



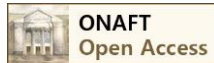
УДК 004.738.5.057.2:027-021.431

ВЕБ ТЕХНОЛОГІЇ В SMART БІБЛІОТЕЦІ

Зінченко І. І.¹, Шершун О. О.², Іванова А. Г.³^{1,2,3}Одеська національна академія харчових технологій, Одеса, УкраїнаORCID: ²<http://orcid.org/0000-0002-0914-7362>E-mail: ¹ lib.onaft@gmail.com, ² aleksander.shershun@gmail.com, ³ ivanovaanast29@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.v12i1.1696>

Анотація. Розглянута проблема неправильного та застарілого трактування діяльності бібліотеки, а також її позиціонування у сучасному світі. Складена інформаційна модель майбутнього програмного забезпечення, тобто ціль та засоби, якими система може оперувати для досягання цієї цілі та, що саме вимагають ці засоби, щоб їх можна було використовувати. За результатами дослідження, для систематизації сучасного уявлення про бібліотеку, було вирішено впроваджувати стандарт Library 3.0, який, в свою чергу, відповідає за модернізацію форми надання послуг бібліотекою за використанням таких технологій, як семантична мережа, хмарові сервіси, мобільні пристрої, у діяльність науково-технічної бібліотеки. Library 3.0 – це стандарт, який відповідає за електронну систематизацію сервісів, що забезпечують життєву діяльність академічної бібліотеки на основі комунікації між підрозділами та користувачем. На даний момент, це є найбільш захоплюючим досягненням у області наукових досліджень та розвитку бібліотек закладів вищої освіти, оскільки представлений стандарт підкреслює контекст, а не тільки засіб надання інформаційних послуг. Було проведено ретельний аналіз на підставі діючих методологій та підходів для створення концептуальної моделі проекту, основних принципів та засад розробки програмного продукту, допоміжних параметрів та засобів для досягнення бажаного результату, аналітичних даних. Був розроблений веб-ресурс для представлення Науково-технічної бібліотеки ОНАХТ за принципами Library 3.0 з урахуванням новітніх можливостей Web 2.0.

Abstract. The problem of incorrect and outdated interpretation of the library activity, as well as its positioning in the modern world, is considered. Compiled information model of the future of software, that is the goal and the means by which the system can operate for achieving this goal and that is demanding these features, so they can be used. According to the results of the research, in order to systematize the modern view of the library, it was decided to implement the Library 3.0 standard, which, in turn, is responsible for modernizing the form of library services using such technologies as semantic network, cloud services, mobile devices, in the scientific- technical library. Library 3.0 is the standard that is responsible for the electronic systematization of services that ensure the life of the academic library through communication between departments and the user. At the moment, this is the most exciting achievement in the research and development of higher education libraries, as the standard presented emphasizes context, not just the means of providing information services. A thorough analysis was carried out on the basis of existing methodologies and approaches to create the conceptual model of the project, basic principles and principles of software development, auxiliary parameters and tools for achieving the desired result, analytical data. A web resource was developed to present the Library of Science and Technology ONAHT on Library 3.0 principles, taking into account the latest Web 2.0 capabilities.

Ключові слова: Library 3.0, Web 2.0, веб-ресурс, HTML5, CSS, API, програмне забезпечення, сучасні технології, науково-технічна бібліотека, розробка програмного продукту, діджиталізація, JSON

Keywords: Library 3.0, Web 2.0, web resource, HTML5, CSS, API, software, modern technologies, scientific and technical library, software development, digitization, JSON

1. Теоретична складова

Невід'ємною складовою розвитку та становлення наукової діяльності закладів вищої освіти є науково-технічна бібліотека. У даному підрозділі знаходиться освітньо-кваліфікаційний обсяг робіт наукових діячів навчального закладу, більш сучасні тематичні роботи випускників та здобувачів вищої освіти, а також загальна література. Для



зручності та покращення якості доступу до будь-яких наукових робіт, у сучасному світі розроблено великий обсяг сервісів, для збереження та відтворення необхідної інформації, якими користується більшість науково-технічних бібліотек у вищих навчальних закладах по всьому світу. Для систематизації сучасного уявлення про бібліотеку, було розроблено стандарт Library 3.0, який, в свою чергу, відповідає за модернізацію форми надання послуг бібліотекою за використанням таких технологій, як семантична мережа, хмарові сервіси, мобільні пристрої. Отже, завдяки імплементації стандарту Library 3.0 у діяльність науково-технічної бібліотеки, кожен користувач отримує зручний та зрозумілий зовнішній вигляд сервісу, швидкий доступ, із будь-якого місця, до наукового фонду бібліотеки, зворотній зв'язок з адміністрацією та співробітниками підрозділів та постійне оновлення для задоволення потреб користувачів.

