



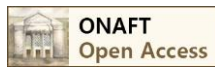
УДК 004.738.5:655.41(048):001.816

РОЗРОБКА ЕЛЕКТРОНОГО ДАЙДЖЕСТУ НАУКОВИХ ДОРОБКІВ

Волкова А. Ю.¹, Харахаш О. В.², Сакалюк О. Ю.³, Козуб О. О.⁴^{1,2,3,4}Одеська національна академія харчових технологій, Одеса, УкраїнаORCID: ¹ <http://orcid.org/0000-0002-4414-6592>, ² <http://orcid.org/0000-0002-7405-4517>, ³ <http://orcid.org/0000-0002-5051-518X>, ⁴ <http://orcid.org/0000-0002-0481-9203>E-mail: ¹ volkovanastasia.w@gmail.com, ² sash.harashash@gmail.com, ³ sakaliuk.olexiy@gmail.com,⁴ oksana.olegovna.kozub@gmail.com

Copyright © 2018 by author and the journal "Automation of technological and business - processes.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>DOI: <https://doi.org/10.15673/atbp.v12i1.1698>

Анотація. Метою проекту було створення програмного забезпечення, а саме веб-платформи, який надає можливість впорядкувати, нормалізувати та спростити пошук наукового доробку, що міститься серед періодичних наукових видань. Для досягнення поставленої мети, в розробці програмного продукту, було сформовано наступні задачі: проаналізувати основні проблеми предметної області; проаналізувати існуючі аналоги; обрати засоби реалізації програмного продукту; розробити програмний продукт.

Об'єктом дослідження виступив онлайн-сервіс, що впорядковує та спрощує пошук серед наукових публікацій. На даний момент, існує декілька аналогів, але головною відмінністю поміж ними є те, що дані програмні продукти не містять перелік наукових видань Одеської національної академії харчових технологій.

Для розробки серверної частини електронного дайджесту був обраний фреймворк Spring та мова програмування Kotlin. Була розроблена реляційна база даних у вільній об'єктно-реляційній СУБД PostgreSQL. Середовищами розробки були обрані IntelliJ IDEA та DataGrip. Для розробки клієнтської частини електронного дайджесту було використано середовище розробки PHPStorm з інтелектуальним редактором, який підтримує передові технології розробки та включає в себе підтримку баз даних, також володіє повним набором інструментів для розробки Інтернет ресурсу. Інтерфейс дайджесту було побудовано за допомогою HTML5/CSS3, для побудови інтерактивного та функціонального додатку було використано мову JavaScript, а також фреймворк Bootstrap для поліпшення розробки.

Abstract. The aim of the project was to create software, namely a web-based platform, which provides an opportunity to streamline, normalize and simplify the search for scientific content contained in periodicals. In order to achieve this goal, in the development of the software product, the following tasks were formed: to analyze the main problems of the subject area; analyze existing analogues; to choose the means of realization of the software product; to develop a software product.

The object of the study was an online service that streamlines and simplifies the search for scientific publications. At the moment, there are several analogues, but the main difference between them is that these software products do not contain a list of scientific publications of the Odessa National Academy of Food Technologies.

The Spring Framework and the Kotlin programming language were chosen to develop the server side of the electronic digest. A relational database was developed in the free PostgreSQL object-relational database. The development environments were chosen by IntelliJ IDEA and DataGrip. PHPStorm's development environment with a smart editor, which supports advanced development technologies and includes database support, also has a complete set of tools for developing an online resource, was used to develop the electronic digest client part. Digest interface was built using HTML5 / CSS3, JavaScript and Bootstrap framework were used to build the interactive and functional application.

Ключові слова: Web 2.0, Science 2.0 веб-ресурс, база даних, PostgreSQL, HTML5, CSS3, програмне забезпечення, наукові доробки, пошук інформації, науково-технічна бібліотека, розробка програмного продукту, діджиталізація

Keywords: Web 2.0, Science 2.0 web resource, database, PostgreSQL, HTML5, CSS3, software, scientific research, information search, scientific and technical library, software development, digitization

1. Теоретична складова

Інформаційні технології та автоматизація робочого процесу відіграють важливу роль у організації процесів моніторингу, аналізу та популяризації наукових здобутків, зокрема це відбувається за допомогою електронних



ресурсів. Для швидкого доступу та пошуку наукових доробків технології Web 2.0 надають такі переваги, як можливість глобального пошуку у Інтернет мережі, вільний дистанційний доступ до наукових праць, індивідуальні ресурси/платформи наукових видань та повідомлення про оновлення дослідницьких тем.

