

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри **Є. О. Давиденко**
«___» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Вебзастосунок інтернет-видання

Спеціальність 121 «Інженерія програмного забезпечення»

121 – КРБ – 408.22020803

Виконав студент 4-го курсу, групи 408 _____ **О. С. Шляхтун**
«___» червня 2023 р.

Керівник канд. техн. наук, викладач _____ **К. Ю. Бережний**
«___» червня 2023 р.

Консультант канд. техн. наук, доцент _____ **А. О. Алексєєва**
«___» червня 2023 р.

Миколаїв – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Чорноморський національний університет
імені Петра Могили**

Факультет комп'ютерних наук

Кафедра Інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри інженерії програмного
забезпечення, канд.техн.наук, доцент,

_____Є.О. Давиденко

«____»_____2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи бакалавра

Видано студенту групи 408 факультету комп'ютерних наук

Шляхтуну Олександрю Сергійовичу

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

Вебзастосунок інтернет-видання

Затверджена наказом по ЧНУ від «17» березня 2023 р. № 60

2. Строк представлення кваліфікаційної роботи «____» _____2023 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом є програмне забезпечення вебзастосунку для
інтернет-видання

4. Перелік питань, що підлягають розробці:

аналіз предметної області, порівняльний аналіз аналогів, визначення вимог та
функціоналу системи, моделювання, проєктування програмного застосунку,
проєктування інтерфейсу користувача, проєктування та розробка БД, кодування
та тестування застосунку

5. Перелік графічних матеріалів:

Презентація

6. Завдання до спеціальної частини

Дослідження питань охорони праці, які безпосередньо пов'язані з діяльністю розробника програмного забезпечення

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Алексєєва А.О., канд. техн. наук, доцент	Кафедра екології Медичного інституту ЧНУ ім. Петра Могили	Спеціальна частина з охорони праці

Керівник роботи канд. техн. наук, викладач Бережний К. Ю.
(посада, прізвище, ім'я, по батькові)

Завдання прийнято до виконання

(підпис)

Шляхтун Олександр Сергійович
(прізвище, ім'я, по батькові студента)

Дата видачі завдання «_» _____ 2023 р

(підпис)

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: Вебзастосунок інтернет-видання

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КРБ	18.01.2023	31.01.2023	виконано
2	Огляд літератури за темою роботи	01.02.2023	15.02.2023	виконано
3	Складання календарного плану КРБ	16.02.2023	21.02.2023	виконано
4	Аналіз предметної області	22.02.2023	28.02.2023	виконано
5	Розробка проєктних рішень	22.02.2023	28.02.2023	виконано
6	Моделювання та конструювання ПЗ	01.03.2023	15.03.2023	виконано
7	Кодування, тестування та апробація розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	16.03.2023	07.03.2023	виконано
8	Розробка спеціальної частини з охорони праці	08.05.2023	14.05.2023	виконано
9	Відгук керівника КРБ	15.05.2023	21.05.2023	виконано
10	Оформлення КРБ та презентації	22.05.2023	25.05.2023	виконано
11	Попередній захист	26.05.2023	31.05.2023	виконано
12	Рецензування	01.06.2023	07.06.2023	виконано
13	Завершення оформлення КРБ та презентації	08.06.2023	15.06.2023	виконано
14	Захист кваліфікаційної роботи	16.06.2023	23.06.2023	виконано

Розробив студент: Шляхтун Олександр Сергійович

(прізвище, ім'я та по-батькові)

_____ (підпис)

«__» _____ 20__р.

Керівник роботи: Бережний Кирило Юрійович

(прізвище, ім'я та по-батькові)

_____ (підпис)

«__» _____ 20__р.

АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Вебзастосунок інтернет–видання»

Студент 408 гр.: Шляхтун Олександр Сергійович

Керівник: канд. техн. наук, викладач Бережний К. Ю.

Актуальність роботи полягає у необхідності розробки вебзастосунку для інтернет–видання, що має високу швидкість завантаження, зручний та інтуїтивно зрозумілий інтерфейс для користувачів, можливість швидкої розсилки новинних листів та зручне керування контентом сайту. Оскільки в сучасному світі інформаційні технології швидко розвиваються, та все більше людей користується інтернетом для отримання новин та інформації, створення високоефективного вебзастосунку для інтернет–видання є важливим завданням. Розробка такого застосунку дозволить забезпечити швидке та зручне відображення новин на сайті, залучення більшої аудиторії та зростання конкурентоспроможності інтернет–видання на ринку медіа–послуг.

Об’єктом роботи є процес розробки вебзастосунку інтернет–видання.

Предметом роботи є підходи та інформаційні технології розробки вебзастосунків.

Метою кваліфікаційної роботи є підвищення якості надання актуальної різноманітної інформації у сфері медіа за рахунок розробки вебзастосунку інтернет–видання.

Для досягнення поставленої мети було:

1. проведено аналіз існуючих інтернет–видань та їх вебзастосунків;
2. розроблено архітектуру вебзастосунку для інтернет–видання;
3. реалізовано створену архітектуру вебзастосунку з використанням сучасних технологій програмування;
4. проведено тестування та аналіз результатів роботи вебзастосунку;
5. розроблено рекомендації щодо подальшого розвитку та удосконалення вебзастосунку.

У першому розділі проведено аналіз предметної сфери вебзастосунків інтернет–видань. У другому змодельовані об’єкт та предмет дослідження. У третьому обґрунтовано вибір засобів розробки. У четвертому виконано кодування та тестування застосунку.

За результатами роботи створенно функціональний вебзастосунок для інтернет–видання з можливістю редагування та публікації новин, коментарів, роботи з користувачами та статистики відвідувань.

КРБ викладена на 81 сторінку, вона містить 4 розділи, 30 ілюстрацій, 5 таблиць, 25 джерел в переліку посилань.

Ключові слова: вебзастосунок, інтернет–видання, управління вмістом, плагіни, шаблони.

ABSTRACT

of the Bachelor's Thesis

“Web application for an online publication”

Student 408: Shlyakhtun Oleksandr Sergiyovych

Supervisor: Candidate of Technical Sciences, Lecturer K. Yu. Berezhnyi

The relevance of the work lies in the need to develop a web application for an online publication that has a high loading speed, a user-friendly and intuitive interface, the ability to quickly send newsletters, and convenient site content management. As information technologies are rapidly developing in the modern world, and more and more people use the internet to get news and information, creating a high-performance web application for an online publication is an important task. Developing such an application will provide fast and convenient news display on the website, attract a larger audience, and increase the competitiveness of the online publication in the media services market.

The object of the work is the process of developing a web application for an online publication. The subject of the work is the approaches and information technologies for developing web applications.

The purpose of the thesis is to improve the quality of providing up-to-date diverse information in the media sphere through the development of a web application for an online publication.

To achieve the set goal, the following was done:

1. conducted an analysis of existing online publications and their web applications.
2. developed the architecture of a web application for an online publication.
3. implemented the created architecture of the web application using modern programming technologies.
4. conducted testing and analysis of the results of the web application.
5. developed recommendations for further development and improvement of the web application.

The first chapter of the work analyzes the subject area of web applications for online publications. In the second chapter, the object and subject of the research are modeled.

The third chapter justifies the choice of development tools. In the fourth chapter, the application is developed, coded, and tested.

As a result of the work, a functional web application for an online publication with the ability to edit and publish news, comments, work with users, and statistics of visits has been created.

The work is presented in 81 pages and consists of 4 chapters, 30 illustrations, 5 tables, and 25 sources in the reference list.

Keywords: web application, online publication, content management, WordPress, plugins, templates.

ЗМІСТ

ВСТУП	3
1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ ІНТЕРНЕТ–ЖУРНАЛІСТИКИ	5
1.1 Аналіз предметної області	5
1.2 Аналіз існуючих аналогів інтернет–видань	6
1.3 Специфікація вимог до програмного вебзастосунок.....	16
Висновки до розділу 1	22
2 ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ТА ТЕХНОЛОГІЇ РОЗРОБКИ	
ВЕБЗАСТОСУНКУ	24
2.1 Обґрунтування вибору мови програмування	24
2.2 Обґрунтування вибору СКБД	36
Висновки до розділу 2	40
3 ПРОЄКТУВАННЯ СИСТЕМИ З ВИКОРИСТАННЯМ МОВИ UML ...	41
3.1 Діаграма прецедентів	42
3.2 Опис варіантів використання.....	43
3.3 Діаграма діяльності	48
3.4 Діаграма станів та переходів	50
3.5 Структура бази даних	52
3.6 Проєктування інтерфейсу користувача	54
Висновки до розділу 3	58
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ	60
4.1 Діаграма класів	60
4.2 Діаграма компонентів	61
4.3 Діаграма взаємодії.....	62
4.4 Інструкція користувача	66
Висновки до розділу 4	70
ВИСНОВКИ	71
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	72

ВСТУП

З моменту зародження Інтернету, медіа–індустрія була однією з найбільш динамічне розвиваючихся галузей, інтернет–видання – не виняток.

За останні десятиліття зростає кількість людей, які отримують новини через Інтернет, а не через традиційні друковані ЗМІ або телебачення. Інтернет–видання забезпечують широкий спектр інформаційних послуг для своїх читачів, таких як швидкий доступ до свіжих новин, глибоких аналізів, відгуків, коментарів та інших матеріалів.

Однак, багато інтернет–видань мають проблеми з ефективним та зручним використанням вебресурсів. Багато з них не мають достатньої функціональності та інтерфейсу для задоволення потреб читачів.

Крім того, зростання популярності Інтернету в останні роки сприяє розвитку інтернет–видань. Зараз практично кожна видавнича компанія має своє віртуальне відділення в мережі Інтернет. У такому контексті особливу актуальність набуває створення якісного та зручного вебзастосунку інтернет–видання.

Актуальність роботи полягає у необхідності розробки вебзастосунку для інтернет–видання, що має високу швидкість завантаження, зручний та інтуїтивно зрозумілий інтерфейс для користувачів, можливість швидкої розсилки новинних листів та зручне керування контентом сайту. Оскільки в сучасному світі інформаційні технології швидко розвиваються, та все більше людей користується інтернетом для отримання новин та інформації, створення високоефективного вебзастосунку для інтернет–видання є важливим завданням. Розробка такого застосунку дозволить забезпечити швидке та зручне відображення новин на сайті, залучення більшої аудиторії та зростання конкурентоспроможності інтернет–видання на ринку медіа–послуг.

Науково–практичне значення цієї роботи полягає в розробці нового продукту, який може бути використаний видавництвами та журналами для створення власних інтернет–видань. Крім того, використання сучасних технологій

при розробці даного вебзастосунку може бути корисним для інших сфер, де вимагається швидкий та зручний доступ до інформації, таких як бізнес, освіта, наука тощо.

Об’єктом кваліфікаційної роботи є процес розробки вебзастосунку інтернет–видання .

Предметом кваліфікаційної роботи є підходи та інформаційні технології розробки вебзастосунків.

Метою кваліфікаційної роботи є підвищення якості надання актуальної різноманітної інформації у сфері медіа за рахунок розробки вебзастосунку інтернет–видання.

Для досягнення мети було поставлено наступні **завдання**:

1. проаналізувати існуючі вебзастосунки для інтернет–видань та визначити їх переваги та недоліки;
2. розробити вимоги до функціоналу та дизайну вебзастосунку;
3. розробити архітектуру та вибрати технології для реалізації вебзастосунку;
4. розробити та реалізувати вебзастосунок з необхідним функціоналом;
5. провести тестування вебзастосунку та зібрати фідбек від користувачів.

1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ ІНТЕРНЕТ–ЖУРНАЛІСТИКИ

1.1 Аналіз предметної області

Сучасний світ характеризується стрімким розвитком інформаційних технологій та безпрецедентною доступністю інформації [1]. Інтернет–технології, зокрема інтернет–видання та інтернет–журналістика, стали невід’ємною частиною нашого повсякденного життя. Вони забезпечують нам можливість швидкого доступу до актуальної інформації, новин та аналітичних матеріалів з усього світу. У цьому вступі ми розглянемо поняття інтернет–видання та інтернет–журналістики, їх роль у сучасному інформаційному просторі та важливість їх розвитку.

Інтернет–видання – це онлайн–платформи [2], що надають користувачам доступ до новин, статей, інформаційних матеріалів та іншого контенту через Інтернет. Вони є електронними аналогами традиційних друкованих видань, таких як газети та журнали, але з великими перевагами: швидким оновленням інформації, можливістю взаємодії з аудиторією та масштабним охопленням читачів по всьому світу. Інтернет–видання забезпечують широкий спектр тематик, від політики та економіки до спорту, культури та науки, задовольняючи інтереси різних аудиторій [3].

Інтернет–журналістика, у свою чергу, є процесом створення, збору, редагування та поширення журналістського контенту в Інтернеті [4]. Вона охоплює всі аспекти журналістики, пристосовані до цифрового середовища. Інтернет–журналістика дозволяє журналістам швидко реагувати на події, публікувати новини в режимі реального часу, використовувати різноманітні мультимедійні елементи, включаючи фотографії, відео, аудіо та інтерактивні елементи.

Одним із ключових аспектів інтернет–журналістики є взаємодія з аудиторією. Інтерактивність та залучення читачів є важливими складовими сучасної журналістики. Коментарі, обговорення на платформах соціальних мереж,

можливість виразити свою думку та поділитися матеріалами з іншими стали невід'ємною частиною інтернет–журналістики. Це дозволяє створювати активну спільноту, взаємодіючу з журналістами, та забезпечує залучення широкого кола людей до процесу створення та обговорення новин.