Актуальність даної теми полягає у постійному розвитку сучасних технологій спрямованих на покращення зручності використання сервісів та доступу до необхідної інформації. На сьогодні, великою проблемою є неправильне та застаріле трактування діяльності бібліотеки, а також її позиціонування у сучасному світі. Науково-технічна бібліотека ОНАХТ намагається слідкувати за новітніми стандартами у світовій науковій спільноті, враховувати бажання та потреби як науково-педагогічно складу академії, так і здобувачів вищої освіти, та впроваджувати сучасні технології до діяльності свого підрозділу у академії.

Головною метою дослідження була розробка веб-ресурсу для представлення Науково-технічної бібліотеки ОНАХТ за принципами Library 3.0 з урахуванням новітніх можливостей Web 2.0.

Для досягнення поставленої мети необхідно: проаналізувати основні засади та принципи стандарту Library 3.0; проаналізувати можливості та переваги стандарту Web 2.0; встановити зв'язок між приведеними стандартами; визначити основні потреби та забаганки користувачів ресурсу; провести аналіз сучасних технологій UI/UX; побудувати залежність між підрозділами НТБ ОНАХТ; виявити основні переваги та недоліки комплексу стандартів; описати взаємодію НТБ ОНАХТ та користувача допомогою структуризації сервісів в одному місці; обрати основні методи розробки програмного забезпечення; розробити сервіс для взаємодії з Науково-технічною бібліотекою ОНАХТ на основі проведеного аналізу.

2. Результати проведеного експерименту

Для розробки серверної частини веб-ресурсу було обрано фреймворк Flask для створення веб-додатків на мові програмування Python. До особливостей даного фреймворку необхідно віднести простоту, добре налагоджену маршрутизацію, вбудований веб сервер та зручність у роботі з API. Для розробки бази даних було обрано MongoDB, яка є документноорієнтованою системою управління базами даних. Приведена СУБД не потребує опису схеми таблиць та використовує JSON подібні документи, а також схему бази даних.

Для розробки клієнтської частини реферативного менеджера було використано середовище розробки PHPStorm з інтелектуальним редактором, який підтримує передові технології розробки та включає в себе підтримку PHP та баз даних, також володіє повним набором інструментів для розробки Інтернет ресурсу. Інтерфейс реферативного менеджера було побудовано за допомогою HTML5/CSS3, для побудови інтерактивного та функціонального додатку було використано мову TypeScript, а також фреймворк Angular 7 для покращення розробки.

Під час дослідження предметної області, було проведено ретельний аналіз на підставі вхідних даних. Вхідними даними виступають: діючі методології та підходи для створення концептуальної моделі проекту, основні принципи та засади розробки програмного продукту, допоміжні параметри та засоби для досягнення бажаного результату, аналітичні дані. На отриманих, методом аналізу, результатах за допомогою методу об'єктно-орієнтованого проектування була розроблена інформаційна модель майбутнього ПЗ. Та було обрана наступна концепція ІС, тобто ціль та засоби, якими система може оперувати для досягання цієї цілі та, що саме вимагають ці засоби, щоб їх можна було використовувати.

Концепція ІС продукту полягає у представленні даних, за допомогою веб-ресурсу, у зручному вигляді, зважаючи на новітні стандарти та основні забаганки користувачів. Опираючись на основні принципи та засади стандарту Library 3.0, необхідно поширювати інформацію та будувати комунікацію на основі потреб цільової аудиторії.

Для отримання інформації про вплив відповідних стандартів на сприйняття користувачем вихідного результату було побудовано графічне зображення обсягу аудиторії, попиту, впливу та якості роботи програмного продукту. Було використано такий інструментарій, як сервіс Google Analytics.

Google Analytics - це сервіс від світової компанії Google, який, за допомогою безкоштовного та у вільному доступі API, вбудовується до веб-ресурсів та аналізує повний обсяг вхідних даних, враховуючи основні ключові параметри. Для ефективного аналізу за допомогою наведеного інструментарію було враховано такі показники трафіку, глибокий аналіз поставлених задач та поступових налаштувань у роботі сервісу. Дана методи дозволяє підвищити конверсію сайту та може бути корисною для спеціалістів в галузі пошукової оптимізації, а також для веб-аналітики.

Конверсія - це важливий параметр, який характеризує відношення відвідувачів сайту, які виконали які-небудь цільові дії. Для сайту дуже важливо залучення трафіку, але більш важливим є поведінка користувача, зважаючи на його дії та позицію на веб-ресурсі, як постійного користувача чи клієнта (для Інтернет магазинів). Найбільшим показником ефективності веб-ресурсу є рівень конверсії, який відображає рівень взаємодії користувача з додатком.



