag_db = new Tag( $db ) ;
$video_tags = new VideoTags( $db ) ;
$category = new Category( $db ) ;

```

```

if( is_post_request() ) {

    $data = json_decode( file_get_contents( "php://input" ) );

    $filename = $__DIR__ . $uploadDirVideo . $_POST['name'];

    $getID3 = new getID3 ;
    $file = $getID3->analyze( $filename );

    $durationStr = formatStrDuration( $file["playtime_string"] );

    $category->name = $_POST['category'];

    $video->name = $_POST['name'];
    $video->title = $_POST['title'];
    $video->description = $_POST['description'];

    if( $_FILES[ $name_param ]['size'] > 0 ){
        $video->poster = uploadFile($_FILES[$name_param], getAmountByte( 20,
'mb' ), $__DIR__ . $uploadDirPoster );
    }

    $video->visibility = $_POST['visibility'];
    $video->allow_comments = $_POST['allowComments'];
    $video->allow_sharing = $_POST['allowSharing'];
    $video->age_limit = $_POST['ageLimit'];
    $video->extension = $file["fileformat"];
    $video->duration_str = $durationStr;
    $video->duration = round( $file["playtime_seconds"] );
    $video->date_create = date("Y-m-d H:i:s");
    $video->type = strstr( $file["mime_type"], '/', true );
    $video->category_id = $category->getId();
    $video->channel_id = $_POST['channelId'];

    $array_tags = explode( " ", $_POST['tags'] );

    foreach ( $array_tags as $tag ){
        #echo json_encode( $tag ) . "\n";
    }

    if( $video->create() ) {

```

```

    $arr_tag_id = array() ;
    foreach ( $array_tags as $t ){
        $tag_db->name = $t ;
        $tag_db->create() ;
        array_push( $arr_tag_id, $tag_db->getId() ) ;
    }
    foreach ( $arr_tag_id as $tag_id ){
        $video->name = $_POST['name'] ;
        $video_tags->video_id = $video->getId() ;
        $video_tags->tag_id = $tag_id ;
        $video_tags->create() ;
    }

    http_response_code( 201 ) ;

    echo json_encode( array(
        "code" => 200
    ), JSON_UNESCAPED_UNICODE ) ;
} else {

    http_response_code( 503 ) ;

    echo json_encode( array(
        "code" => 503
    ), JSON_UNESCAPED_UNICODE ) ;
}

}

?>

```

Пошук інформації відбувається ще секунднo тому при створенні пошукової системи треба врахувати параметри пошуку і сам алгоритм. Пошукова система що надає користувачеві можливість швидкого доступу до необхідної інформації з допомогою пошуку у великій колекції доступних даних. Одне з найвідоміших застосувань пошукових систем – веб-сервіси. Поєднання різновидів контенту та його додаткові параметри.

```
<?php
```

```

#search
include_once "../header/HTTPheaders.php" ;
HTTPHeader("GET") ;

include_once '../config/database.php' ;
include_once '../config/selectQuery.php' ;

include_once '../object/video.php' ;
include_once '../object/channel.php' ;
include_once '../object/playlist.php' ;

include_once '../libs/http_requests.php' ;

include_once '../libs/information_volume.php' ;

$database = new Database() ;
$db = $database->getConnection() ;

$video = new Video($db) ;

$channel = new Channel($db) ;

$playlist = new Playlist($db) ;

$keywords = isset( $_GET["searchText"] ) ? $_GET["searchText"] : "" ;

$page = isset( $_GET["page"] ) ? $_GET["page"] : 1 ;
$records_per_page = isset( $_GET["count"] ) ? $_GET["count"] : 5 ;
$from_record_num = ( $records_per_page * $page ) - $records_per_page ;

$filtration_date = isset( $_GET["date"] ) ? $_GET["date"] : null ;
$filtration_duration = isset( $_GET["duration"] ) ? $_GET["duration"] : null ;
$filtration_type = isset( $_GET["type"] ) ? $_GET["type"] : null ;