Розвиток інтернет–видань та інтернет–журналістики має значний вплив на суспільство. Вони змінюють спосіб, яким ми сприймаємо новини та інформацію, розширюють наші можливості доступу до світової подійності, але також поставляють перед нами виклики, пов'язані з вірогідністю та достовірністю інформації. Тому важливо проводити аналіз та дослідження цієї сфери, розуміти її особливості та виявляти способи покращення якості журналістського контенту в онлайн–середовищі.

У даній роботі ми поглибимо наше розуміння предметної сфери інтернет–журналістики, проведемо аналіз її розвитку, вивчимо ключові аспекти та тенденції, а також дослідимо вплив цієї галузі на суспільство.

1.2 Аналіз існуючих аналогів інтернет–видань

Для визначення функцій та характеристик інтернет–видань, які є актуальними для сучасних користувачів, проведено аналіз п'яти новинних сайтів: Укр.нет, Українська правда, TSN (Телевізійні новини), Радіо Свобода та Інтерфакс.

1) Назва: TSN (Телевізійні новини).

Розробник (дистриб'ютор): Компанія «1+1 медіа».

Архітектура: 3–tier web application.

Мова реалізації: PHP, JavaScript, HTML, CSS.

Перелік функцій, характеристик:

1. публікація новин та відеоматеріалів на сайті;
2. можливість перегляду новин та відеоматеріалів онлайн;
3. розділ новин за темами;
4. пошук новин за ключовими словами;

5. підписка на розсилку новин.

Аналіз переваг та недоліків даного ПЗ:

Переваги:

- широкий спектр інформації про події в Україні та світі;
- зручний та швидкий пошук інформації;
- наявність відеоматеріалів;
- можливість підписки на розсилку новин.

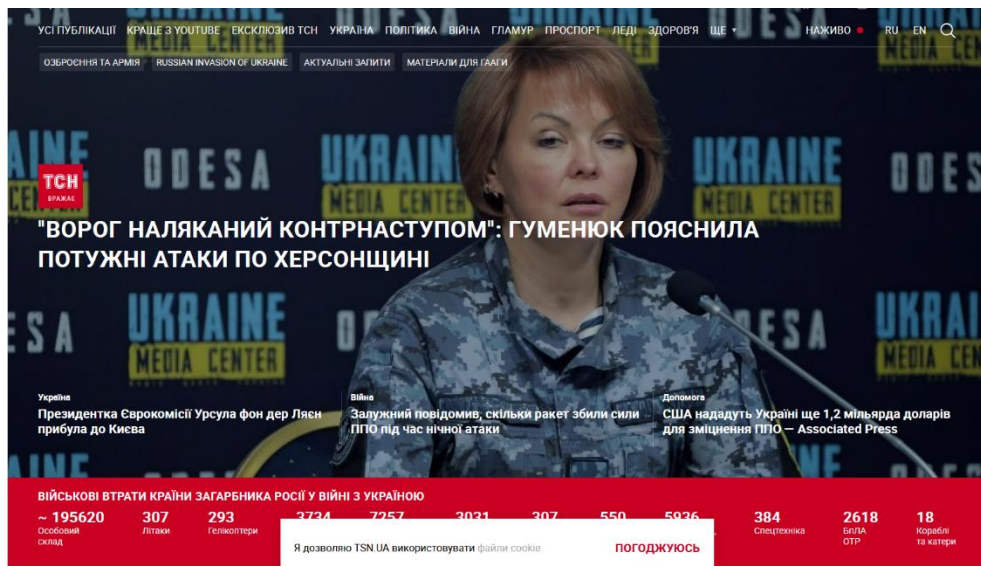


Рисунок 1.1 – Сайт TSN

Недоліки:

- наявність багатьох рекламних банерів;
- загальний дизайн та інтерфейс сайту можна покращити;
- можливість появи «фейкових» новин.

Джерело інформації: <https://tsn.ua/>

2) Назва: Радіо Свобода

Розробник: Радіо Свобода.

Архітектура: 3-tier web application.

Мова реалізації: JavaScript, PHP, HTML, CSS

Перелік функцій та характеристик:

1. новини різних категорій та тематик;

2. радіопрограми та подкасти;
3. статті та репортажі;
4. можливість перегляду відео та прослуховування аудіо–матеріалів;
5. розділи для різних регіонів та країн світу.

Аналіз переваг та недоліків даного ПЗ:

Переваги:

- різноманітність матеріалів та їх доступність;
- багатомовність та охоплення різних регіонів світу;
- можливість прослуховування аудіо– та перегляду відео–матеріалів;
- переважно безкоштовна публікація матеріалів.

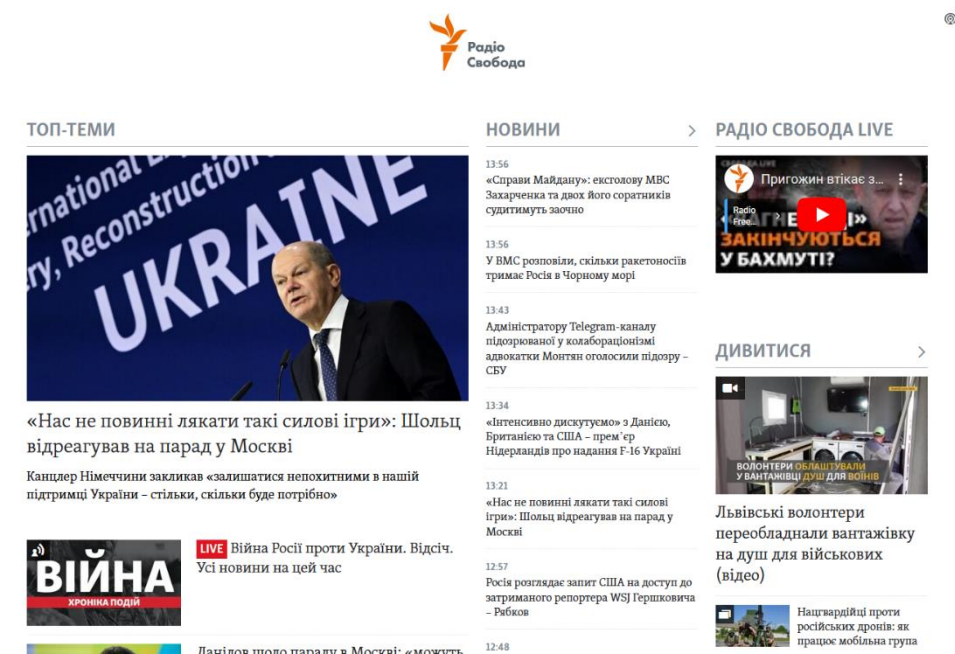


Рисунок 1.2 – Сайт Радіо Свобода

Недоліки:

- велика кількість рекламних банерів та оголошень;
- не завжди високоякісний переклад матеріалів на українську мову.

Джерело інформації: <https://www.radiosvoboda.org/>

3) Назва: Інтерфакс

Розробник: Група компаній Інтерфакс.

Кафедра інженерії програмного забезпечення
Вебзастосунок інтернет–видання

Архітектура: 3–tier web application.

Мова реалізації: PHP.

Перелік функцій, характеристик:

1. новини: Доступ до актуальних новин та аналітики;
2. аналітика: Консолідовані новини про компанії, ринки та фінанси;
3. рейтинги: Дані рейтингів і звіти з найважливішими економічними показниками;
4. трансляції: Потоківне відео та аудіо трансляції новинних подій;
5. архів: Інформація, зібрана протягом багатьох років, з можливістю пошуку та фільтрування.

The screenshot displays the Interfax-Ukraine website interface. At the top, there is a navigation bar with links for 'Про агентство', 'Продукти та послуги', 'Пресцентр', and 'Вхід для клієнтів'. The main header features the 'interfax-УКРАЇНА' logo and a search bar. Below the header, there are several sections: 'СПЕЦТЕМИ' with topics like 'Червоний Хрест', 'Відновлення України', 'Зерно', 'Війна', 'Енергетика', and 'Reform Energy'; a 'СУМАРНІ ВТРАТИ ВОРОГА НА 440 ДЕНЬ ВІЙНИ.' table; and three columns of news articles under the headings 'ОСТАННІ НОВИНИ' and 'ПРЕСРЕЛІЗИ'. The table shows various military losses with icons and numerical data.

СУМАРНІ ВТРАТИ ВОРОГА НА 440 ДЕНЬ ВІЙНИ.												
195620	3734	7257	3031	554	307	308	294	5969	18	955	2618	388
(+552)	(+4)	(+4)	(+12)	(+0)	(+1)	(+0)	(+0)	(+11)	(+0)	(+0)	(+4)	(+4)

Рисунок 1.3 – Сайт Інтерфакс

Аналіз переваг та недоліків даного ПЗ:

Переваги:

- актуальні новини та аналітика;
- консолідовані новини про компанії, ринки та фінанси;

- доступ до рейтингів та звітів з найважливішими економічними показниками;
- потокове відео та аудіо трансляції новинних подій;
- можливість пошуку та фільтрування інформації в архіві.

Недоліки:

- інтерфейс може бути складним у використанні для новачків;
- деякі розділи та функції можуть бути недоступні без підписки.

Джерело інформації: <https://interfax.com.ua/>

4) Назва: Укр.нет;

Розробник (дистриб'ютор): ТОВ «Український портал».

Архітектура: 3-tier web application.

Мова реалізації: PHP, JavaScript.

Перелік функцій та характеристик:

1. новини з України та світу;
2. пошта;
3. пошукова система;
4. погода;
5. біржа праці та оголошення.

Аналіз переваг та недоліків:

Переваги:

- велика кількість новин з різних галузей;
- зручний інтерфейс користувача;
- безкоштовний доступ до всіх функцій.

Недоліки:

- наявність реклами на сайті;
- не завжди точна інформація про погоду;
- можливість поширення фейкових новин.

Джерело інформації: <https://www.ukr.net/ua/>.

Кафедра інженерії програмного забезпечення
Вебзастосунок інтернет-видання

The screenshot shows the Ukr.net website interface. At the top left is the logo 'ukr.net це – мій інтернет!'. Below it are navigation links for 'День Європи', 'День перемоги над нацизмом', and 'Другий світовий війна'. There are sections for 'Пошта' (with a search bar), 'Вибране' (with links to Snoprik, Kinoafisha, Orakul, Proizd ua, Booking, and Robota ua), 'Автосалони України', 'Політика', 'Економіка', 'Медицина', 'Нерухомість', 'Бізнес', 'Подорожі', and 'Війна'. The main content area features a list of news items with dates and headlines, such as '13.59 У Києві відбулася зустріч Зеленського з президенткою Єврокомісії (Рубрика)', '13.47 21 постріл із танка: найманці ПВК "Вагнер" знищили Свято-Благовещенський храм УПЦ (МП) у Бахмуті. ВІДЕО (Цензор.Нет)', and '13.42 Саміт G7: війна в Україні стане головним питанням зустрічі (binn2017)'. On the right side, there is a weather widget for Mikolajiv showing a temperature of +13°C, a search bar, and a 'Погода' section with a table for 'На сьогодні' and 'завтра'. Below that are sections for 'Автомобілі', 'Автосалони України', 'Техніка та прилади', 'Волонтер', 'Офіційні канали', and 'Допомога біженцям'.

Рисунок 1.4 – Сайт Укр.нет

5) Назва: Українська правда.

Розробник: ПАТ «Українська правда».

Архітектура: 3-tier web application.

Мова реалізації: PHP.

Перелік функцій та характеристик:

1. публікація новин;
2. розділ новин за категоріями;
3. пошук за новинами;
4. перегляд статей, блогів, інтерв'ю;
5. онлайн-трансляції.

Аналіз переваг та недоліків:

Переваги:

- велика кількість свіжих новин;
- зручний та швидкий пошук за новинами;
- онлайн-трансляції та звіти з подій.

Недоліки:

- реклама на сайті може заважати користувачам;
- не завжди акуратно оформлені новини;
- відсутність функції перекладу сайту на інші мови.

Джерело інформації: <https://www.pravda.com.ua/>

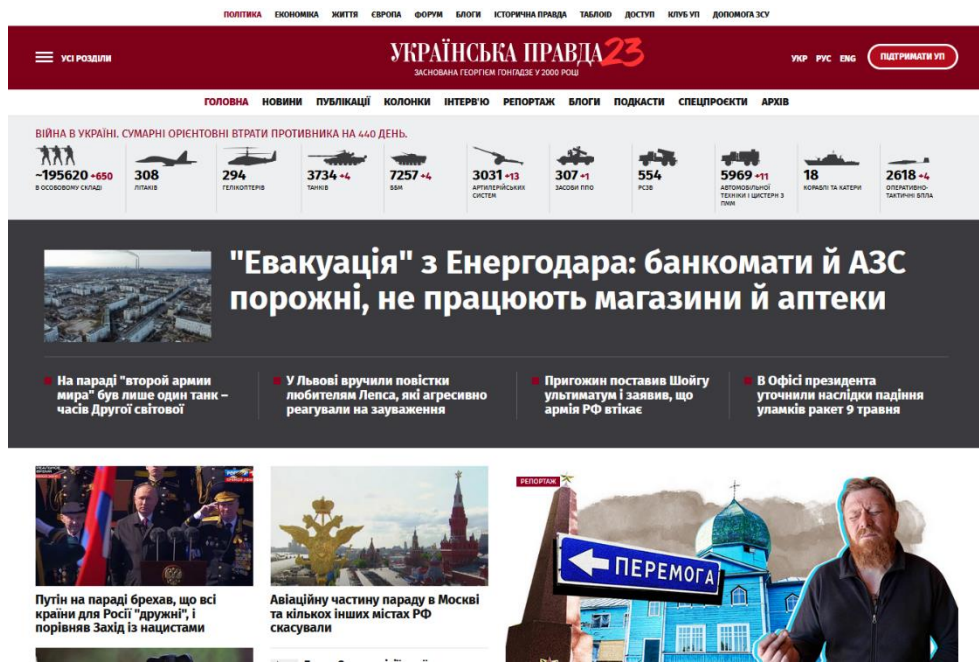


Рисунок 1.5 – Сайт Українська правда

Зважаючи на обрані новинні сайти, розглянутий аналіз визначив наступне:

1. Українська правда – це один з провідних українських інформаційних порталів, який заснований у 2000 році. На сайті представлені новини політики, економіки, культури та спорту. Сайт має велику аудиторію та досить високий рівень довіри в Україні.