Алгоритм роботи Google Analytics наступний: кожного разу, коли користувач виконує які-небудь дії на сторінках сайту, обробляється код вистежування JavaScript; до браузеру записується унікальний cookies; усі дії користувача фіксуються у одному пакеті (hit), який відправляється на сервер системи; відбувається обробка вхідних даних та подальша передача у акаунті системи у вигляді звітів.

Для оцінки результативності роботи веб-ресурсу, приведений інструментарії спирається на такі показники: кількість унікальних відвідувачів, кількість переглянутих сторінок, середній час, який було проведено на сайті, джерело трафіку та ключові слова [1].

На рисунку 1 зображено аналіз попиту серед відвідувачів, за такими критеріями, як їх обсяг, унікальність, кількість сеансів, попиту сторінок веб-ресурсу, час проведення на додатку. На основі перерахованих показників побудовано графік та діаграму, які варіюються за шкалою часу: у годинах, за день, тиждень та місяць. Зважаючи на наведений витяг з Google Analytics, розроблений ресурс у рамках дослідження, користується попитом серед користувачів у обсязі 223 користувачі, у середньому 580 сеансів, по два з половиною сеанси на одного відвідувача, серед загального обсягу переглянуте 712 сторінок при повторній навігації по сайту та середній час сеансів приблизно дві хвилини.

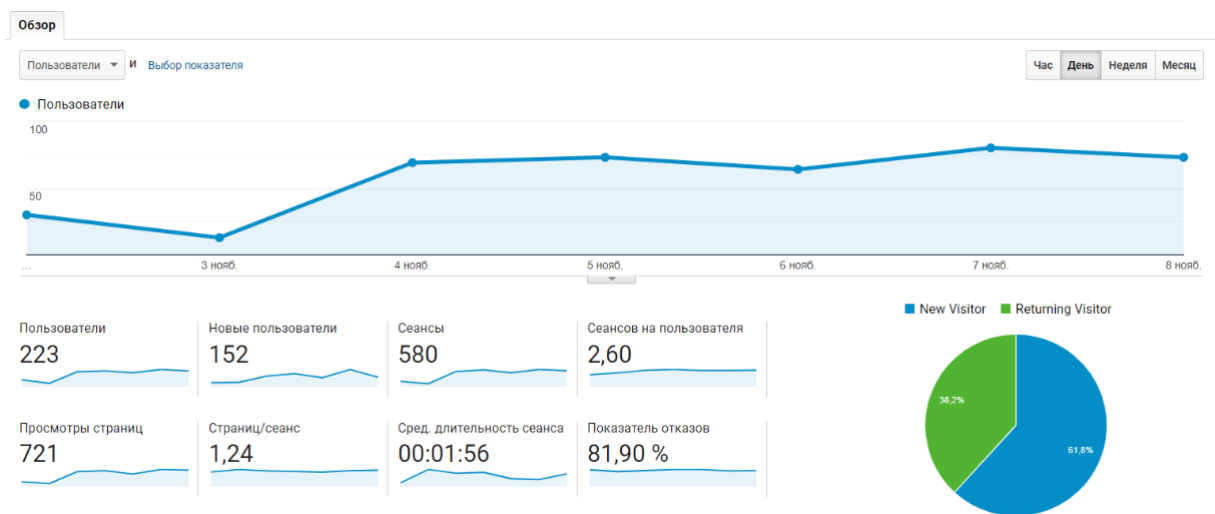


Рис. 1 – Аналіз кількості відвідувачів ресурсу

Для покращення попиту веб-ресурсу серед співробітників та здобувачів вищої освіти академії, необхідно враховувати вподобання серед сучасних браузерів, найбільш популярні пристрої для відображення сайту та коефіцієнт відвідування окремих сторінок веб-додатку.

Найбільш поширеним браузером серед користувачів є Google Chrome (475 відвідувачів), друге місце посідає Opera - загальна кількість 46 осіб, для користувачів продукцією Apple провідним браузером виступає Safari у кількості 16 користувачів. Для запуску ресурсу стартовою сторінкою виступає розділ за адресою /home, іноді до початкової сторінки використовується локалізація російською мовою, при неправильному введенні адреси домашньої сторінки, відбувається переадресація до сторінки з помилкою 404, яка говорить, що сторінки за даною адресою не знайдено.

Також, важливим фактором адаптації веб-ресурсу, для покращення комунікації додатку з користувачем, виступає версія браузера, оскільки технології для розробки програмних додатків у мережі Інтернет мають певні обмеження у тій, чи іншій версії. Зважаючи на проведений аналіз, більшість відвідувачів використовують останні версії браузера Google Chrome.