Кількість наукового контенту, який щорічно потрапляє до мережі налічує сотні тисяч одиниць, серед яких, принаймні декілька тисяча належить науковцям Одеської національної академії харчових технологій. З метою систематизації наукового контенту вчених ОНАХТ було прийнято рішення створити електронний дайджест.

В якості електронного дайджесту наукових публікацій/контенту буде виступати електронний ресурс, який надаватиме вільний доступ до інформації про наукові доробки. Основними етапами розробки електронного ресурсу є визначення структурних цілей, розробка інструментарію, вибір наукових видань, та розробка зручного та функціонального інтерфейсу [2, 3].

Таким чином, застосування Web 2.0 в рамках Science 2.0 є основою для зручної організації моніторингу наукових доробків, та інформації про них.

2. Результати проведенного дослідження

Програмне забезпечення (ПЗ) розроблене в рамках дослідження становить інформаційно-управляючу систему для публікацій науковців ОНАХТ. Розроблене ПЗ має трьохрівневу архітектуру, що складається з клієнта, сервера додатку та серверу бази даних. Клієнтська частина Інтернет ресурсу реалізує інтерфейс, відправляє запити до серверної частини та оброблює відповідь отриману зі сторони серверу. Серверна частина отримує запити від клієнта, виконує обчислення та відправляє сформовану сторінку по мережі. Для безпечного та правильного формування і роботи з даними, була створена реляційна база даних.

В якості системи управління базою даних (СУБД) було обрано PostgreSQL, об'єктно-реляційна база даних. Це означає, що, хоча в першу чергу це реляційна модель, вона також включає в себе такі функції, як успадкування таблиць і перевантаження функцій, які частіше пов'язані з об'єктно-орієнтованими БД. Postgres здатний ефективно обробляти декілька завдань одночасно, ця властивість відома як паралелізм. Це досягається без блокувань читання завдяки реалізації MVCC, що забезпечує атомарність, узгодженість, ізоляцію і довговічність своїх транзакцій, також відомих як відповідність ACID [1, 4]

В якості середовища розробки інтегроване, інструментальне середовище, призначене для допомоги розробникам у створенні програмних додатків за наявності всього необхідного інструментарію. Багато IDE включають додаткові можливості для тестування, організації та рефакторінга коду. Додаткові функції, такі як автозаповнення, поряд з можливостями збірки, значно розширюють можливості розробника і підвищують швидкість їх розробки. Вибір оптимальної IDE залежить від таких факторів, як мовна підтримка, потреби операційної системи і витрат, пов'язаних з використанням IDE і т.д.

IntelliJ IDEA – інтегроване середовище розробки програмного забезпечення з підтримкою багатьох мов програмування. Система поставляється у вигляді урізаної по функціональності безкоштовної версії «Community Edition» і повнофункціональної комерційної версії «Ultimate Edition», для якої активні розробники відкритих проєктів мають можливість отримати безкоштовну ліцензію. Програмне забезпечення JetBrains IntelliJ IDEA – це провідне середовище швидкої розробки на мові програмування Java. IntelliJ IDEA становить високотехнологічний комплекс інтегрованих інструментів програмування, включає інтелектуальний редактор записів з розвинутими засобами автоматизації, потужні інструменти рефакторингу коду, вбудована підтримка стандартів розробки на рівні підприємств, механізм інтеграції середовища тестування та системами керування версій, унікальний інструмент оптимізації та перевірки коду, а також інноваційний візуальний конструктор графічних інтерфейсів.

Для зв'язку бази даних (БД) з додатком використовувалась технологія RESTful API. RESTful API - це прикладний програмний інтерфейс, який використовує HTTP- запити для отримання, вилучення, розміщення і видалення даних. API-інтерфейс RESTful, також званий веб-службою RESTful, заснований на технології передачі стану уявлень (REST), а його архітектурний стиль і підхід до комунікацій, часто використовується при розробці веб-сервісів. Одним з ключових переваг API REST є те, що він забезпечує гнучкість. Дані не прив'язані до ресурсів або методів, тому REST може обробляти кілька типів викликів, повертати різні формати даних. Така гнучкість дозволяє розробникам створювати API, який відповідає їх потребам, а також потребам різних клієнтів [2]. Цей протокол також виділяється завдяки своїй масштабованості. Завдяки розподілу між клієнтом і сервером продукт може бути масштабований командою розробників без особливих труднощів [6, 7].