//echo $filtration_type ;

switch ( $filtration_type ) {
    case 'Video':
    case 'Movie':
        searchVideoAndMovie($keywords, $filtration_date, $filtration_duration,
$filtration_type, $from_record_num, $records_per_page, $video) ;
        break;
    case 'Channel' :

```

```

        break ;
    case 'Playlist' :
        break ;
    default:
        if( $filtration_duration != null || $filtration_date != null ){
            searchVideoAndMovie($keywords, $filtration_date, $filtration_duration,
$filtration_type, $from_record_num, $records_per_page, $video) ;
        }else{
            $query = getSearch($keywords,$from_record_num, $records_per_page) ;

            $stmt = $db->prepare( $query );

            if( $stmt->execute() )

            $num = $stmt->rowCount() ;

            echo $num . "\n" ;

            if ( $num > 0 ) {
                $search_arr = array() ;
                $search_arr["data"] = array() ;

                while ( $row = $stmt->fetch( PDO::FETCH_ASSOC ) ) {
                    extract( $row ) ;
                    switch($row["Type"]){
                        case 'video' :

$video_getOneInfoChannelVideo =
getOneInfoChannelVideo($row["Id"]) ;

$video_stmt = $stmt_v =
$video->readAllPaging(0,1000,$query_getOneInfoChannelVideo) ;
                $row_v = $stmt_v->fetch( PDO::FETCH_ASSOC ) ;
                extract($row_v) ;
                $video_item = array(
                    "channelId" => (int)$row_v["ChId"],
                    "channelName" => $row_v["ChName"],
                    "imageUrl" => $row_v["ChImageUrl"],
                    "icon" => $row_v["ChIcon"],
                    "categories" => $row_v["Category"],
                    "type" => $row_v["Type"],
                    "id" => (int)$row_v["Id"],
                    "name" => $row_v["Name"],
                    "title" => $row_v["Title"],
                    "description" => $row_v["Description"],

```

```

        "poster" => $row_v["Poster"],
        "totalLikesCount" => $row_v["TotalLikesCount"],
        "totalDislikesCount" => $row_v["TotalDislikesCount"],
        "totalViewCount" => $row_v["TotalViewCount"],
        "extension" => $row_v["Extension"],
        "durationStr" => $row_v["DurationStr"],
        "duration" => (int)$row_v["Duration"],
        "dateCreate" => $row_v["DateCreate"],

        "isBlue" => rand(0,1) == 1 ? true : false
    );
    array_push( $search_arr["data"], $video_item );
    break ;
case 'channel' :
    $channel->id = $row["Id"] ;
    $stmt_c = $channel->readOne() ;
    $num_c = $stmt_c->rowCount() ;
    $row_c = $stmt_c->fetch( PDO::FETCH_ASSOC ) ;
    $channel_item = array(
        "id" => $row_c["Id"],
        "name" => $row_c["Name"],
        "imageUrl" => $row_c["ImageUrl"],
        "icon" => $row_c["Icon"],
        "type" => $row_c["Type"]
    );
    array_push( $search_arr["data"], $channel_item );
    break ;
case 'playlist' :
    $playlist->id = (int)$row["Id"] ;
    $stmt_pl = $playlist->readOnePlayList() ;
    $row_pl = $stmt_pl->fetch( PDO::FETCH_ASSOC ) ;
    extract( $row_pl ) ;

    $playlist_item = array(
        "id" => (int)$row_pl["Id"],
        "name" => $row_pl["Name"],
        "description" => $row_pl["Description"],
        "visibility" => $row_pl["Visibility"] > 0 ? true : false,
        "type" => $row_pl["Type"],
        "countVideo" => "",
        "videoName" => "",
        "duartion" => "",
    );

```

```

        $playlist->id = $row["Id"] ;
        $stmt_vpl = $playlist->infoPlayListAboutVideo() ;
        $num_vpl = $stmt_vpl->rowCount() ;
        if( $num_vpl > 0 ){
            $row_vpl = $stmt_vpl->fetch( PDO::FETCH_ASSOC ) ;
            extract( $row_vpl ) ;