3. Обґрунтування та опис моделі та/або методу дослідження

При розробці системи дуже складно передбачити усі можливі варіанти поведінки на певні події, що в системі відбуваються, коли уявлення та данні о системі знаходяться в голові у розробника, а при масштабних проектах це не можливо, тому слід використати моделювання для побудови моделі системи, що вирішує дану проблему. Збудована модель системи надає можливість чітко бачити структуру системи та в разі поліпшує можливість прогнозування поведінки системи за рахунок можливості підстановки деяких змінних системи та розрахунку результату, який можемо отримати на виході системи.

Модель - об'єкт, який імітує чи копіює роботу об'єкту-оригіналу, що забезпечує можливість дослідити деякі властивості оригіналу; надає спрощене уявлення роботи системи для більш простого та комфортного її аналізу і передбачення її поведінки й отримання якісних, достовірних і кількісних результатів, які необхідні при прийнятті правильного управлінського рішення. При вирішенні та дослідженні конкретного спектру задач, коли необхідно виявити певні властивості піддослідного об'єкта чи системи, модель являється не тільки ефективним, але й часом



єдиним можливим інструментом для дослідження. Один об'єкт може бути описаний безліччю моделей, а різні об'єкти можуть бути описані однією й тією самою моделлю.

Моделювання – процес уявлення об'єкта реального світу абстрактною моделлю для одержання інформації про нього за допомогою проведення ряду експериментів з його моделлю. Під моделюванням зазвичай розуміють створення точного опису системи та її елементів; метод пізнання, що полягає в побудові та дослідженні повудіки моделей. Моделювання зменшую складність дослідження об'єкту, що надає можливість його відтворити та використати у подальших експериментах. Воно використовується при досліді існуючих систем, коли дослід чи експеримент над реальний об'єктом чи системою неможливі чи не рентабельні у світі певних причин.

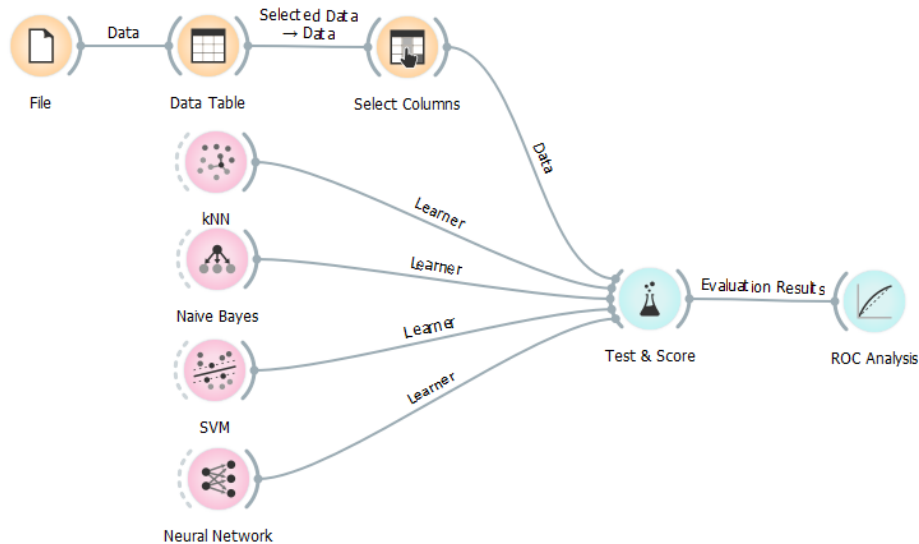


Рис. 2 – Схема проведення аналізу проекту

Для візуалізації процесу моделювання ПЗ, було побудовано аналітичну схему “stack holding” даних для обробки інформації отриманої після дослідження аналітичної складової веб-ресурсу НТБ ОНАХТ у середовищі для візуалізації даних Orange, зображено на рисунку 2.

Для побудови схематичного зображення проведеного експерименту було використано наступні компоненти у середовищі Orange:

1. Data Table – компонент для структурованого виведення даних, на вході отримуються певний файл та за обраними параметрами сортується його зміст для подальшої візуалізації та зручної роботи з необхідною інформацією.

2. Select Columns – компонент для відсортування необхідних рядків для роботи з ними через аналітичні компоненти, обирається відповідні стовпчики та тип роботи з ними.

3. Test and Score – даний компонент відповідає за візуалізацію даних у бажаному вигляді, перша функція віджету виведення даних у таблицю за класифікаторами, а друга – відтворення результату класифікації даних для подальшого роботи з ними за допомогою інших віджетів.

4. kNN – візуалізація даних за допомогою нейронних зв'язків, на вході отримуються дані та методи для структуризації інформації, результуючою виступає нейронний алгоритм та навчаюча модель.