2. Український телевізійний новинний сайт TSN (Телевізійні новини) є одним з найбільших в Україні. Сайт є одним з найпопулярніших джерел новин в країні. TSN пропонує новини з різних категорій, включаючи політику, економіку, культуру та спорт.

3. Радіо Свобода – це радіостанція, що транслюється українською мовою в різних країнах. Сайт має значну аудиторію в Україні та пропонує новини з різних категорій, включаючи політику, економіку, культуру, науку та технології.

4. Інтерфакс – це один з провідних новинних агентств в Росії та в СНД, який надає інформацію з різних країн та регіонів. Сайт містить новини з різних сфер, включаючи політику, економіку, науку та технології.

5. Укр.нет – це один з найпопулярніших порталів в Україні, який пропонує новини з різних сфер, включаючи політику, економіку, культуру, спорт, науку та технології. Сайт має велику аудиторію та високий рівень довіри в Україні.

Загалом, можна відзначити, що існуючі аналоги є досить функціональними та зручними для користування, але, водночас, є деякі недоліки, такі як погана оптимізація для мобільних пристроїв, повільна швидкість завантаження сторінок та обмежені можливості взаємодії з користувачами [5]. Враховуючи ці недоліки, можна зробити висновок про необхідність розробки власного вебзастосунку для інтернет-видання з урахуванням сучасних вимог користувачів та функціональних можливостей.

Зважаючи на проведений огляд існуючих аналогів та отриману інформацію, доцільно провести аналіз існуючих аналогів за деякими аспектами.

Функціональні можливості:

- українська правда: Надає розмаїття новинних категорій, має пошук, коментарі, можливість підписки на розсилку новин;
- український телевізійний новинний сайт TSN: Передає новини у відеоформаті, надає можливість коментування та підписки на новини за інтересами;
- радіо Свобода: Зосереджується на новинах з політики, має аудіоверсію новин, коментарі, можливість підписки на розсилку новин;
- інтерфакс: Надає новини з різних сфер, пошук, коментарі, підписку на новини та розсилку;

- укр.нет: Публікує новини з різних галузей, має пошук, коментарі, можливість підписки на новини за інтересами.

Надійність та швидкодія:

- українська правда: Забезпечує стабільну роботу та достатню швидкодію завантаження сторінок;
- український телевізійний новинний сайт TSN: Здатний швидко завантажувати відео та матеріали, працює стабільно;
- радіо Свобода: Має стабільну роботу, швидке відтворення аудіо, низьку частоту помилок та збоїв;
- інтерфакс: Забезпечує стабільність та швидкодію, сторінки завантажуються швидко;
- укр.нет: Завантажує сторінки швидко та має мінімальну кількість помилок.

Мобільна доступність:

- українська правда: Надає мобільну версію та додаток для iOS і Android;
- український телевізійний новинний сайт TSN: Має мобільну версію та додаток для iOS і Android з можливістю перегляду відео;
- радіо Свобода: Присутня мобільна версія та додатки для iOS і Android для прослуховування аудіо новин;
- інтерфакс: Має мобільну версію та додаток для iOS і Android;
- укр.нет: Надає мобільну версію, але не має окремого додатку.

Користувацький інтерфейс:

- українська правда: Має зручний інтерфейс зі зрозумілою навігацією та ергономічним дизайном;
- український телевізійний новинний сайт TSN: Має привабливий інтерфейс з візуально привабливими відеоматеріалами;
- радіо Свобода: Має простий інтерфейс з функціональністю, спрямованою на прослуховування аудіо новин;

- інтерфакс: Має зрозумілий інтерфейс зі зручною навігацією та доступом до різних розділів новин;

- укр.нет: Має простий інтерфейс з можливістю налаштування індивідуальних налаштувань для користувачів.

Вартість абонементу або підписки:

- українська правда: Має різні підписки, включаючи безкоштовну та платні варіанти;

- український телевізійний новинний сайт TSN: Безкоштовний доступ до новин, але можлива підписка для отримання додаткових функцій;

- радіо Свобода: Безкоштовний доступ до новин та аудіо матеріалів;

- інтерфакс: Безкоштовний доступ до новин, але існують платні пакети з додатковими функціями;

- укр.нет: Безкоштовний доступ до новин, але існують платні пакети з додатковими функціями та безрекламними варіантами.

На підставі проведеного аналізу існуючих аналогів вебзастосунку інтернет–видання, можна зробити висновок, що створення власного вебзастосунку інтернет–видання є доцільним. Нижче перераховані основні аргументи, що підтверджують цей висновок:

- унікальність контенту: Створення власного вебзастосунку дозволить представити унікальний контент, який відповідає цільовій аудиторії та враховує її потреби та інтереси. Це надасть конкурентну перевагу та приверне більше користувачів;

- гнучкість та контроль: Власний вебзастосунок дасть можливість повністю контролювати зовнішній вигляд, функціональність та користувацький досвід. Це дозволить адаптувати систему під власні потреби, впроваджувати нові функції та вдосконалення, що позитивно вплине на задоволення користувачів;

- надійність та достовірність: Власний вебзастосунок забезпечить більшу надійність та достовірність інформації, оскільки весь контент буде

контролюватись власною командою професійних журналістів. Це збільшить довіру користувачів та покращить репутацію системи;

– призначення специфічних функцій: Власний вебзастосунок дасть можливість реалізувати специфічні функції, які відповідають потребам та очікуванням користувачів [6]. Наприклад, можливість коментування новин, персоналізація контенту, надання підписок тощо, що підсилить залученість та зацікавленість користувачів;

– конкурентна перевага: Маючи власний вебзастосунок інтернет–видання, компанія отримує конкурентну перевагу на ринку новинних джерел. Це дозволить залучити більше рекламодавців, розширити аудиторію та збільшити прибуткові потоки.

Отже, створення вебзастосунку інтернет–видання має багато переваг, які сприяють залученню та утриманню користувачів, забезпечують контроль, гнучкість та високу якість інформації. Це дозволяє підтримувати конкурентоспроможність на ринку та створювати успішну платформу для поширення новин.

1.3 Специфікація вимог до програмного вебзастосунку

Формулювання специфікації вимог є критичним етапом в процесі розробки програмного забезпечення. Цей процес передбачає збір, аналіз та документування вимог, що визначають функціональність, характеристики та обмеження програмного продукту. Специфікація вимог є вихідним документом, який встановлює основу для подальшого проєктування, розробки та тестування програмного забезпечення.

Перший крок у формулюванні специфікації вимог – це збір вимог від зацікавлених сторін, таких як клієнти, користувачі, бізнес–аналітики та інші [7]. Цей процес може включати в себе проведення інтерв'ю, спостереження, аналіз існуючих процесів та взаємодії з користувачами. Результатом цього етапу є

список вимог, які визначають потреби та очікування від програмного забезпечення.

Далі слід провести аналіз та уточнення зібраних вимог. Це включає виявлення протиріччя та неоднозначностей, визначення пріоритетів вимог, встановлення зв'язків між вимогами та формулювання їх у вимірних, специфічних та перевіряємих термінах. Наприклад, вимога може бути сформульована як «Система повинна забезпечувати швидку відповідь на запити користувачів, які становлять не більше 5 секунд».

Після аналізу вимог необхідно документувати їх у вигляді специфікації вимог. Цей документ повинен містити визначення системи та її цілей, опис функціональних та нефункціональних вимог, опис обмежень та умов роботи системи, вимоги до експлуатації та підтримки, а також вимоги до документації та навчання користувачів. Документ специфікації вимог повинен бути чітким, зрозумілим та зручним для використання розробниками, тестувальниками та іншими учасниками процесу розробки.

В цьому розділі ми детально розглянемо процес формулювання специфікації вимог та надамо практичні поради щодо визначення, аналізу та документування вимог до програмного забезпечення.

1. Призначення та межі проєкту:

1.1 Призначення системи (застосунку), для якої розробляється програмне забезпечення: Призначення системи полягає у створенні вебзастосунку для інтернет–видання з метою оптимізації його функціональних можливостей та поліпшення взаємодії з користувачами.

1.2 Погодження, що ухвалені в програмній документації: У програмній документації ухвалені вимоги до функціональності, продуктивності, зручності використання та безпеки вебзастосунку, який буде розроблено.

1.3 Межі проєкту ПЗ: Межі проєкту ПЗ включають в себе обмеження часу та бюджету проєкту, а також вимоги до функціональності, продуктивності, зручності використання та безпеки вебзастосунку, який буде розроблено.

2. Загальний опис:

2.1 Сфера застосування: Програмне забезпечення розробляється для використання в сфері обробки та аналізу новинних статей з різних джерел. Це може включати в себе роботу в новинах, журналістиці, маркетингу, наукових дослідженнях тощо.

2.2 Характеристики користувачів: Програмне забезпечення призначене для використання різними категоріями користувачів, включаючи професіоналів в галузі новин та журналістики, маркетологів, дослідників тощо. Користувачі повинні мати базові навички роботи з комп'ютером та Інтернетом.

2.3 Загальна структура і склад системи: Програмне забезпечення складатиметься з декількох компонентів, включаючи інтерфейс користувача, систему збору та обробки даних, систему збереження даних [8] тощо. Система повинна бути взаємодійною та забезпечувати гнучкість та розширюваність.

2.4 Загальні обмеження: Програмне забезпечення повинно працювати на різних платформах та операційних системах. Вимоги до апаратного забезпечення повинні бути мінімальними. Також повинні бути дотримані стандарти безпеки та захисту персональних даних.

3. Окремі функції системи:

3.1 Реєстрація користувачів – можливість створення облікового запису користувача та збереження даних про нього;

3.2 Авторизація користувачів – можливість перевірки правильності введення облікових даних користувача та надання доступу до функціоналу системи.

3.3 Пошук новин – можливість здійснювати пошук новин за заданими критеріями, такими як тема, ключові слова, дата, джерело.

3.4 Фільтрація новин – можливість фільтрувати новини за різними параметрами, такими як категорії новин, рейтинг, дата тощо.

3.5 Перегляд новин – можливість перегляду повідомлень з можливістю прокрутки, прокручування, збільшення та зменшення шрифту тощо.

3.6 Підписка на новини – можливість користувачів підписатися на отримання новин з обраних категорій або джерел.

Ці функції дозволяють користувачам здійснювати різні дії в системі, що сприяє комфортному та зручному використанню програмного забезпечення.

Функціональні вимоги на основі функцій системи:

1. Реєстрація користувачів:

1.1 Можливість створення нового користувача з введенням його персональних даних, таких як ім'я, прізвище, адреса електронної пошти, номер телефону.

1.2 Можливість входу в систему під існуючим логіном та паролем.

2. Пошук та перегляд новин:

2.1 Можливість пошуку новин за ключовими словами або за рубриками.

2.2 Можливість перегляду повної статті або її краткого опису.

2.3 Можливість сортування новин за датою, рейтингом, популярністю.

3. Редагування профілю користувача:

3.1 Можливість зміни особистих даних, таких як ім'я, прізвище, номер телефону.

3.2 Можливість зміни адреси електронної пошти та пароля.

4. Оцінка та коментування новин:

4.1 Можливість поставити оцінку новині від 1 до 5 зірок.

4.2 Можливість залишати коментарі під статтею.

5. Підписка на новини:

5.1 Можливість підписатися на рубрики та отримувати повідомлення про нові статті з цих рубрик на електронну пошту або месенджер.

5.2 Можливість налаштування частоти отримання сповіщень про нові статті.

6. Поширення новин:

6.1 Можливість поширення новин через соціальні мережі або месенджери.

6.2 Можливість відправки статті на електронну пошту.

7. Вимоги до інформаційного забезпечення:

7.1 Вимоги до баз даних:

7.1.1 Система повинна забезпечувати збереження та оновлення даних в базі даних.

7.1.2 База даних повинна бути захищена від несанкціонованого доступу до даних.

7.1.3 Система повинна забезпечувати можливість створення резервних копій баз даних.

7.1.4 Система повинна забезпечувати можливість відновлення даних з резервних копій в разі їх втрати або пошкодження.

7.1.5 База даних повинна бути ефективно оптимізована для швидкого доступу до даних.

7.2 Вимоги до інтерфейсу користувача:

7.2.1 Система повинна мати зручний та інтуїтивно зрозумілий інтерфейс користувача.

7.2.2 Система повинна мати можливість персоналізації інтерфейсу користувача відповідно до потреб користувача.

7.2.3 Система повинна мати можливість забезпечувати доступ до інформації через різні пристрої (наприклад, мобільні телефони, планшети, комп'ютери тощо).

7.2.4 Система повинна забезпечувати можливість перегляду статистичних даних та аналітичних звітів за допомогою інтерфейсу користувача.

7.3 Вимоги до забезпечення безпеки:

7.3.1 Система повинна забезпечувати безпеку даних, що передаються по мережі.