Як зазначалось вище архітектура серверу була розроблена як RESTful API, клієнтська частина має доступ до даних з БД через посилання конкретних запитів до серверу. URL цих запитів має вигляд «./api/назва_таблиці/http_метод/...». Дані, які відправляються до клієнтської частини мають формат JSON та кодування UTF8. Також ці дані можуть розрізнятися за змістом, якщо наприкінці запиту є постфікс «-tree» - це означає, що в даних відобразатимуться певні обов'язкові зв'язки [5].

3. Програмна підтримка дослідження. Опис основних процедур та функцій

Збережена SQL функція пошуку на рівні БД



Функція `publication_records(title, date, authors, keywords, all_record)` містить 2 курсори `_authors` та `_keywords`, які конвертують прийняті данні у масив. Це дозволяє нам обирати декілька рядків із результату великого запиту. Через створений курсор можна отримувати рядки, застосовуючи команду `FETCH`.

```
_authors = concat('{', authors, '}') :: character varying[];  
_keywords = concat('{', keywords, '}') :: character varying[];
```

Параметр `all_record` приймаючи значення `true` видає усі існуючі записи у базі даних та відправляє їх на клієнт на головну “Книгу”

```
if all_record then  
  <<all_records>>  
  begin  
    for rec in select *  
      from publications  
      where id_publication <> 0  
      order by title asc  
    loop  
      exit all_records when rec isnull;  
      return next rec;  
    end loop;  
  end;
```

Запит у циклі вже дозволяє виконувати пошук по усім параметрам: дата, ключові слова, автори та назва.

```
for rec in select distinct p.id_publication,  
  p.type,  
  p.abstract,  
  p.date,  
  p.id_rating,  
  p.doi,  
  p.title,  
  p.id_journal  
from publications p,  
  authors a,  
  keywords k,  
  publications_authors pa,  
  publications_keywords pk  
where p.id_publication = pa.id_publication  
and a.id_author = pa.id_author  
and p.id_publication = pk.id_publication  
and k.id_keyword = pk.id_keyword  
and case  
  when _title notnull then lower(p.title) ilike concat('%', lower(_title), '%')  
  else p.id_publication = p.id_publication end  
and case  
  when _date notnull then p.date :: character varying ilike concat('%', _date, '%')  
  else p.id_publication = p.id_publication end  
and case  
  when _keywords notnull and _keywords[1] notnull then k.word = any (_keywords)  
  else p.id_publication = p.id_publication end  
and case  
  when _authors notnull and _authors[1] notnull then  
    concat(a.first_name, ' ', a.middle_name, ' ', a.surname) = any (_authors)  
  else p.id_publication = p.id_publication end  
and p.id_publication <> 0  
order by p.title  
loop  
begin  
  exit some_records when rec isnull;  
  return next rec;  
end;  
end loop;
```



При цьому, враховано, що користувач може ввести неповну назву публікації, таким чином, пошук назви працює по принципу “містить слово”. Також враховано можливість ігнорування реєстру тексту, тому не буде важливим у якому форматі отримуються дані для пошуку.

```
when _title notnull then lower(p.title) ilike concat('%', lower(_title), '%')
else p.id_publication = p.id_publication end
```

Пошук по ключовим словам та авторам відбуваються по масиву даних за допомогою оператора ANY. Так як ці запити знаходяться у циклі, проходить пошук по масиву на кожному новому ключовому слові чи авторі.

```
and case
```

```
when _keywords notnull and _keywords[1] notnull then k.word = any (_keywords)
else p.id_publication = p.id_publication end
```

```
and case
```

```
when _authors notnull and _authors[1] notnull then
concat(a.first_name, ' ', a.middle_name, ' ', a.surname) = any (_authors)
else p.id_publication = p.id_publication end
```

SQL функція створення нових записів в БД

Для кожної таблиці в БД було створено по 4 функції для основних операцій:

- вибірка (SELECT FROM);
- видалення (DELETE FROM);
- додавання (INSERT INTO);
- оновлення (UPDATE SET).

Це було зроблено для зручності використання SQL запитів у програмному коді самого додатка.

Кожна така функція має наступний функціонал:

- вхідні параметри мають значення за замовчуванням, тому при виклику в неї можна передавати не всі параметри;
- кожна така функція повертає результат у вигляді запису з таблиці, для якої була створена;
- якщо операція невдала (такого запису не існує, друкарська помилка в переданому параметрі, неможливо додати/оновити через повторення даних тощо) повертається нульовий запис із таблиці, де в кожній комірці міститься «Нульове значення»;

• для запитів на вибірку або деяких видалень присутня можливість вказати параметр «всі записи» та, якщо є, вказати додатковий параметр конкретизації.