5. Naive Bayes - цей віджет надає вивчений класифікатор на виході, використовує алгоритм навчання з налаштуваннями, визначеними користувачем. Її можна подавати у віджети для тестування учнів, наприклад тестових членів навчання. Класифікатор – наївний байєсівський класифікатор (підтип загального класифікатора), побудований із навчальних прикладів на вході. Якщо приклади не наводяться, на виході немає класифікатора.

6. SVM – використовує стандартний LibSVM учень, який підтримує кілька вбудованих типів ядра та визначені користувачем функції ядер, написані на Python. Тип ядра позначається константами Linear, Polynomial, RBF, Sigmoid та Custom, визначеними в Orange.classification.svm.kernels. Спеціальна функція ядра повинна приймати два екземпляри даних і повертати поплавок.

7. Neural Network – віджет використовує багатосаровий алгоритм Perceptron sklearn, який дозволяє вивчати лінійні та нелінійні моделі.

8. ROC Analysis – демонструє криві ROC для моделей та відповідний опуклий корпус. Враховуючи витрати на помилкові позитиви та хибні негативи, він також може визначити оптимальний класифікатор та поріг [2].



4. Проектування системи

Інформаційна модель – це модель, що описує інформаційні процеси або містить інформацію про властивості і стан об'єктів, процесів, явищ. На рисунку 3. зображена інформаційна модель взаємодії клієнта з сервером та взаємодія компонентів сервера між собою.

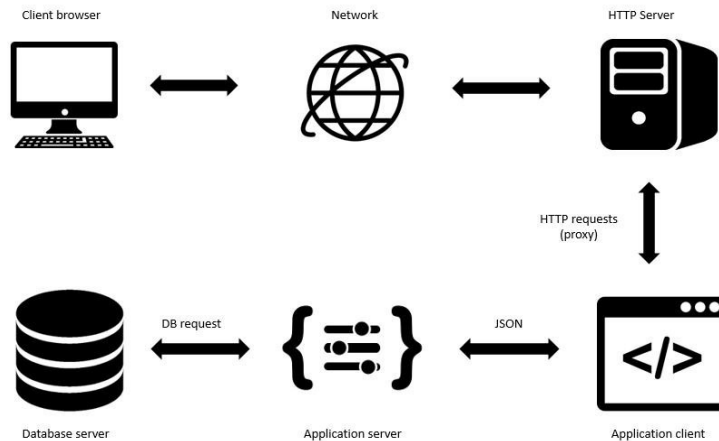


Рис. 3 – Інформаційна модель системи

Пояснення стосовно блоків схеми: Client Browser – клієнтський агент (інакше браузер) для взаємодії з сервером додатку за допомогою передачі гіпертексту; Network – глобальна мережа інтернет; HTTP Server – вільний веб-сервер; Application client – інтерфейс додатку, передає запити клієнта; Application Server – програмне середовище для функціонування серверу-додатків та БД, відповідає на запити клієнта; Database Server – сервер БД, який вміщує базу, побудовану на основі моделі, яка задовольняє потреби додатку.

Для реалізації серверної частини було обрано Flask Framework, який являє собою платформу Python, яка забезпечує можливість розробки веб-додатків мовою Python. Це є мікрофреймворк, тобто каркас додатку, який дозволяє реалізувати базові функції. За необхідності розширення додатку, можливо використовувати сторонні бібліотеки так, ніби їх підтримує сам фреймворк [3].

Поширюється Flask на умовах ліцензії BSD (Berkeley Software Distribution licence), є безкоштовним програмним забезпеченням з відкритим вихідним кодом та може змінюватися спільнотою без обмежень.

Мова Python зі своєю появою дуже швидко поширилася у спільноті розробників ПЗ. Це зумовлено її універсальністю, в тому числі Python придатна для написання веб-серверів. Однак, довгий час користувачі багатьох веб-додатків, створених за допомоги Python, були обмежені у виборі веб-серверів, які можливо було використовувати сумісно з додатками.

Зазвичай, розробники обмежувалися підтримкою одного способу підключення до веб-сервера. Одні додатки могли використовувати протокол CGI (Common Gateway Interface) або Fast CGI, інші прив'язувалися до модуля Apache mod_python, а деякі додатки взагалі були в змозі підтримувати лише API, специфічне для одного сервера. Така ситуація сповільнювала розповсюдження веб-додатків, написаних на Python. З цієї причини з'явився інтерфейс шлюза веб-сервера - протокол WSGI (Web Server Gateway Interface).

Головною метою розробки даного протоколу стало розмежування вибору каркасів для розробки веб-додатків від вибору веб-серверів. Це дозволяє розробникам зосереджуватися на своїх задачах та відрізняє WSGI від інших протоколів зв'язку з сервером.