            $playlist_item["countVideo"] = (int)$row_vpl["Count"] > 0 ?
(int)$row_vpl["Count"] : 0 ;
            $playlist_item["videoName"] = $row_vpl["Name"] != "" ?
$row_vpl["Name"] : null ;
            $playlist_item["duartion"] = $row_vpl["Duration"] > 0 ?
$row_vpl["Duration"] : null ;
        }

        array_push( $search_arr["data"], $playlist_item ) ;
        break ;
    }
}
echo json_encode($search_arr);
}
}
break;
}

function searchVideoAndMovie($keywords, $filtration_date, $filtration_duration,
$filtration_type, $from_record_num, $records_per_page, $video){
    $conv_date = convertDateStrInDateType($filtration_date);
    $conv_dur = convertDurationStrInSecond($filtration_duration);

    if( $filtration_type == 'Movie' ){
        $query = getVideoSearch($keywords, $conv_date, $conv_dur,
$filtration_type) ;
    }
    else{
        $query = getVideoSearch($keywords, $conv_date, $conv_dur ) ;
    }
    $stmt = $video->readAllPaging( $from_record_num, $records_per_page, $query
);
    $num = $stmt->rowCount() ;

    if ( $num > 0 ) {
        $videos_arr = array() ;
    }
}

```

```

$videos_arr["data"] = array() ;

while ( $row = $stmt->fetch( PDO::FETCH_ASSOC ) ) {
    extract( $row ) ;

    $video_item = array(
        "channelId" => (int)$row["ChId"],
        "channelName" => $row["ChName"],
        "imageUrl" => $row["ChImageUrl"],
        "icon" => $row["ChIcon"],
        "categories" => $row["Category"],
        "type" => $row["Type"],
        "id" => (int)$row["Id"],
        "name" => $row["Name"],
        "title" => $row["Title"],
        "description" => $row["Description"],
        "poster" => $row["Poster"],
        "totalLikesCount" => $row["TotalLikesCount"],
        "totalDislikesCount" => $row["TotalDislikesCount"],
        "totalViewCount" => $row["TotalViewCount"],
        "extension" => $row["Extension"],
        "durationStr" => $row["DurationStr"],
        "duration" => (int)$row["Duration"],
        "dateCreate" => $row["DateCreate"],

        "isBlue" => rand(0,1) == 1 ? true : false
    );

    array_push( $videos_arr["data"], $video_item ) ;
}
http_response_code(200);
echo json_encode($videos_arr);
}
}

die() ;

echo json_encode( $supported_date['Today'] ) . "\n" ;
echo json_encode( $supported_duration['>= 1 hour'] ) . "\n" ;

$stmt = $video->search( $keywords, $from_record_num, $records_per_page ) ;
$num = $stmt->rowCount() ;

```

```

echo "num : " . $num . "\n" ;

if ( $num > 0 ) {

    $videos_arr = array() ;
    $videos_arr["data"] = array() ;

    while ( $row = $stmt->fetch( PDO::FETCH_ASSOC ) ) {
        extract( $row ) ;

        $video_item = array(
            "channelId" => $row["ChId"],
            "channelName" => $row["ChName"],
            "channelImageUrl" => $row["ChImageUrl"],
            "categories" => $row["Categories"],
            "type" => $row["Video"],
            "id" => $row["Id"],
            "name" => $row["Name"],
            "title" => $row["Title"],
            "description" => $row["Description"],
            "poster" => $row["Poster"],
            "totalLikesCount" => $row["TotalLikesCount"],
            "totalDislikesCount" => $row["TotalDislikesCount"],
            "totalViewCount" => $row["TotalViewCount"],
            "extension" => $row["Extension"],
            "duration" => $row["Duration"],
            "dateCreate" => $row["DateCreate"]
        );

        array_push( $videos_arr["data"], $video_item ) ;
    }
    http_response_code(200);

    echo json_encode($videos_arr);
}

else {
    http_response_code(404);

    echo json_encode( array(
        "message" => "Content not found"
    ), JSON_UNESCAPED_UNICODE);
}

```