Нижче наведено приклад такої функції на додавання нового запису в існуючий розділ публікацій. У переліку вхідних параметрів для кожного задаються значення за замовчуванням. Якщо було передано null або нічого, береться значення default для цього стовпця:

```
create or replace function publication_insert(_type publication_type DEFAULT NULL::publication_type,
__abstract text DEFAULT NULL::text,
__date date DEFAULT NULL::date,
__doi character varying DEFAULT NULL::character varying,
__title character varying DEFAULT NULL::character varying,
__stars double precision DEFAULT NULL::double precision,
__seen integer DEFAULT NULL::integer,
__title_journal character varying DEFAULT NULL::character varying)
returns SETOF publications
language plpgsql
```

```
as
```

```
$$
```

```
declare
```

```
rec publications%rowtype;
```

```
__id_rating int;
```

```
__id_journal int;
```

```
begin
```

```
begin
```

```
__id_rating = (
select (rating_insert(
__stars,
__seen
)).id_rating );
```

```
__id_rating = (
select (journal_record(
__title_journal
```



```
)).id_journal);
```

```
insert into publications (type, abstract, date, id_rating, doi, title, id_journal)
```

```
values (_type, _abstract, _date, __id_rating, _doi, _title, __id_journal) returning * into rec;
```

Якщо при створенні виникає помилка, оператор exception перехопить її та функція поверне інформацію про помилку

```
when sqlstate '23505' then
```

```
begin
```

```
raise notice 'Запис вже існує';
```

```
select into rec (publication_record(_doi => _doi)).*;
```

```
select rating_delete(__id_rating);
```

```
end;
```

```
when sqlstate '23502' then
```

```
begin
```

```
raise notice 'Нульове значення';
```

```
select into rec (publication_record()).*;
```

```
-- select rating_delete(__id_rating);
```

```
end;
```

3.Рекомендації щодо впровадження та використання

Програмне забезпечення, яке розроблено в рамках дослідження, має класифікацію клієнт-серверного додатку з тривірневою архітектурою: клієнт, сервер додатків та сервер бази даних. Для користування даним веб-додатком необхідно мати браузер – програмне забезпечення для перегляду веб-сторінок. Рекомендується використовувати такі браузери як: Google Chrome, Safari, Mozilla FireFox, Microsoft Edge або Internet Explorer. В інших браузерах робота даного веб-додатку не перевірялась. Ресурс не передбачає мобільної версії, так як, від самого початку створювався виключно як десктопний проект.

На рисунку 1 можна побачити головну сторінку сайту, на якій можна знайти соціальні мережі організації та фіксоване меню з основними можливостями сайту.

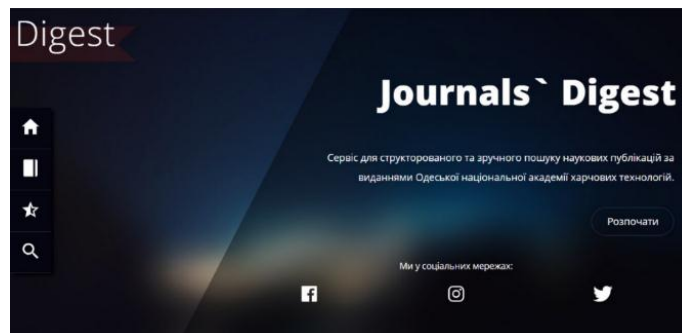


Рис. 1 – Головна сторінка сайту

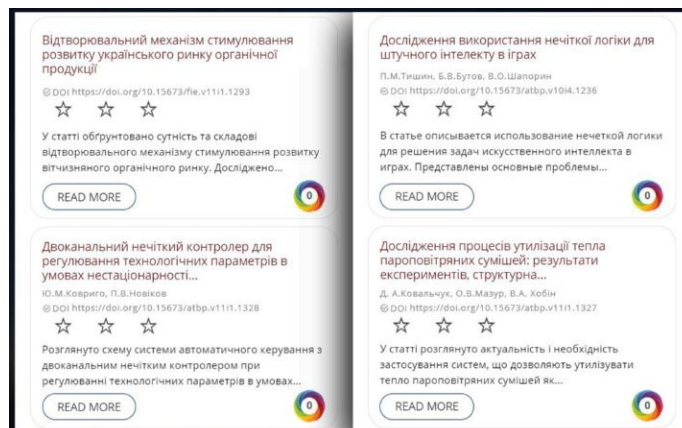


Рис. 2 – Книга сайту

Після натискання на один з цих елементів, користувача перенаправляє на сторінку з книгою публікації, як показано на рисунку 2.