З точки зору WSGI, веб-додаток поділяють на дві частини: сервер (шлюз) та безпосередньо додаток (каркас для створення додатків). Для звернення до додатку, серверна частина використовує виклик об'єкту. Також, WSGI дозволяє створювати додатки-посередники, які є додатком до веб-сервера та сервером для веб-додатку. Такі посередники можуть бути використані для попередньої обробки запитів до додатку або подальшої обробки його відповідей. Таким чином, сервер викликає додаток для кожного http-запиту, йому призначеного [4].

Клієнтська частина була реалізована завдяки набору інструментарію для створення якісного сучасного Інтернет ресурсу. Процес створення веб-додатку є досить трудомістким та потребує багато часу. Для створення проекту було проаналізовано існуючі засоби та методи реалізації програмного продукту, мови програмування, а також технології для поліпшення розробки. Опіраючись на проведений аналіз, для реалізації клієнтської частини були обрано такі технології, як зв'язка HTML5, CSS3, фреймворки Angular 5 та Bootstrap, а також бібліотека JQuery.

HTML (Hyper Text Mark Language) – це мова розмітки гіпертексту для створення веб-сторінок. Дана мова інтерпретується браузером та відображається у вигляді документу зручному та зрозумілому людському оку. HTML є



невід'ємною частиною та основою практично будь-якої веб-сторінки. Саме HTML наділяє зміст сторінки певним сенсом, а реалізується це завдяки спеціально орієнтованим тегам [5].

Згенерований HTML документ інтерпретується браузером, тобто будує його структуру та відображає її у відповідності інструкціям застосованих до файлу, а саме каскадні таблиці стилів та скрипти. Якщо текстова розмітка документу побудована правильно, то у вікні браузера буде відображена HTML сторінка, яка містить усі прописані елементи.

Кожен HTML документ дотримується певної структури для подальшої правильної інтерпретації браузером. Усі основні правила, яких дотримується мова HTML, які містяться у файлі оголошення типу документа, який представляє собою XML документ. Даний документ визначає які теги, атрибути та їх значення відповідають конкретній версії HTML документу.

На протязі багатьох років HTML розвивається та набуває певних змін, на сьогодні існує найбільш сучасна версія під назвою HTML5. В першу чергу, HTML5 є мовою гіпертекстової розмітки, але на відміну від HTML та XHTML, набув великий обсяг функцій та було усунуто основні недоліки [6].

Зазвичай створення веб-сторінок відбувається у зв'язці мови гіпертекстової розмітки HTML та каскадної таблиці стилів CSS. Використання відповідної комбінації покращує зовнішній вигляд веб-сторінки та надає їй необхідної інтерактивності.

CSS (Cascading Style Sheets) чи каскадні таблиці стилів, використовуються для виконання опису зовнішнього вигляду документу, написаного мовою розмітки. Зазвичай CSS використовується для створення та зміни стилю елементів веб-сторінки, написаних на мовах HTML та XHTML.

Головною метою створення CSS було відокремлення опису логічної структури веб-сторінки від оформлення зовнішнього вигляду. Отже, мова HTML описує логічну структуру веб-сторінки, а CSS описує зовнішній вигляд даної структури. Відокремлення опису логічної структури та представлення документа дозволяє більш гнучко керувати зовнішнім видом документа та позбавитись від дублювання частин коду. За допомогою CSS є можливість задавати веб-сторінці та окремим її елементам розміри, кольори, поля та відступи, та інші зміни у зовнішньому вигляді. [14]

CSS3 є останньою версією каскадної таблиці стилів та розширює і додає нові можливості до побудови зовнішнього вигляду веб-сторінки. Головними перевагами виступають комбінація фонового зображення та кольору, додавання тіней та прозорості до блоків, нові можливості до редагування тексту і тому подібне.

HTML та CSS є основою будь якого сайту, незважаючи на напрям розробки серверної чи клієнтською частини, знання даної зв'язки є обов'язковою. Наведена технологія надає можливість створювати сучасні, адаптивні та інтерактивні сайти. Для маніпуляції елементами веб сторінки та наданням їм візуальних ефектів використовується допоміжні технології такі як бібліотека JQuery та фреймворк Angular 5.

Angular - це фреймворк від компанії Google для створення Інтернет ресурсів, насамперед спрямованим на створення односторінкового Інтернет ресурсу. Зважаючи на це, є спадкоємцем іншого фреймворку AngularJS, але потрібно пам'ятати, що Angular це не нова версія AngularJS, а повністю новий фреймворк. Фреймворк володіє таким функціоналом, як двостороннє зв'язування, яке дозволяє динамічно змінювати дані в одному місті клієнтської частини при зміні даних моделі в іншій частині, шаблони, маршрутизація і тому подібне. Використовує мову програмування TypeScript [7].