7.3.2 Система повинна забезпечувати можливість автентифікації користувачів.

7.3.3 Система повинна забезпечувати можливість контролю доступу до даних.

7.3.4 Система повинна мати можливість виявлення та реагування на вторгнення в систему.

7.3.5 Система повинна забезпечувати захист від шкідливого програмного забезпечення.

8. Вимоги до технічного забезпечення:

Комп'ютер з можливістю роботи в мережі Інтернет, що має наступні параметри:

- процесор: не менше як Intel Core i3;
- оперативна пам'ять: не менше як 4 Гб;
- жорсткий диск: не менше як 128 Гб.

9. Вимоги до програмного забезпечення

9.1 Архітектура програмної системи:

- система повинна мати клієнт–серверну архітектуру;
- клієнтська частина системи повинна бути розроблена на мові JavaScript;
- серверна частина повинна бути написана на мові Python.

9.2 Системне програмне забезпечення:

- система повинна бути розроблена для операційних систем Windows та Linux;
- мінімальні вимоги до операційної системи: Windows 7 або Linux з ядром версії 3.0 і вище;
- мінімальні вимоги до пам'яті: 2 ГБ RAM.

9.3 Мережне програмне забезпечення:

- система повинна підтримувати протоколи HTTP та HTTPS;
- система повинна мати можливість взаємодії з базою даних, яка знаходиться на віддаленому сервері.

9.4 Програмне забезпечення ведення інформаційної бази:

- для зберігання інформації повинна використовуватися PostgreSQL;
- налаштування СКБД повинно бути здійснене відповідно до рекомендацій виробника;

- система повинна забезпечувати резервне копіювання бази даних.

9.5 Мова і технологія розробки ПЗ:

- клієнтська частина повинна бути розроблена з використанням фреймворка React;
- серверна частина повинна бути розроблена з використанням фреймворка Flask;
- мова програмування для серверної частини – Python 3.8.

Висновки до розділу 1

У першому розділі було проведено детальний аналіз різних аспектів предметної області інтернет-журналістики, включаючи її визначення, основні характеристики та значення в сучасному медіа-середовищі.

У процесі огляду галузі були виявлені різноманітні існуючі вебзастосунки інтернет-видань, такі як «Українська правда», «Український телевізійний новинний сайт TSN», «Радіо Свобода», «Інтерфакс» та «Укр.нет». Ці вебзастосунки є популярними та мають свої переваги та недоліки. Вони надають доступ до актуальних новин, але можуть мати обмежені можливості та інтерфейс.

Проведений аналіз існуючих аналогів вебзастосунку інтернет-видання розкрив низку їхніх недоліків. Зокрема, присутність рекламних банерів, обмежена персоналізація, відсутність можливості коментувати новини та незадовільна достовірність інформації.

Були визначені основні функціональні та нефункціональні вимоги до розробки вебзастосунку для інтернет-журналістики. Це включає вимоги до функцій збору, обробки та публікації новин, можливостей взаємодії з користувачами, дизайну та інтерфейсу. Вимоги були сформульовані з урахуванням потреб користувачів та цілей інтернет-журналістики.

Загальний аналіз предметної сфери інтернет-журналістики дозволив зрозуміти її значення та потенціал у сучасному медіа-середовищі. Аналіз існуючих аналогів дав уявлення про те, які функції та можливості можуть бути

реалізовані в програмному вебзастосунку. Специфікації вимог стали основою для подальшого проектування та розробки застосунку.

Результати аналізу предметної сфери, існуючих аналогів та вимог до програмного забезпечення стануть основою для успішної реалізації проєкту та досягнення його поставлених цілей.

Крім того, аналіз предметної сфери інтернет–журналістики дозволив виявити ключові виклики і тенденції, з якими стикаються сучасні видання. Наприклад, зростання популярності мобільних пристроїв і соціальних мереж, швидкий доступ до інформації та зміни споживацьких звичок створюють нові вимоги до інтернет–журналістики.

Дослідження існуючих аналогів інтернет–видань дозволило ідентифікувати їхні сильні та слабкі сторони. Це дало можливість зосередитись на розробці програмного вебзастосунку, який забезпечує найкращі практики та унікальні функціональні можливості, що відповідають потребам і очікуванням цільової аудиторії.

Специфікації вимог до програмного вебзастосунку стали важливим етапом, оскільки вони визначають основні функціональні вимоги та критерії успіху для розробки. Це дозволяє забезпечити якість, надійність та ефективність застосунку, а також враховувати потреби користувачів та вимоги до забезпечення безпеки й конфіденційності інформації.

2 ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ТА ТЕХНОЛОГІЇ РОЗРОБКИ ВЕБЗАСТОСУНКУ

2.1 Обґрунтування вибору мови програмування

При розробці інтернет–видання, насамперед ми зіштовхнемося з використанням HTML (мови гіпертекстової розмітки) і CSS (каскадних таблиць стилів), які є інструментами для створення розмітки сторінок. Проте, для реалізації більш складних функцій, ми повинні вибрати мову програмування, яка найкраще відповідає нашим потребам.

HTML і CSS є мовами розмітки, що використовуються для створення електронних документів та розробки вебсайтів відповідно [9]. Проте, мови вебпрограмування складніші і можуть бути розділені на внутрішні та зовнішні мови веброзробки.

Зазвичай, веброзробка включає розробку на стороні сервера (внутрішня частина), розробку на стороні клієнта (зовнішня частина) та використання технологій баз даних (рис. 2.1) . Як приклад можна взяти LinkedIn – найбільшу професійну мережу у світі. Для забезпечення спілкування кандидатів з різних регіонів, пошуку роботи та отримання нових навичок, LinkedIn був розроблений з використанням трьох різних мов вебпрограмування: JavaScript, Java і Scala, а також була використана розподілена база даних Voldemort для зберігання великої кількості профілів [10].

Отже, мови програмування для веброзробки можна охарактеризувати як «складні логічні інструкції та процеси», які допомагають створювати вебсайти, що відповідають певним вимогам.

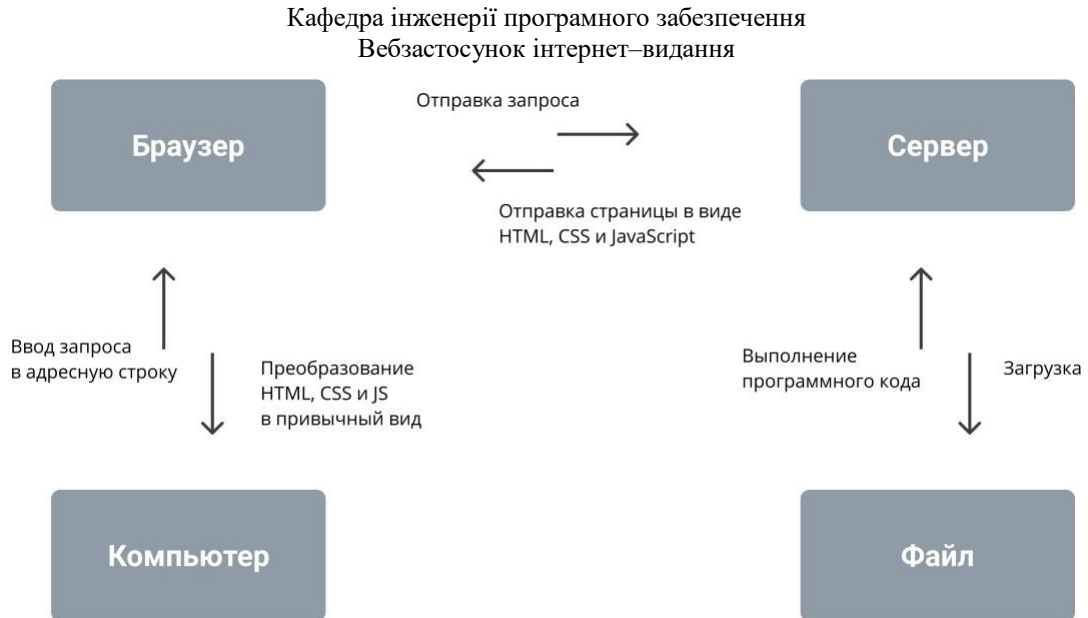


Рисунок 2.1 – Схема роботи типового Web-застосунку

Розглянемо найпопулярніші мови веброзробки і визначимо, яка з них найкраще підійде для вирішення нашого завдання.

JavaScript є однією з найпопулярніших мов програмування у світі технологій, особливо у контексті веброзробки. Він широко використовується як для розробки клієнтського (переднього) інтерфейсу, так і для внутрішнього (серверного) програмування. JavaScript має багато переваг, що пояснюють його популярність (рис. 2.2).

Перш за все, JavaScript надає гнучкість і широкі можливості для розробки клієнтської частини вебзастосунків. Він підтримує динамічну типізацію, що дозволяє змінювати типи даних під час виконання програми. Це дозволяє швидко та зручно працювати з даними.

Другою перевагою JavaScript є підтримка об'єктно-орієнтованого програмування. Ви можете створювати класи, об'єкти, методи та успадкування, що спрощує організацію коду та розвиток вебзастосунків.

Крім того, JavaScript є мовою, що виконується на стороні клієнта (браузера) і сервера (за допомогою платформи Node.js). Це дозволяє створювати повноцінні вебзастосунки, які працюють на обох сторонах. Node.js зокрема став популярним вебфреймворком для серверного програмування на JavaScript.

JavaScript також має велику спільноту розробників, що означає наявність багато документації, розширень та відкритих джерел коду. Це дає можливість швидко знайти рішення для різних завдань і отримати підтримку від інших розробників.

Враховуючи ці фактори, JavaScript є хорошим вибором для розробки вебзастосунків, оскільки він надає розширені можливості для роботи як на стороні клієнта, так і на стороні сервера, має багато функціональних можливостей і підтримується широкою спільнотою розробників.

Фреймворки JavaScript для серверної веброботи: Next.js, Express, MeteorJS і т.д. Популярні вебсайти, що використовують JavaScript: Facebook, Google, eBay і тощо.

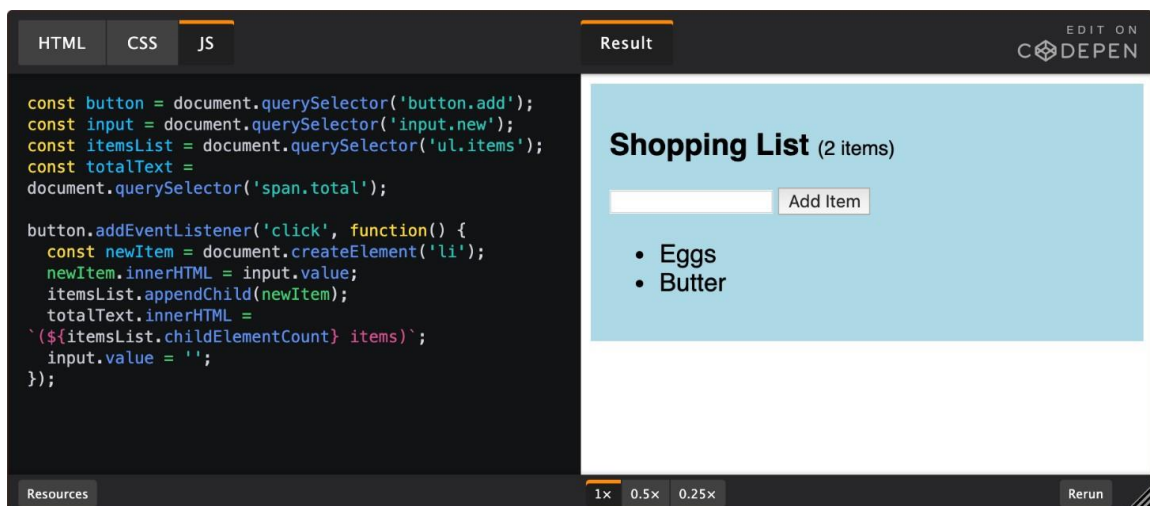


Рисунок 2.2 – Приклад розробки на JavaScript

Python є ще однією популярною мовою програмування, яка широко використовується в веброботі. Ця мова має дуже широкий спектр застосувань і є особливо підходящою для серверної веброботи (рис. 2.3).

Однією з переваг Python є його величезний набір стандартних бібліотек, які дозволяють розробникам простіше та ефективніше виконувати різні завдання. Крім того, Python має читабельний синтаксис, що полегшує розуміння коду і забезпечує більшу продуктивність розробників.

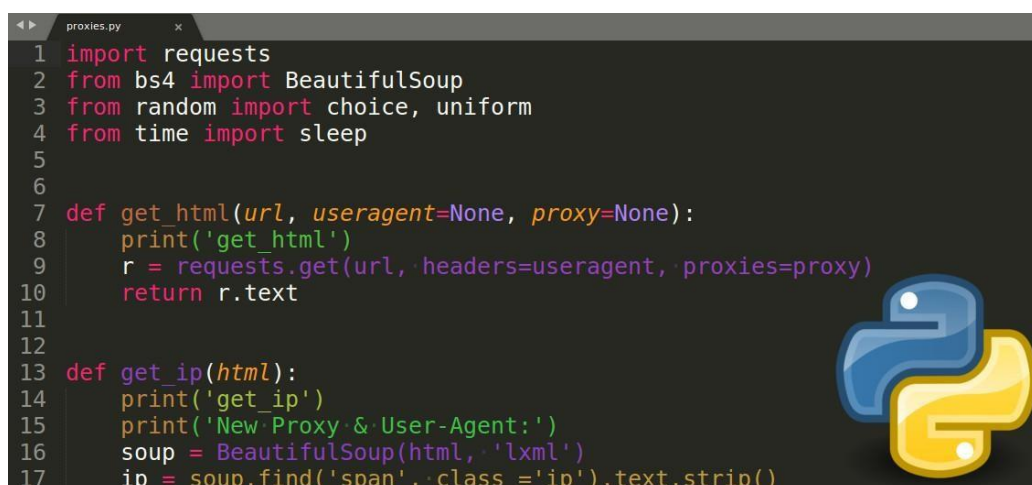
Іншою вагомою перевагою Python є його здатність до простої інтеграції з іншими мовами програмування. Це дозволяє розробникам використовувати спеціалізовані бібліотеки та інструменти, що розширюють можливості вебзастосунків.