Ця сторінка містить вже знайоме користувачу фіксоване бокове меню, книгу с публікаціями, та новий елемент - кнопка з розділами журналів видання ОНАХТ.

Також необхідно відзначити, що було розроблено веб-додаток який задовольняє майже всі потреби для людей які будуть його використовувати з урахуванням предметної області.

4. Практичне значення отриманих результатів

Результатом дослідження є електронний науковий дайджест, який включає в себе реалізацію всіх поставлених вимог, рішення побудованих задач та усунення виявлених проблем. Розроблений електронний науковий дайджест є повноцінним, адаптивним і функціональним ПЗ, який може бути запровадженом для використання у роботі ОНАХТ для співробітників та здобувачів вищої освіти.

Список використаних джерел:

- [1] Kwanya T., Stilwell C., Peter G. Library 3.0 Intelligent Libraries and Apomediation. Chandos Publishing, 2015. 190 p.
- [2] B. van Wyk, H. Geldenhuys. Learn 3.0 Meets Library 3.0: A Case Study / International Conference on e-Learning. 2018. 480 p.
- [3] A Beginner's Guide to Neural Networks and Deep Learning // Skymind: [Веб-сайт]. URL: <https://skymind.ai/wiki/neural-network> (дата звернення: 07.11.2019).
- [4] Object Relational Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.sqlalchemy.org/en/latest/orm/tutorial.html> - Назва з екрану. - Дата перегляду: 15.04.2018.
- [5] The Architecture of Open Source Applications (Volume 2) SQLAlchemy [Електронний ресурс] – Режим доступу до ресурсу: <http://aosabook.org/en/sqlalchemy.html> - Назва з екрану. - Дата перегляду: 15.04.2018.
- [6] Науменко Д. HTML, CSS, PHP, JavaScript, SQL – что и зачем? [Електронний ресурс] / Дмитрий Науменко – Режим доступу до ресурсу: <http://codeharmony.ru/materials/125>.
- [7] Язык HTML 5 – преимущества и недостатки [Електронний ресурс] – Режим доступу до ресурсу: <https://seodirection.ru/html5/>.

References:

- [1] T. Kwanya, C. Stilwell, and P. G. Underwood, *Library 3.0: intelligent libraries and apomediation*. Amsterdam: Chandos Publishing is an imprint of Elsevier, 2015.
- [2] B. van Wyk, H. Geldenhuys, “Learn 3.0 Meets Library 3.0: A Case Study”. International Conference on e-Learning, 2018.
- [3] “A Beginner's Guide to Neural Networks and Deep Learning,” *Pathmind*. [Online]. Available: <https://skymind.ai/wiki/neural-network>. [Accessed: 07-Nov-2019].
- [4] “SQLAlchemy 1.3 Documentation,” *Object Relational Tutorial - SQLAlchemy 1.3 Documentation*. [Online]. Available: <https://docs.sqlalchemy.org/en/latest/orm/tutorial.html>. [Accessed: 15-Apr-2018].
- [5] “SQLAlchemy,” *The Architecture of Open Source Applications (Volume 2): SQLAlchemy*. [Online]. Available: <http://aosabook.org/en/sqlalchemy.html>. [Accessed: 15-Apr-2018].
- [6] “HTML, CSS, PHP, JavaScript, SQL – что и зачем?” [Online]. Available: <http://codeharmony.ru/materials/125>. [Accessed: 10-Mar-2019].
- [7] “SEO Direction,” *Jazik HTML 5 – preimushestva I nedostatki*. [Online]. Available: <https://seodirection.ru/html5/>. [Accessed: 05-Mar-2019].

УДК 004.383.2:004.738.5:004.771

ПРИНЦИПИ ПОБУДОВИ ХМАР ТЕГІВ ДАНИХ

Хараш К. М.¹, Ольшевська О. В.², Титуренко Ж. А.³

^{1,2,3}Одеська національна академія харчових технологій, Одеса, Україна

ORCID: ² <http://orcid.org/0000-0002-4512-3915>, ³ <http://orcid.org/0000-0001-6774-1688>

E-mail: ² olshevskia.olga@gmail.com, ³ janettrnk@gmail.com

Copyright © 2018 by author and the journal “Automation of technological and business - processes.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: <https://doi.org/10.15673/atbp.v12i1.1699>