5. Програмна підтримка дослідження

Опис основних процедур та функцій

Для правильної взаємодії клієнтської частини з інформацією яка поступає з серверу, було створено клас `HttpService` для роботи з API, які формуються на стороні backend. Даний клас містить такі ключові поняття:

- `url` – посилання за яким отримуються дані для подальшої роботи;
- `public responseCache = new Map()` – для хешування даних під час пересування між вкладками сервісу;
- `public getNews(): Observable<any>` – отримання даних для подальшої роботи з ними, у даному методі відбувається занесення даних до змінної, встановлення заголовків та хешування даних до сталої, яка зберігається у сховищі браузера на час проведення відповідної сесії.
- `public postMessage(body)` – метод для відправки даних а сервер з форми зворотного зв'язку.

```
export class HttpService {
  url = 'https://library.onaft.edu.ua/api';
  public responseCache = new Map();
  constructor(private httpClient: HttpClient, private cacheUrl: CacheService) {
    cacheUrl.addToCache(this.url);
  }
  public getNews(): Observable<any> {
    const headers = new HttpHeaders();
    headers.set('Cache-Control', 'public, max-age=31536000');
  }
}
```



```
const newsCache = this.responseCache.get(this.url + '/news');
if (newsCache) {
    return of(newsCache);
}

const response = this.httpClient.get<any>(this.url + '/news', {headers});
response.subscribe(data => this.responseCache.set(this.url + '/news', data));
return response;
}
{...}
public postMessage(body) {
    return this.httpClient.post('https://library.onaft.edu.ua/api', body);
}
}
```

Сервіс для хешування даних CacheService виконує функцію збереження вхідних даних для роботи у браузері, під час однієї сесії, без постійного звернення до сторони backend за необхідним набором даних.

```
export class CacheService {
    private services = [];
    constructor() {}
    public addToCache(serviceUri: string) {
        return this.services.indexOf(serviceUri) > -1;
    }
    public add(serviceUri: string) {
        if (!this.addToCache(serviceUri)) {
            this.services.push(serviceUri);
        }
    }
}
```

6. Рекомендації щодо впровадження та використання

Для початку роботи з програмним продуктом розробленим у рамках дослідження потрібно скористатися одним з вище представлених браузерів на будь-якій платформі, яка відповідає вказаним системним вимогам. Також, потрібно зауважити, що розроблений Інтернет ресурс є адаптивним під будь-який пристрій. Отже, у адресному рядку потрібно ввести адресу розташування Інтернет ресурсу: library.onaft.edu.ua. Після введення адреси, потрапляємо на початкову сторінку Інтернет ресурсу, яка зображена на рисунку 4.

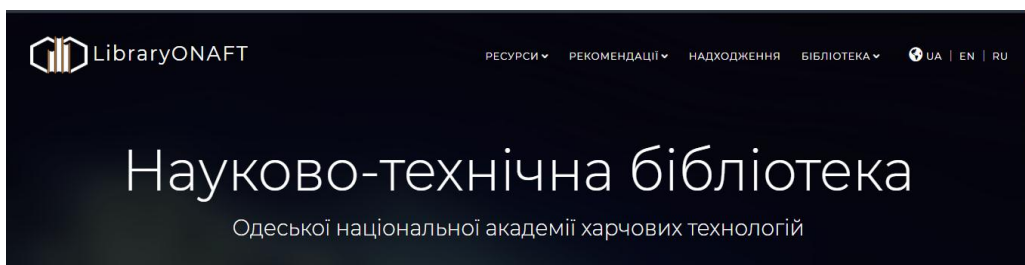


Рис. 4 – Головна сторінка сайту

На головній сторінці веб ресурсу знаходиться секція з основними новинами та заходами Науково-технічної бібліотеки ОНАХТ для інформування співробітників та здобувачів вищої освіти про важливі наукові семінари, нові надходження та можливості наукової бібліотеки. Нижче знаходиться блок з інформацією про бібліотеку. У лівій частині знаходиться описова інформація про діяльність підрозділу академії та технічні зали. Права частина містить багатокутники з назвами підрозділами бібліотеки та інформацією про основну діяльність відділу, при наведенні на один з шестикутників він обертається, та на зворотній стороні відображено інформацію про керівника підрозділу та місце знаходження. За блоком «Про бібліотеку» знаходить форма зворотного зв'язку. За бажанням користувач може звернутись до співробітника бібліотеки та викласти свої побажання чи зауваження. Останній блок головної сторінки ресурсу містить основні питання стосовно функціонування бібліотеки та отримання певного спрямування інформації. Для того, щоб отримати відповідь на одне з приведених питань, необхідно натиснуте на відповідний розділ та буде відображено блок з інформацією. Навігаційне меню поділяється на такі розділи, як «Ресурси», «Рекомендації»,



«Надходження» та «Бібліотека». Також, у правому куті знаходиться вибір мови перекладу ресурсу, представлено три основні мови: українська, російська та англійська.