Python також підтримує програмування з графічним інтерфейсом, що дозволяє створювати вебдодатки зі зручним та привабливим користувацьким інтерфейсом. Крім того, Python є переносимою мовою, що означає, що вебзастосунки, розроблені на Python, можуть працювати на різних платформах без необхідності великих змін.

Для розробки вебзастосунків на Python використовуються різні фреймворки, такі як Django, Flask, Pyramid і багато інших. Ці фреймворки надають готові інструменти та шаблони для швидкого розроблення вебдодатків з різноманітними функціональними можливостями.

Деякі популярні вебсайти, що використовують Python, включають Spotify, Pinterest, Instacart та багато інших. Це свідчить про потужність і ефективність Python у веброботці.

Враховуючи ці фактори, використання Python в розробці вебзастосунків може бути чудовим вибором, оскільки ця мова надає розробникам багато можливостей, широкий спектр інструментів та велику підтримку спільноти.



```
proxies.py
1 import requests
2 from bs4 import BeautifulSoup
3 from random import choice, uniform
4 from time import sleep
5
6
7 def get_html(url, useragent=None, proxy=None):
8     print('get_html')
9     r = requests.get(url, headers=useragent, proxies=proxy)
10    return r.text
11
12
13 def get_ip(html):
14    print('get_ip')
15    print('New Proxy & User-Agent:')
16    soup = BeautifulSoup(html, 'lxml')
17    ip = soup.find('span', class='ip').text.strip()
```




Рисунок 2.3 – Приклад розробки на Python

PHP є дуже поширеною мовою програмування для веброзробки. Ця мова, яка була створена у 1994 році, спеціально призначена для створення динамічних вебсторінок (рис. 2.4) [11].

Однією з основних переваг PHP є його простота в освоєнні. Синтаксис PHP схожий на мови, такі як C і JavaScript, тому вивчення PHP не є складним для розробників з досвідом у цих мовах.

PHP є кросплатформною мовою, що означає, що вебзастосунки, написані на PHP, можуть працювати на різних операційних системах. Вона підтримує різноманітні бази даних, включаючи MySQL, SQLite та інші, що дає розробникам велику гнучкість у виборі технологій для своїх проєктів.

Один зі сильних боків PHP – це підтримка об'єктно-орієнтованого програмування (ООП). Це дозволяє розробникам створювати більш структурований і модульний код, що полегшує розробку та підтримку вебзастосунків [12].

PHP має широку підтримку спільноти, що робить її привабливим вибором для розробників. Вона має велику кількість сторонніх бібліотек, фреймворків та інструментів, які сприяють швидкому розробленню та покращують продуктивність.

Для розробки вебзастосунків на PHP використовуються різні фреймворки, такі як Laravel, CodeIgniter, Symfony та інші. Ці фреймворки надають готові інструменти та шаблони, що спрощують процес розробки та підвищують якість коду [13].

Деякі популярні вебсайти, що використовують PHP, включають WordPress, MailChimp, Flickr та інші. Це свідчить про широке застосування PHP у веброзробці та його ефективність у створенні різноманітних вебзастосунків.

```
<?php

namespace Email;

class Client
{
    public function send(string $emailAddress, Message $message): bool
    {
        if (!$this->validateParameters($emailAddress, $message)) {
            return false;
        }

        return $this->sendToMailServer([
            'to' => $emailAddress,
            'text' => $message->text(),
        ]);
    }
}
```

Рисунок 2.4 – Приклад розробки на PHP

Ruby є ще однією популярною мовою програмування для веброботи. Одним з основних факторів, які роблять Ruby привабливою, є середовище Ruby on Rails, відоме також як Rails [14]. Rails є вебфреймворком, який дозволяє швидко розробляти вебзастосунки (рис. 2.5).

Однією з особливостей Ruby on Rails є його архітектура Model–View–Controller (MVC), яка надає стандартну структуру для роботи з базою даних, вебслужбами та вебсторінками. Використовуючи Rails, розробники можуть легко працювати з вебстандартами, такими як JSON або XML, для передачі даних, а також використовувати HTML, CSS та JavaScript для взаємодії з користувачем.

Ruby є високорівневою та чисто об'єктно-орієнтованою мовою. Кожне значення в Ruby є об'єктом, а мова не має примітивних типів даних. Вона також має строгую динамічну типізацію, що дозволяє змінювати типи даних залежно від контексту. Ruby також має автоматичне складання сміття для управління пам'яттю, що полегшує життя розробників у цьому аспекті.

Синтаксис Ruby схожий на синтаксис мови Python та Perl. Він є гнучким і легко читається програмістами. Однак, через свою гнучкість, Ruby може вимагати уваги до деталей, оскільки деякі конструкції можуть мати неочікувану поведінку.

Деякі з вебсайтів, які використовують Ruby та Ruby on Rails, включають Github, Shopify, Airbnb, Groupon, GoodReads та Kickstarter. Це свідчить про потужність та ефективність Ruby для веброзробки.

```
1 require "time"
2
3 class InvoiceItem
4   attr_reader :id, :item_id, :invoice_id, :created_at
5
6   attr_accessor :quantity, :unit_price, :updated_at
7
8   def initialize(params)
9     @id = params[:id].to_i
10    @item_id = params[:item_id].to_i
11    @invoice_id = params[:invoice_id].to_i
12    @quantity = params[:quantity].to_i
13    @unit_price = BigDecimal(params[:unit_price])
14    @created_at = Time.parse(params[:created_at].to_s)
15    @updated_at = Time.parse(params[:updated_at].to_s)
16  end
17
18  def unit_price_to_dollars
19    @unit_price.to_f
20  end
21 end
22
```

Рисунок 2.5 – Приклад розробки на Ruby

Java є однією з найпопулярніших мов програмування для серверної веброзробки (рис. 2.6). Вона є об'єктно-орієнтованою мовою, яка знайшла широке застосування в розробці вебзастосунків для підприємств, а також застосунків для Android, настільних систем, наукових проєктів та багатьох інших областей.

Однією з основних переваг використання Java є його принцип «написати один раз і запустити будь-де» (write once, run anywhere). Код Java компілюється в байт-код, який може бути виконаний на будь-якій платформі, що підтримує Java Virtual Machine (JVM), без необхідності у повторній компіляції. Це дозволяє

розробникам писати вебзастосунки на Java, які можуть працювати на різних операційних системах.

Java також підтримує багатопоточність, що дозволяє одночасно виконувати багато потоків і використовувати потенціал багатоядерних процесорів для покращення продуктивності. Вона має вбудовану підтримку концепцій об'єктно-орієнтованого програмування і надає широкий спектр бібліотек з відкритим вихідним кодом для спрощення розробки.

Деякі з популярних фреймворків для веброзробки на Java включають Spring, Struts і Grails. Ці фреймворки надають розробникам потужні інструменти та структури для ефективною розробки вебзастосунків.

Деякі відомі вебсайти, що використовують Java, включають LinkedIn, IRCTC, Yahoo та інші. Це підтверджує велику популярність та надійність Java для серверної веброзробки.

```
def get_symbols(file_name):
    with open(file_name, "r") as in_file:
        records = []
        count = 0
        symbol_set = ""
        for line in in_file:
            symbol_set = symbol_set + line[:-1] + ','
            count = count + 1
            if count % 50 == 0:
                records.append(symbol_set)
                symbol_set = ""
        symbols.append(symbol_set)
    return records
```

Рисунок 2.6 – Приклад розробки на Java

C# дійсно є однією з провідних мов програмування, яка набула значної популярності протягом останніх кількох років. Ця загального призначення мова була розроблена Microsoft з використанням платформи .NET, але вона

застосовується не тільки для серверної веброзробки, але і в багатьох інших областях, таких як розробка програм для Windows і розробка ігор.

C# пропонує широкий набір корисних функцій, які сприяють швидкій компіляції, функціональній сумісності, масштабованості та оновленню компонентів (рис. 2.7). Вона підтримує компонентно-орієнтований підхід до програмування, що сприяє структурованості коду та полегшує розробку. Крім того, C# має багатий набір бібліотек, які сприяють покращенню продуктивності розробників та прискорюють процес розробки.

Одним з основних фреймворків для серверної веброзробки на C# є .NET, який надає розробникам широкі можливості для створення вебзастосунків. .NET включає в себе різні інструменти, бібліотеки і середовища розробки, які допомагають побудувати потужні та масштабовані вебдодатки.

Деякі популярні вебсайти, які використовують C#, включають GoDaddy, Marketwatch, Stack Overflow та інші. Це свідчить про широке використання C# в індустрії та його релевантність у веброзробці.

Таким чином, якщо ви зацікавлені в веброзробці і шукаєте мову програмування, C# є гідним варіантом для вас, зокрема в сполученні з фреймворком .NET.

```
public async Task<User> RegisterAsync(string login, string password)
{
    var hash = Hash(password);
    var exist = await _repository.Where<User>(s => s.Login == login && s.Password == hash).AnyAsync();

    if (exist)
        throw new AppError("User already exist");

    var id = Guid.NewGuid();
    var user = new User(id, login, hash);

    await _repository.SaveAsync(user);
    await _emailService.SendAsync(user.Login, "Добро пожаловать", "Бла бла бла");

    return user;
}
```

Рисунок 2.7 – Приклад розробки на C#

Для аналізу характеристик зазначених у попередньому параграфі мов програмування складемо порівняльні таблиці для проведення порівняння за

такими характеристиками як: парадигми (табл. 2.1), типізація (табл. 2.3), управління пам'яттю (табл. 2.4), об'єктно-орієнтованими можливостями та частотою використання (табл. 2.5).

Таблиця 2.1 – Порівняльна характеристика за парадигмами

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Імперативна	+	+	+	+	+	+
Об'єктно-орієнтована	+	+	+	+	+	+
Функціональна	+/-	+/-	+/-	+/-	+	+
Рефлексивна	+/-	+/-	+	+	+	+
Узагальнене програмування	+	+	+	+	+	+
Логічна	-	-	-	-	-	-
Декларативна	+/-	-	+/-	+	+	+
Розподілена	+/-	+	-	+	+/-	+/-

Таблиця 2.2 – Порівняльна характеристика за типізацією

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Статична типізація	+	+	-	-	-	-
Динамічна типізація	+	-	+	+	+	+
Явна типізація	+/-	+	-	+/-	+/-	-
Неявна типізація	+	-	+	+	+	+
Неявне приведення типів	-	+	+	+	+	+

Таблиця 2.3 – Порівняльна характеристика за можливостями керування пам'яттю

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Створення об'єктів на стеку	+	-	-	-	-	-
Некеровані вказівники	+	-	-	-	-	-
Ручне керування пам'яттю	+	-	-	-	-	-
Збирання сміття	+	+	+	+	+	+

Таблиця 2.4 – Порівняльна характеристика за функціональними можливостями

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Декларації чистоти функцій	-	-	-	-	-	-
First class функції	+	-	+	+	+	+
Анонімні функції	+	+	+	+	+	+
Лексичні замкнення	+	+	+	+	+	+
Часткове застосування	-	-	+	-	+	+
Карування	+	-	+	+	+	+

Таблиця 2.5 – Порівняльна характеристика за об'єктно –
орієнтованими можливостями

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Інтерфейси	+	+	+	+	+	-
Мультиметоди	+/-	-	-	-	-	-
Міксини	-	+	+	+	+	+
Перейменування при спадкуванні	-	-	-	-	-	-
Множинне спадкування	-	-	-	-	+	-

Проведений W3Tech.com аналіз усіх діючих сайтів дозволив отримати наступний графік (рис. 2.8)

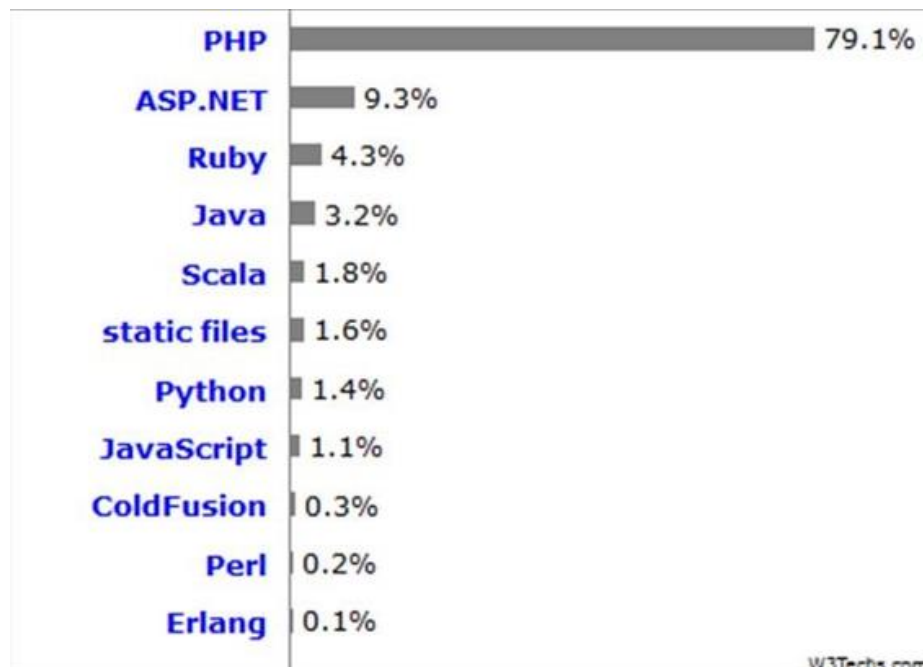


Рисунок 2.8 – Частота використання у існуючих вебсайтах

З проведеного аналізу ми бачимо, що біля 79% сайтів, про бекенд яких відомо, використовують PHP у своїй конструкції .