```
}  
?>
```

```
function getVideoSearch( $word, $date = null, $duration = null, $typeCategory =  
null ){  
    $query = "WHERE (vid.Name LIKE '%{$word}%'  
        OR vid.Title LIKE '%{$word}%'  
        OR vid.Description LIKE '%{$word}%'  
        OR cat.Name LIKE '%{$word}%')" ;  
  
    if($date != null && $duration != null && $typeCategory != null)  
        return $query . " AND vid.DateCreate > CAST('{ $date}' AS DATETIME)  
            AND vid.Duration {$duration}  
            AND vid.CategoryId = (SELECT Id  
                FROM Categories  
                WHERE Name LIKE '%{$typeCategory}%')" ;  
    else if($date != null && $duration != null)  
        return $query . " AND vid.DateCreate > CAST('{ $date}' AS DATETIME)  
            AND vid.Duration {$duration}" ;  
    else if($date != null && $typeCategory != null)  
        return $query . " AND vid.DateCreate > CAST('{ $date}' AS DATETIME)  
            AND vid.CategoryId = (SELECT Id  
                FROM Categories  
                WHERE Name LIKE '%{$typeCategory}%')" ;  
    else if($duration != null && $typeCategory != null)  
        return $query . " AND vid.Duration {$duration}  
            AND vid.CategoryId = (SELECT Id  
                FROM Categories  
                WHERE Name LIKE '%{$typeCategory}%')" ;  
    else if($typeCategory != null)  
        return $query . " AND vid.CategoryId = (SELECT Id  
            FROM Categories  
            WHERE Name LIKE '%{$typeCategory}%')" ;  
    else if($duration != null)  
        return $query . " AND vid.Duration {$duration}" ;  
    else if($date != null)  
        return $query . " AND vid.DateCreate > CAST('{ $date}' AS DATETIME)" ;  
    else  
        return $query;  
}
```

РОЗДІЛ 3

РЕЗУЛЬТАТИ ВИПРОБУВАНЬ

Результатом перевірки програмного коду серверної частини являється платформа Postman що дозволяє розробникам проектувати, створювати, тестувати та повторювати свої API.

Початок роботи нового користувача з додатком починається з реєстрації користувача у системі. Під час реєстрації здійснюється перевірка засобів комунікації - на електронну пошту надсилається код підтвердження, який має бути введений для завершення етапу реєстрації.

Якщо користувач раніше проходив реєстрацію, то він може увійти до системи за допомогою персональних даних, зазначених при реєстрації.

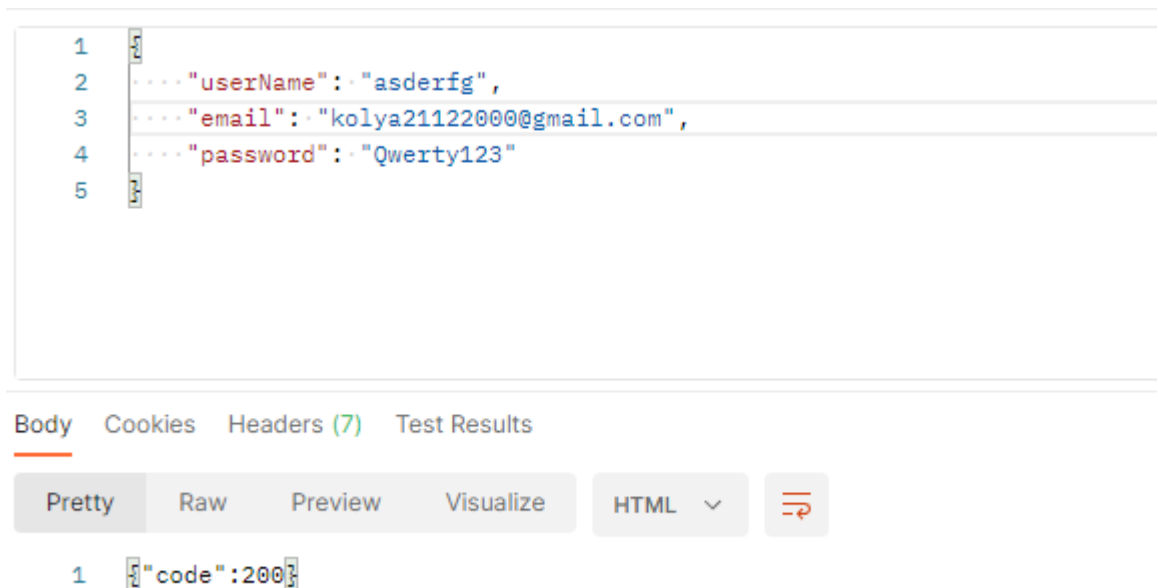


Рисунок 1 - Реєстрація користувача



Після входу до системи користувач отримує доступ до основних функцій системи. Однією з таких функцій є передплата на канал, оцінювання та викладання свого контенту, створення свого каналу и т.д.

Також користувач може встановити свій аккаунт та даних перейшовши на вкладку забув пароль, він вказує вже зареєстровану поштову скриньку та отримує ссылку на відновлення паролю від свого аккаунту.



FUNTAB ID: PASSWORD RESET

To reset your password, please click on the link or copy it:

<https://funtab.com.ua>

Sincerely,
Team Funtab