Якщо навести на пункт меню «Ресурси» з'явиться випадające навігаційне меню, з такими пунктами:

- Електронний каталог – перенаправляє на один з ресурсів бібліотеки, які містять інформацію про основні видання академії, наявний пошук за категоріями, назвою, авторами та розширена форма. Для деяких видань наявна можливість переглянути повний зміст наукової роботи.

- Електронний архів (репозитарій) – науковий репозитарій бібліотеки, який містить перелік всіх наукових робіт на базі академічної діяльності.

- Центр наукової періодики (КЦВ) – веб-ресурс координаційного центру видання, який містить корисні посилання та інформативну складову для наукових діячів.

- Патенти – сторінка, яка містить сортування патентів за роками видання та інформацію про них: назву патенту, основні характеристики та автора

Наступний розділ меню «Рекомендації», який складається з наступних підрозділів:

- Енциклопедія вчених – список видатних діячів Одеської національної академії харчових технологій. На відповідній сторінці ресурсу представлено таку інформацію про діячів: повне ім'я, коротка біографія та, при натисканні на кнопку «Детальніше», відкривається сторінка з повною біографією та науковими роботами видатного діяча академії.

7. Практичне значення отриманих результатів

У рамках дослідження було розроблено програмний продукт, а саме веб-додаток для доступу до наукового фонду, основної інформації, актуальних заходів та ресурсів Науково-технічної бібліотеки Одеської національної академії харчових технологій. Даний ресурс реалізує основні переваги світового стандарту Library 3.0 для покращення сприйняття користувачами представленої інформації у вигляді веб-додатку, який розроблено за допомогою технології Web 2.0. Практичне значення отриманих результатів - це програмний продукт, який задовольняє всіх основним потребам користувачів, завдяки зручному та зрозумілому дизайну, правильній структуризації представленої інформації, організованій навігації по веб-додатку з використанням анімації та новітніх стандартів дизайну, підтримка програмного продукту та постійне оновлення даних для кінцевого відображення для здобувачів вищої освіти та співробітників академії.

Список використаних джерел:

- [1] Егорова И. Н., Кадушкевич О. Н. МЕТОДИКА ЭФФЕКТИВНОГО ИСПОЛЬЗОВАНИЯ ИНСТРУМЕНТОВ GOOGLE ANALYTICS // Scientific Journal «ScienceRise». 2016. Т. 1, вып. 18. С. 40-44. <https://doi.org/10.15587/2313-8416.2016.58765>.
- [2] Documentation // Orange: [Веб-сайт]. URL: <https://orange.biolab.si/docs/> (дата звернення: 12.11.2019).
- [3] 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.
- [4] 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.
- [5] Науменко Д. HTML, CSS, PHP, JavaScript, SQL – что и зачем? [Електронний ресурс] / Дмитрий Науменко – Режим доступу до ресурсу: <http://codeharmony.ru/materials/125>.
- [6] Язык HTML 5 – преимущества и недостатки [Електронний ресурс] – Режим доступу до ресурсу: <https://seodirection.ru/html5/>.
- [7] Введение в Angular [Електронний ресурс] – Режим доступу до ресурсу: <https://metanit.com/web/angular2/1.1.php>

References

- [1] Egorova I. N., Kadushkevich O. N., “METODIKA E`FFEKTIVNOGO ISPOL`ZOVANIYa INSTRUMENTOV GOOGLE ANALYTICS”, *Scientific Journal «ScienceRise»*, 2016, V.1, Iss. 18., p. 40-44. <https://doi.org/10.15587/2313-8416.2016.58765>.
- [2] Bioinformatics Laboratory and University of Ljubljana, *Orange Data Mining - Documentation*. [Online]. Available: <https://orange.biolab.si/docs/>. [Accessed: 05-Mar-2020].
- [3] “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].
- [4] “SQLAlchemy,” *The Architecture of Open Source Applications (Volume 2): SQLAlchemy*. [Online]. Available: <http://aosabook.org/en/sqlalchemy.html>. [Accessed: 15-Apr-2018].
- [5] “HTML, CSS, PHP, JavaScript, SQL – что и зачем?,” [Online]. Available: <http://codeharmony.ru/materials/125>. [Accessed: 10-Mar-2019].
- [6] “SEO Direction,” *Jazik HTML 5 – preimushestva I nedostatki*. [Online]. Available: <https://seodirection.ru/html5/>. [Accessed: 05-Mar-2019].
- [7] “Vvedeniye v Angular,” [Online]. Available: <https://metanit.com/web/angular2/1.1.php>. [Accessed: 05-Mar-2020].