Для написання візуальної частини інтерфейсу вебзастосунку інтернет–видання було використано HTML як основну розмітку для сторінок, які відображатимуться перед користувачем та CSS для стилізації цих сторінок та надання їм необхідного нам вигляду. В якості мови для виконання бекенд частини було використано PHP, проведений аналіз показав, що ця мова найбільш підходить для виконання поставленого завдання, так як вона має досить зрозумілу структуру, є абсолютно вільно використовуваною, має хороші показники в області продуктивності і відмінно підходить для написання вебзастосунків, що недвозначно підтверджується частотою її застосування.

2.2 Обґрунтування вибору СКБД

Бази даних є необхідною складовою інформаційних систем і відіграють ключову роль у зберіганні, управлінні та маніпулюванні даними [15]. Вони дозволяють систематизувати та організувати великі обсяги даних, забезпечуючи швидкий доступ до них. Бази даних існують 2-х типів: реляційні та нереляційні [16].

Наведені приклади ілюструють різноманітні сфери використання баз даних. Онлайн телефонний довідник використовує базу даних для зберігання інформації про людей та їх контактні дані. Постачальник електроенергії використовує базу даних для управління рахунками та вирішенням питань клієнтів. Facebook використовує базу даних для зберігання даних про користувачів, їх дії, повідомлення, рекламу тощо.

Бази даних дозволяють структурувати дані, виконувати операції зчитування, запису та модифікації даних, а також робити складні запити для отримання необхідної інформації [17]. Різні системи керування базами даних (СКБД), такі як MySQL, PostgreSQL, Oracle, Microsoft SQL Server, надають засоби для роботи з базами даних з різними вимогами щодо масштабу, надійності та продуктивності [18].

Ефективне використання баз даних вимагає проєктування правильної структури бази даних, оптимізації запитів та забезпечення безпеки даних. Добре спроектована база даних дозволяє ефективно управляти великим обсягом даних і забезпечує надійність та швидкодію системи.

Таким чином, бази даних є необхідною складовою будь-якої інформаційної системи і дозволяють забезпечити ефективне управління даними в різних галузях та сферах застосування.

Давайте розглянемо найпопулярніші системи управління базами даних та з'ясуємо, яка з них найкраще підходить для вирішення нашого завдання.

Одна з найпопулярніших систем реляційних баз даних – MySQL. Початково вона була відкритою системою з відкритим вихідним кодом, але зараз належить корпорації Oracle. MySQL широко використовується як основа прикладного програмного забезпечення LAMP, що означає, що вона є складовою стеку Linux, Apache, MySQL та Perl/PHP/Python [19]. MySQL побудована на мовах програмування C і C++, і добре працює на різних системних платформах, таких як Windows, Linux, MacOS, IRIX та інші.

Основні переваги MySQL полягають у тому, що вона доступна для безкоштовного завантаження для спільноти. Версія MySQL Community Edition з базовим набором інструментів є хорошим варіантом для початківців. Синтаксис MySQL досить простий, а сама система має невелику складність. Розробники навіть називають MySQL «людською» базою даних, оскільки її структура і стиль дуже зрозумілі. MySQL часто використовується разом з мовою програмування PHP.

Таким чином, MySQL є популярною системою управління базами даних з простим синтаксисом та доступним варіантом для початківців. Вона добре підходить для веброботи, особливо в тандемі з PHP.

MariaDB є форком MySQL з відкритим вихідним кодом, який також надає комерційну підтримку. Він працює під ліцензією GNU та має ті самі команди, API та бібліотеки, що й MySQL.

Основні переваги MariaDB включають:

- шифрування: MariaDB надає функції безпеки, такі як внутрішня безпека, перевірка паролів, аутентифікація PAM та LDAP, Kerberos та ролі користувачів. Зашифровані табличні простори, таблиці та журнали забезпечують надійний рівень захисту даних;

- широкий функціонал: MariaDB пропонує багато нових функцій, включаючи підтримку ГІС для зберігання координат та запитів щодо місцезнаходження. Динамічні стовпці дозволяють обробляти дані SQL та NoSQL в одній системі для різноманітних потреб [20];

- висока продуктивність: Незважаючи на те, що MariaDB походить від двигуна MySQL, вона значно покращила продуктивність. Розширені функції оптимізації поліпшують керування пулом потоків та обробку даних. Наприклад, при видаленні рядків з таблиці, операційна система відразу звертається до вільного простору, усуваючи прогалини в табличному просторі.

Отже, MariaDB є потужною системою управління базами даних зі шифруванням, широким функціоналом та високою продуктивністю. Вона може бути гарним вибором для проєктів, які потребують надійного зберігання та обробки даних.

Oracle – це система управління реляційними базами даних, розроблена та підтримувана компанією Oracle. Вона пропонує кілька моделей даних, включаючи документ, граф, реляційну модель та модель «ключ–значення» в межах однієї бази даних. Останні версії Oracle зосередилися на хмарних обчисленнях.

Основні переваги Oracle включають:

- інновації для щоденного робочого процесу: Oracle неперервно впроваджує нові технології хмарних обчислень, забезпечуючи активний захист даних, поділ, покращене резервне копіювання та відновлення. Кожен новий випуск спрямований на інновації та підтримку;

– потужна технічна підтримка та документація: Oracle надає високоякісну підтримку клієнтів та вичерпну технічну документацію. Це дозволяє знайти рішення для різних проблем та запитів;

– велика ємність: Oracle забезпечує можливість розміщувати та обробляти величезні обсяги даних. Запровадження функції мультиарендності спрощує управління багатьма базами даних та забезпечує ефективне управління ними.

Отже, Oracle є потужною системою управління базами даних, яка надає інновації для щоденної роботи, має потужну технічну підтримку та дозволяє обробляти великі обсяги даних. Вона може бути використана для різноманітних проєктів з вимогами до безпеки, масштабованості та продуктивності.

Проведемо аналіз усіх випадків використання СКБД та отримаємо наступний графік (рис. 2.9).

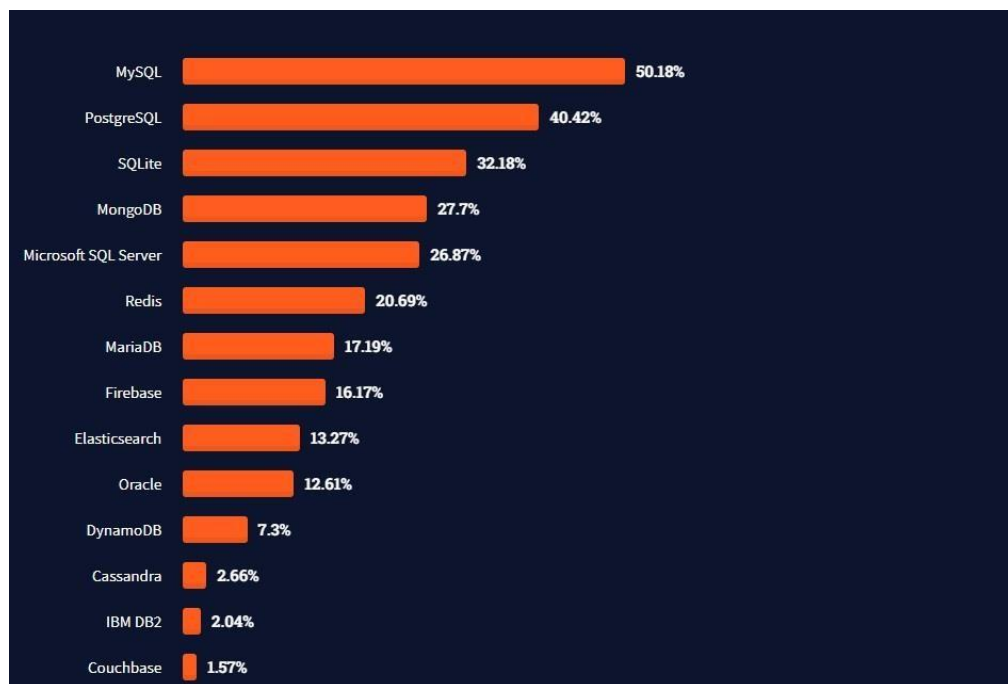


Рисунок 2.9 – Аналіз використання СКБД

Як бачимо, близько 50% сайтів, про бекенд яких відомо, використовують базу даних MySQL у своїй конструкції.

В якості СКБД була обрана MySQL, так як вона достатньо проста у використанні, має великий функціонал, безпечна, має хорошу швидкість роботи, а

також має інтерфейс для взаємодії з PHP. Як IDE для розробки було вибрано PHPStorm, а для візуалізації роботи з базою даних PhpMyAdmin.

Висновки до розділу 2

У другому розділі були розглянуті різні інструменти та технології, які можна використовувати для розробки вебзастосунків. З урахуванням аналізу різних варіантів, моїми вибраними інструментами для розробки вебзастосунків є MySQL, PhpMyAdmin, VSCode, HTML та CSS.

MySQL є надійною та популярною системою управління реляційними базами даних, яка забезпечує ефективне зберігання та маніпулювання даними. Він має простий синтаксис та легко поєднується з мовами програмування, зокрема з PHP, що робить його ідеальним вибором для веброзробки.

PhpMyAdmin є потужним інструментом для адміністрування бази даних MySQL. Він надає зручний вебінтерфейс, який дозволяє легко керувати базою даних, створювати таблиці, виконувати запити та імпортувати/експортувати дані. PhpMyAdmin спрощує роботу з базою даних і полегшує процес розробки.

VSCode (Visual Studio Code) є популярним редактором коду, який надає розширені можливості для розробки вебзастосунків. Він підтримує різні мови програмування, включаючи HTML, CSS та PHP, і має багато корисних функцій, таких як автодоповнення коду, налагоджувач та інтеграція з контролем версій. VSCode допомагає підвищити продуктивність розробників і забезпечує зручне середовище для написання коду.

Таким чином, використання MySQL, PhpMyAdmin, VSCode, HTML та CSS у розробці вебзастосунків є доцільним вибором, що дозволить досягти успішних результатів у створенні вебзастосунків.

3 ПРОЄКТУВАННЯ СИСТЕМИ З ВИКОРИСТАННЯМ МОВИ UML

Мова UML (Unified Modeling Language) забезпечує стандартизований нотаційний інструментарій для моделювання систем, дозволяючи розробникам глибше розуміти, проєктувати та комунікувати різні аспекти системи.

Використання мови UML дозволяє створювати діаграми, що ілюструють структуру, поведінку та взаємодію компонентів системи. Це включає діаграми класів, діаграми послідовності, діаграми діяльності та багато інших. Кожна діаграма UML служить для визначення конкретного аспекту системи та допомагає в розумінні взаємодії між компонентами [21].

Перевагою використання мови UML є можливість створення абстрактної моделі системи, яка дозволяє зосередитися на важливих аспектах та ігнорувати незначні деталі. Це сприяє виявленню та вирішенню проблем ще на ранніх етапах розробки, забезпечує більшу точність та однозначність у сприйнятті системи розробниками та зацікавленими сторонами.

Крім того, використання UML полегшує комунікацію між розробниками, дизайнерами, тестувальниками та іншими учасниками процесу розробки. Стандартизований набір діаграм дозволяє створювати моделі, які зрозумілі та доступні для всіх членів команди.

У процесі проєктування системи з використанням мови UML, розробники можуть отримати більш чітке уявлення про вимоги та функціональність системи, а також виявити можливі проблеми чи неузгодженості ще до початку реалізації. Це допомагає знизити ризик помилок та забезпечити ефективність у розробці програмного забезпечення.

Таким чином, використання мови UML в проєктуванні системи є важливим інструментом, що сприяє розумінню, комунікації та аналізу системи, а також покращує ефективність та якість процесу розробки.

3.1 Діаграма прецедентів

На рисунку 3.1 наведено діаграму прецедентів для вебзастосунку інтернет-видання. Ця діаграма відображає взаємодію акторів (користувач, редактор, адміністратор) з вебзастосунком інтернет-видання. Користувач може переглядати статті, шукати статті, реєструватися, авторизуватися та коментувати статті. Редактор може створювати, редагувати та видаляти статті. Адміністратор відповідає за управління користувачами та модерацію коментарів.



Рисунок 3.1 – Діаграма прецедентів

Загальна структура діаграми прецедентів для вебзастосунку інтернет-видання така:

1. Актори:

- користувач: основний актор, який використовує вебзастосунок для перегляду статей, пошуку, реєстрації, авторизації та коментування статей;
- редактор: актор, який відповідає за створення, редагування та видалення статей;

– адміністратор: актор, який має повний доступ до управління користувачами та модерації коментарів.