На рис. 4 наведено повідомлення з ссилкою на оновлення паролю.

У підсумку, всі функції серверної частини, що були регламентовані технічним завданням було реалізовано та випробувано. Поточна версія додатку, що реалізує серверної частину, доступна у особистому кабінеті https://adm.tools/hosting/?account_id

ВИСНОВКИ

Проведено огляд інформаційних систем, що враховують створення розподіленої системи управління контентом, який має потоковий зміст, виявлено, що суттєве поширення мають системи, які пов'язані із наданням потокового контенту. Саме для таких систем відомості про повідомлення та надання контенту мають основне значення. У якості прикладу вибрано для реалізації систему відео

Визначено програмні засоби (інтерфейси API), які дозволяють отримувати дані про користувача, дані про локацію користувача, перевагу надано бібліотеці getID3. Також для взаємозв'язку між клієнтом і сервером за допомогою створення JWT Token бібліотекою php-jwt.

Спроектовано, реалізовано та випробувано систему узгодження взаємодії користувачів з урахуванням геопозиціонування її учасників. Система складена за архітектурою «Progressive Web Application» (PWA) та складається з трьох частин, розміщених за адресами:

- «Mockup»
<https://www.figma.com/file/NWgcgIIItsZgB1ugP0pxNNm/Design-FunTab?node-id=0%3A1>
- «Backend», <https://funtab.com.ua>
- «Frontend» https://adm.tools/hosting/?account_id

У систему впроваджено засоби покращення інформаційної безпеки. Також вжито засобів підтвердження персональних даних користувача, зокрема, електронної пошти. Це ускладнює зловмисникам вживання підроблених (невласних) даних для дискредитації системи.

У систему впроваджено засоби покращення інформаційної безпеки: AES-128 – це сучасний стандарт шифрування даних. Він не має собі рівних за рівнем безпеки та захисту, який він пропонує: шифрування із симетричним ключем, блоковий шифр. Він називається блоковим, тому що цей тип шифру поділяє інформацію, яка повинна бути зашифрована (відомий як відкритий текст) на розділи, звані блоками. Щоб бути більш конкретним, AES використовує розмір блоку 128 біт. Візьмемо, наприклад, додаток. Якщо він використовує

256-бітний AES замість AES 128, він буде потрібний більше обчислювальної потужності. Практичний ефект в тому, що це потрібно більше сировинної потужності від акумулятора, тому ваш телефон помре швидше. Таким чином, при використанні 256-бітного шифрування AES золотий стандарт це просто неможливо для повсякденного використання.

Систему було випробувано шляхом інсталяції на фізичні пристрої та проведення тестових сеансів комунікації. Результати випробування підтвердили загальну працездатність створеної системи.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Стратегія розвитку інформаційного суспільства в Україні. Розпорядження Кабінету Міністрів України від 15 травня 2013 р. № 386-р. // Урядовий портал. URL: <https://www.kmu.gov.ua/npas/246420577>