2. Прецеденти (функціональні можливості):

- переглядати статті: користувач може переглядати доступні статті;
- пошук статей: користувач може шукати статті за певними критеріями;
- реєстрація: користувач може створити обліковий запис у вебзастосунку;
- авторизація: користувач може увійти до свого облікового запису для отримання розширених можливостей;
- коментувати статті: користувач може залишати коментарі під статтями;
- створювати статті: редактор може створювати нові статті;
- редагувати статті: редактор може вносити зміни до вже існуючих статей;
- видалити статті: редактор може видаляти статті, які більше не потрібні;
- управління користувачами: адміністратор має можливість керувати обліковими записами користувачів;
- модерація коментарів: адміністратор може модерувати коментарі, що дозволяє забезпечити безпеку та адекватність обговорення.

3.2 Опис варіантів використання

Сценарії використання дозволяють сформулювати та задокументувати вимоги для полегшення розробки та підтримки застосунку. Для демонстрації можливих форм написання сценаріїв, перші 3 сценарії представлені в:

- короткій формі;
- поверхневій формі;
- повній формі.

Сценарій №1(коротка форма):

Назва: Перегляд останніх новин.

Опис: Користувач відкриває додаток і бачить список останніх новин, впорядкованих за датою публікації. Він може переглядати заголовки новин та клікати на них, щоб перейти до повної статті.

Сценарій №2(поверхнева форма):

Назва: Пошук та перегляд статей за категоріями.

Опис: Користувач, бажаючи знайти цікаву статтю на вебзастосунку інтернет–видання, використовує функціонал пошуку та перегляду статей за категоріями.

Головний сценарій (успішний):

1. користувач відкриває вебзастосунок інтернет–видання та переходить на сторінку пошуку статей;
2. система відображає список доступних категорій статей;
3. користувач обирає категорію, яка його цікавить;
4. система відображає списки статей, відносяться до обраної категорії;
5. користувач переглядає заголовки статей та обирає конкретну статтю для перегляду;
6. система відкриває обрану статтю та відображає її повний зміст на сторінці;
7. користувач переглядає статтю, читає її та за потреби може скористатись іншими функціями, такими як коментування або поділитися статтею з іншими користувачами;
8. користувач може повернутись до списку статей або виконати новий пошук за іншими категоріями.

Альтернативний сценарій:

1. користувач обирає категорію, але в цій категорії відсутні статті;
2. система не знаходить результатів, що відповідають пошуковому запиту користувача, та відображає повідомлення про відсутність результатів.

Сценарій №3(повна форма):

Назва: Пошук та перегляд статей за категоріями.

Область застосування: Вебзастосунок інтернет–видання.

Рівень: ціль користувача.

Головний актор: Користувач.

Зацікавлені сторони:

Користувач: Бажає знайти цікаву статтю в обраній категорії та переглянути її повний зміст.

Вебзастосунок: Забезпечує функціонал пошуку та перегляду статей за категоріями.

Передумови:

Користувач має доступ до вебзастосунку інтернет–видання.

Вебзастосунок містить базу даних з категоріями та статтями.

Гарантії успіху: Користувач може успішно знайти та переглянути статтю в обраній категорії.

Основний успішний сценарій:

1. користувач відкриває вебзастосунок інтернет–видання;
2. користувач переходить на сторінку пошуку статей;
3. система відображає список доступних категорій статей;
4. користувач обирає категорію, яка його цікавить;
5. система відображає списки статей, відносяться до обраної категорії;
6. користувач переглядає заголовки статей та обирає конкретну статтю для перегляду;
7. система відкриває обрану статтю та відображає її повний зміст на сторінці;
8. користувач переглядає статтю, читає її та може скористатись іншими функціями, такими як коментування або поділитися статтею з іншими користувачами.

Розширення: 1а. Користувач обирає категорію, але в цій категорії відсутні статті.

- система відображає повідомлення про відсутність статей у даній категорії;
- спеціальні вимоги:
 - доступ до бази даних з категоріями та статтями;
 - відображення статей з обраної категорії на сторінці;

- збереження інформації про користувачів, їхні вибрані категорії та переглянуті статті;
- можливість коментувати статті та взаємодіяти з іншими користувачами;
- можливість поділитися статтею на соціальних мережах або іншими засобами спілкування.
- список технологій та варіацій даних:
 - вебзастосунок інтернет–видання розроблений на вебтехнологіях, таких як HTML, CSS, JavaScript;
 - використання серверного програмування (наприклад, PHP, Python, Ruby) для обробки запитів користувачів та доступу до бази даних;
 - база даних для збереження категорій, статей, користувачів та пов'язаної інформації (може бути реляційною або нереляційною, залежно від вимог проєкту).

Частота виникнення: Використовується відразу після відкриття вебзастосунку інтернет–видання та протягом всього періоду користування.

Додатково:

- забезпечення безпеки та конфіденційності користувачів;
- можливість розширення функціоналу пошуку та перегляду статей (наприклад, додавання фільтрів, сортування тощо);
- використання адаптивного дизайну для підтримки різних пристроїв та розмірів екранів.

Сценарій №4:

Назва: Перегляд статей.

Діючі особи: Користувач, система.

Мета: Ознайомлення зі статтями на вебсайті.

Успішний сценарій:

- користувач відкриває вебзастосунок;
- користувач переходить на сторінку зі списком статей;

– користувач обирає потрібну статтю для перегляду.

Результат: Користувач може прочитати вміст обраної статті.

Сценарій №5:

Назва: Пошук статей.

Діючі особи: Користувач.

Мета: Знаходження статей за заданими критеріями.

Успішний сценарій:

- користувач відкриває вебзастосунок;
- користувач вводить ключові слова або фільтри для пошуку статей;
- користувач натискає кнопку «Пошук».

Результат: Користувач отримує список статей, що відповідають заданим критеріям.

Сценарій №6:

Назва: Реєстрація нового користувача.

Діючі особи: Новий користувач.

Мета: Створення облікового запису на вебсайті.

Успішний сценарій:

- користувач відкриває вебзастосунок;
- користувач обирає опцію реєстрації;
- користувач вводить необхідну інформацію (ім'я, електронну пошту, пароль);
- користувач натискає кнопку «Зареєструватися».

Результат: Новий користувач успішно створює обліковий запис на вебсайті.

Сценарій №7:

Назва: Авторизація користувача.

Діючі особи: Зареєстрований користувач.

Мета: Увійти в обліковий запис на вебсайті.

Успішний сценарій:

- користувач відкриває вебзастосунок;

- користувач обирає опцію авторизації;
- користувач вводить свою електронну пошту та пароль;
- користувач натискає кнопку «Увійти».

Результат: Користувач успішно авторизується і отримує доступ до свого облікового запису.

Сценарій №8:

Назва: Додавання коментаря до статті.

Діючі особи: Зареєстрований користувач.

Мета: Залишити коментар під статтею.

Успішний сценарій:

- користувач відкриває сторінку зі статтею;
- користувач прокручує сторінку до розділу коментарів;
- користувач вводить свій коментар в поле введення;
- користувач натискає кнопку «Відправити коментар».

Результат: Коментар користувача додається до статті та відображається для інших користувачів.

3.3 Діаграма діяльності

На рисунку 3.2 наведено діаграму діяльності для вебзастосунку інтернет–видання. Ця діаграма діяльності відображає послідовність дій у вебдодатку для інтернет-журналістики. Основна мета додатку полягає в забезпеченні користувачам можливості авторизуватися, здійснювати пошук та переглядати статті.

Процес розпочинається з входу користувача до системи, після чого перевіряється правильність авторизації. Якщо авторизація успішна, виконується завантаження користувальницького профілю, і користувач отримує доступ до основного функціоналу.

Далі користувач може виконувати пошук інформації. Він обирає категорію і виконує пошук, після чого результати відображаються. Користувач може обрати конкретну статтю для перегляду, після чого відкривається відповідна стаття.

Процес завершується, коли користувач вирішує покинути додаток або здійснити інші дії.

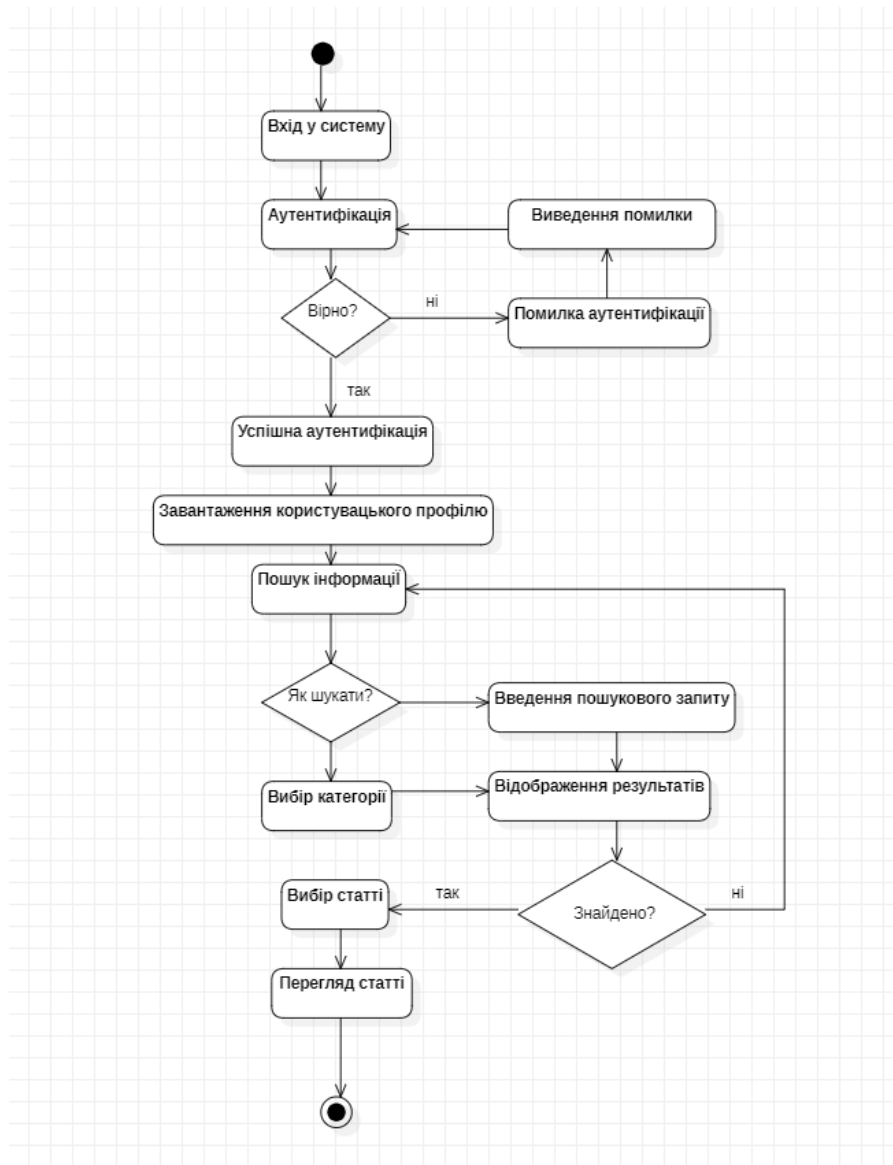


Рисунок 3.2 – Діаграма створення новин

Діаграма діяльності надає загальний огляд послідовності дій, які користувач може виконувати в вебдодатку для інтернет-журналістики. Вона допомагає

зрозуміти, як користувач взаємодіє з системою та які операції можуть бути виконані.

3.4 Діаграма станів та переходів

На рисунку 3.3 наведено діаграму станів та переходів вебзастосунку інтернет-видання. Зважаючи на розмір і складність системи «вебзастосунок інтернет-видання», для опису процесу її функціонування в цілому можна використати діаграму станів, яка відображатиме основні стани системи та переходи між ними. Statechart diagrams або діаграми станів зазвичай демонструють певні сценарії, які будуть виконуватися під час функціонування системи [22].

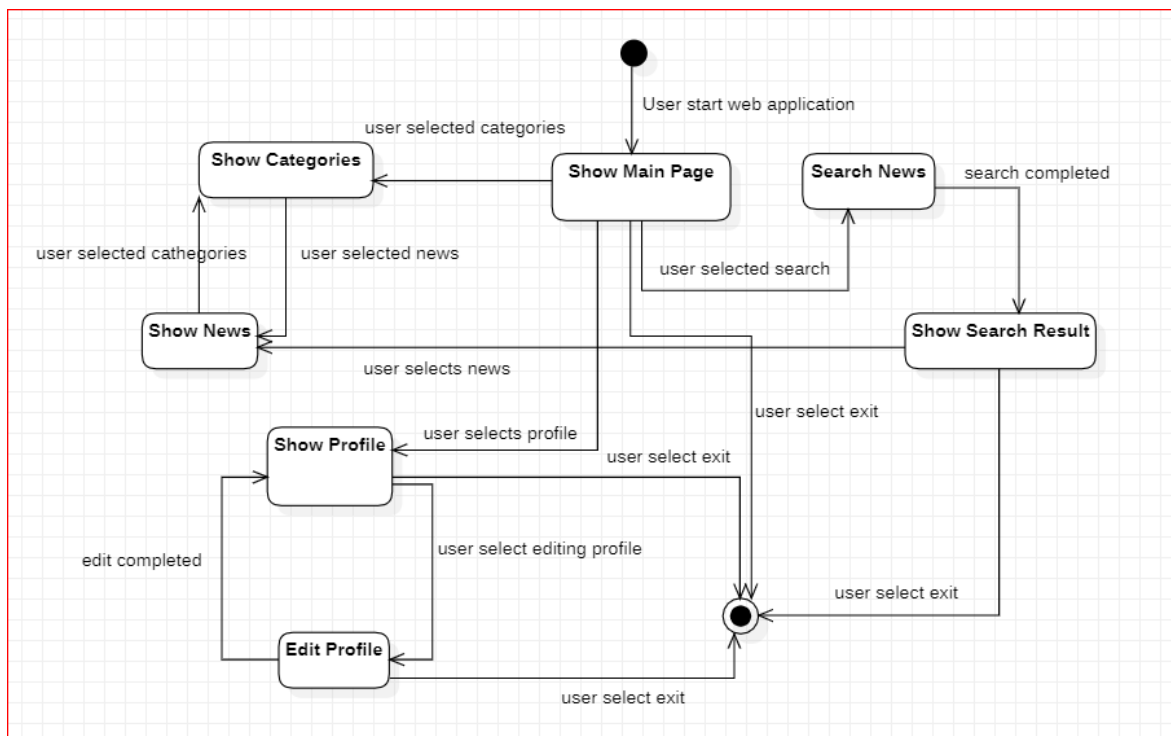


Рисунок 3.3 – Діаграма станів для опису процесу функціонування обраної системи

Основні стани системи:

- початок: Стартовий стан системи;
- відображення Головної Сторінки: Стан, в якому система відображає головну сторінку інтернет-видання;

- відображення Категорій: Стан, в якому система відображає список категорій новин;
- відображення Новини: Стан, в якому система відображає окрему новину;
- пошук Новин: Стан, в якому користувач може виконати пошук новин за певними критеріями;
- відображення Результатів Пошуку: Стан, в якому система відображає результати пошуку новин;
- відображення Профілю Користувача: Стан, в якому система відображає профіль користувача зі збереженими налаштуваннями;
- редагування Профілю Користувача: Стан, в якому користувач може редагувати свій профіль;
- вихід: Кінцевий стан системи, в якому користувач завершує роботу з системою.