2. Лященко А. А. Особливості реалізації стандартів доступу до баз геопросторових даних в середовищі універсальних СКБД. // Геоінформаційні технології у територіальному управлінні: матеріали II міжнар. наук.-практ. конф. Одеса: ОРІДУ НАДУ, 2015. с. 17-21 URL: http://www.oridu.odessa.ua/7/7/Book_new_2.pdf
3. Стратегія кібербезпеки України. Указ Президента України від 14 травня 2021 року // Урядовий кур'єр від 28.08.2021 — № 165. URL: <https://zakon.rada.gov.ua/laws/show/447/2021/print>
4. Решетило В.П., Федотова Ю.В. Невизначеність та ризик: співвідношення понять та специфіка прийняття рішень // Проблеми системного підходу в економіці, випуск № 3(77)-2, 2020. с. 149. URL: http://psae-jrnl.nau.in.ua/journal/3_77_2_2020_ukr/23.pdf
5. Ладичук Д.О. Проектування бази геопросторових даних. Навчальний посібник / Ладичук Д.О., Шапоринська Н.М. - Херсон. - 2020. - 128 с. ISBN: 978-966-289-344-1
6. Олександр Смичніков. Параноя або сучасні реалії: як зберегти конфіденційність у мережі. - ФБЮЧЕР МЕДІА. URL: <https://mind.ua/openmind/20183329-paranoya-abo-suchasni-realiyi-yak-zberegiti-konfidencijnist-u-merezhi> (дата звернення 30.05.2022)
7. Анісімов А.В. Інформаційні системи та бази даних: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. / Анісімов А.В., Кулябко П.П. – Київ. – 2017. – 110 с.
8. Величко О.М. Інтелектуальні інформаційні системи: структура і застосування. Підручник / Величко О.М., Гордієнко Т.Б. - Херсон. - 2022. - 728 с. ISBN: 978-966-289-552-0
9. Булгакова О.С. Інформатика: візуальне програмування. Навчальний посібник (стереотипне видання) / Булгакова О.С., Зосімов В.В., Броницька Н.А., Танкова Н.В. - Херсон. - 2020. - 312 с. ISBN: 978-966-289-039-6
10. Гоменюк С.І. Математичне моделювання геометричних об'єктів у паралельних комп'ютерних системах. Монографія / Гоменюк С.І., Чопоров С.В., Аль-Атамнех Б.Г. - Херсон. - 2019. - 112 с. ISBN: 978-966-916-714-9
11. Булгакова О.С. Методи та системи штучного інтелекту: теорія та практика. Навчальний посібник / Булгакова О.С., Зосімов В.В., Поздєєв В.О. - Херсон. - 2020. - 356 с. ISBN: 978-966-289-364-9
12. Покотилова В.І. Використання інформаційних технологій в теорії прийняття рішень. Навчальний посібник / Покотилова В.І., Фомішина В.М., Лугінін О.Є. - Херсон. - 2019. - 240 с. ISBN: 978-966-289-277-2

13. Величко О.М. Основи системного аналізу і прийняття оптимальних рішень. Підручник / Величко О.М., Гордієнко Т.Б. - Херсон. - 2021. - 672 с. ISBN: 978-966-289-553-7
14. Чемакіна О.В. Дизайн системи візуальної інформації. Навчальний посібник / Чемакіна О.В., Рубцов А.Л., Свірко В.О. - Херсон. - 2019. - 200 с. ISBN: 978-966-289-216-1
15. Шеховцов А.В. Комп'ютерні технології для дизайнерів. Навчальний посібник (стереотипне видання) / Шеховцов А.В., Полетаєва Г.Н., Крючковський Д.О., Бараненко Р.В. - Херсон. - 2019. - 318 с. ISBN: 978-966-2393-08-8
16. Чайковська М.П. Концептуально-методологічні засади управління маркетинговими ІТ-проєктами в умовах цифрових трансформацій. Монографія / Чайковська М.П. - Херсон. - 2021. - 370 с. ISBN: 978-966-289-537-7