Переходи між станами в системі:

1. початок → Відображення Головної Сторінки: Початок роботи системи, коли користувач відкриває додаток і система переходить до відображення головної сторінки;
2. відображення Головної Сторінки → Відображення Категорій: Користувач обирає категорію новин і система переходить до відображення списку новин за цією категорією;
3. відображення Категорій → Відображення Новини: Користувач обирає конкретну новину зі списку, і система переходить до відображення окремої новини;
4. відображення Головної Сторінки → Пошук Новин: Користувач вибирає опцію пошуку новин і система переходить до стану пошуку новин;
5. пошук Новин → Відображення Результатів Пошуку: Після введення критеріїв пошуку користувачем, система відображає результати пошуку новин;
6. відображення Результатів Пошуку → Відображення Новини: Користувач обирає конкретну новину з результатів пошуку і переходить до її відображення;

7. відображення Головної Сторінки → Відображення Профілю Користувача: Користувач обирає опцію перегляду свого профілю і система переходить до відображення профілю користувача;

8. відображення Профілю Користувача → Редагування Профілю Користувача: Користувач обирає опцію редагування свого профілю і переходить до стану редагування профілю;

9. редагування Профілю Користувача → Відображення Профілю Користувача: Користувач завершує редагування профілю і система повертається до відображення профілю користувача;

10. будь-який стан → Вихід: Користувач завершує роботу з системою і виходить з неї.

Крім того, важливо зазначити, що при перебуванні об'єкта чи модельованої системи в початковому стані, її поведінка може супроводжуватися реалізацією певних внутрішніх дій [23].

Ці переходи між станами демонструють основні етапи функціонування системи, що розробляється та взаємодію користувача з нею.

3.5 Структура бази даних

Структура бази даних, зображена на діаграмі класів, включає наступні класи та їх зв'язки (рис. 3.5):

1. Клас “Users”:

- user_id: int (унікальний ідентифікатор користувача);
- username: string (ім'я користувача);
- email: string (адреса електронної пошти);
- password: string (хешований пароль);
- role: string (роль користувача).

2. Клас “Article”:

- article_id: int (унікальний ідентифікатор статті);
- title: string (назва статті);

- content: string (вміст статті);
- created_at: datetime (дата створення статті);
- rating: int (рейтинг статті);
- author: Користувач (зв'язок з класом “Users” – автором статті);
- category: Категорія (зв'язок з класом “Categories” – категорією статті).

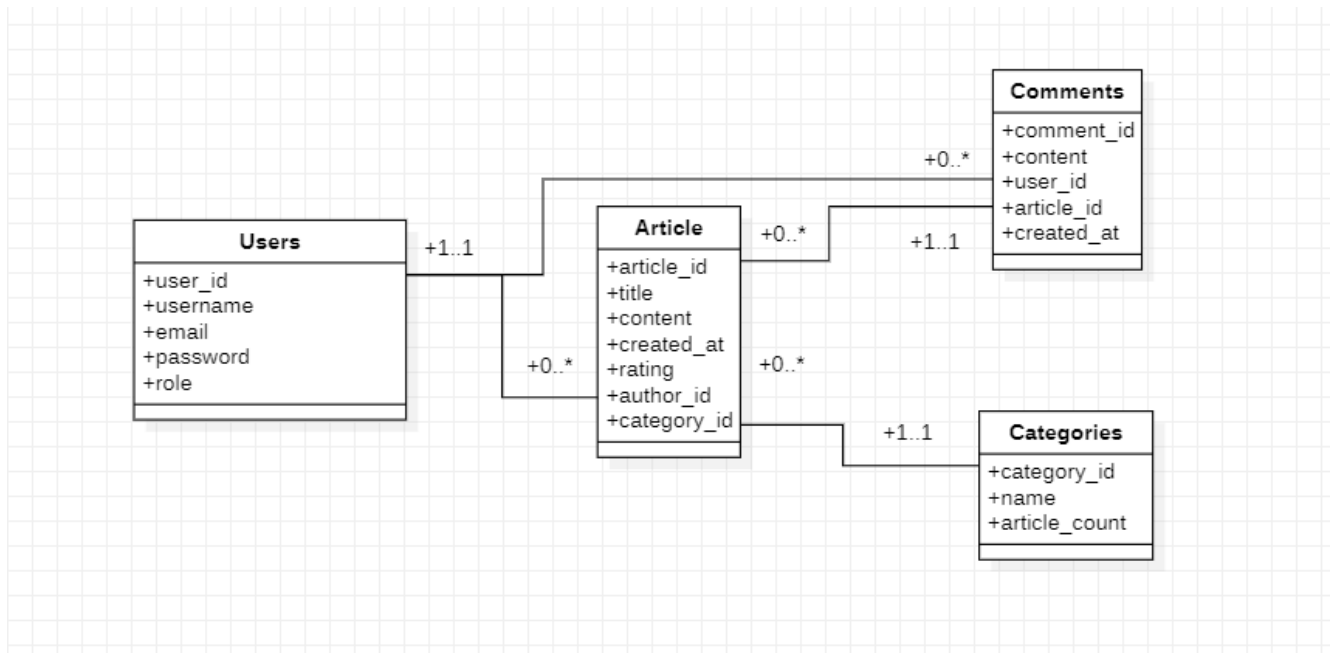


Рисунок 3.5 – Структура бази даних

3. Клас “Categories”:

- category_id: int (унікальний ідентифікатор категорії);
- name: string (назва категорії);
- article_count: int (кількість статей у категорії).

4. Клас “Comments”:

- comment_id: int (унікальний ідентифікатор коментаря);
- content: string (вміст коментаря);
- created_at: datetime (дата коменту);
- user: Користувач (зв'язок з класом “Users” – автором коментаря);
- article: Стаття (зв'язок з класом “Article” – статтею, до якої залишено коментар).

Зв'язки:

- у класі “Article” є зв'язок «один-до-багатьох» з класом “Users” (автор статті може мати багато статей);
- у класі “Article” є зв'язок «багато-до-одного» з класом “Categories” (одна категорія може мати багато статей, але кожна стаття належить до однієї категорії);
- у класі “Comments” є зв'язок «багато-до-одного» з класом “Users” (один користувач може залишити багато коментарів, але кожен коментар належить до одного користувача);
- у класі “Comments” є зв'язок «багато-до-одного» з класом “Article” (одна стаття може мати багато коментарів, але кожен коментар належить до однієї статті).

Ця структура бази даних дозволяє зберігати інформацію про користувачів, статті, категорії та коментарі, а також встановлювати зв'язки між ними для зручного управління та отримання необхідної інформації.

3.6 Проєктування інтерфейсу користувача

Дизайн вебзастосунку відіграє важливу роль у залученні та задоволенні користувачів. Він створює візуальну привабливість, забезпечує зручність взаємодії та допомагає досягти поставлених цілей. Огляд дизайну вебзастосунку включає в себе оцінку різних аспектів дизайну, таких як загальна структура, використання кольорів, шрифтів, графічних елементів та їх гармонія, навігаційні елементи, розташування контенту та багато іншого.

Один з ключових аспектів дизайну вебзастосунку – це його структура. Грамотно побудована структура дозволяє користувачам швидко зорієнтуватись на сайті, знайти потрібну інформацію та здійснити необхідні дії. Чітко виділені розділи, логічні підменю та зручна навігація допомагають забезпечити зручність взаємодії та відсутність заплутаності.

Колірна палітра та використання кольорів є ще одним важливим аспектом дизайну вебзастосунку. Кольори можуть впливати на настрій користувачів, передавати брендову ідентичність та створювати візуальну привабливість. Важливо вибирати кольори, які гармонійно поєднуються між собою та відповідають задуманому стилю та концепції вебзастосунку.

Шрифти також впливають на загальний вигляд та зручність сприйняття вебзастосунку. Вони можуть передавати настрій, стиль та допомагати утримувати увагу користувачів. Важливо вибирати читабельні шрифти, які зберігають свою читабельність навіть на різних пристроях та розмірах екранів.

Графічні елементи, такі як фотографії, ілюстрації, іконки тощо, також мають значення у дизайні вебзастосунку. Вони можуть доповнювати контент, передавати інформацію, створювати емоційне зв'язок з користувачем. Важливо обрати графічні елементи, які відповідають загальному стилю та концепції вебзастосунку.

Огляд дизайну вебзастосунку допомагає визначити, наскільки добре впроваджено дизайнерські рішення та які аспекти можна поліпшити. Він враховує багато факторів, які впливають на взаємодію користувачів з вебзастосунком і допомагає забезпечити зручність використання та задоволення від взаємодії з ним.

Для створення mock-ups використано спеціалізований застосунок “Balsamiq Wireframes”.

Знайомство користувача з вебзастосунком завжди починається з головної сторінки (рис. 3.6), тому саме цей mock-up створено першим. Це стартова точка

для користувачів, де вони отримують загальну інформацію про видання, останні новини, популярні статті, та можуть перейти до інших розділів або статей.

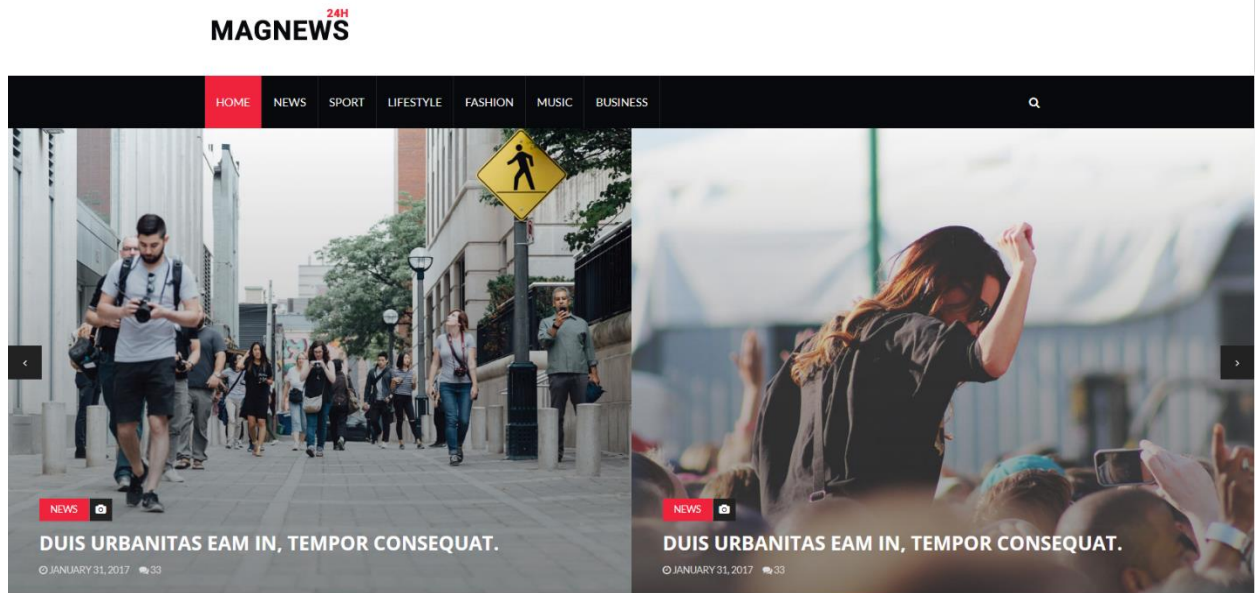


Рисунок 3.6 – Москир головної сторінки

Також потрібно створити шапку сайту (рис. 3.7). Вона також виконує функції навігації.



Рисунок 3.7 – Москир шапки сайту

Сторінки новин або статей(рис. 3.8). Ці сторінки містять конкретні новини або статті з обраних категорій. Вони мають заголовок, текстовий контент, зображення, можуть мати відео або аудіо матеріали. Користувачі можуть читати та коментувати статті, а також використовувати кнопки для поділу статті в соціальних мережах.

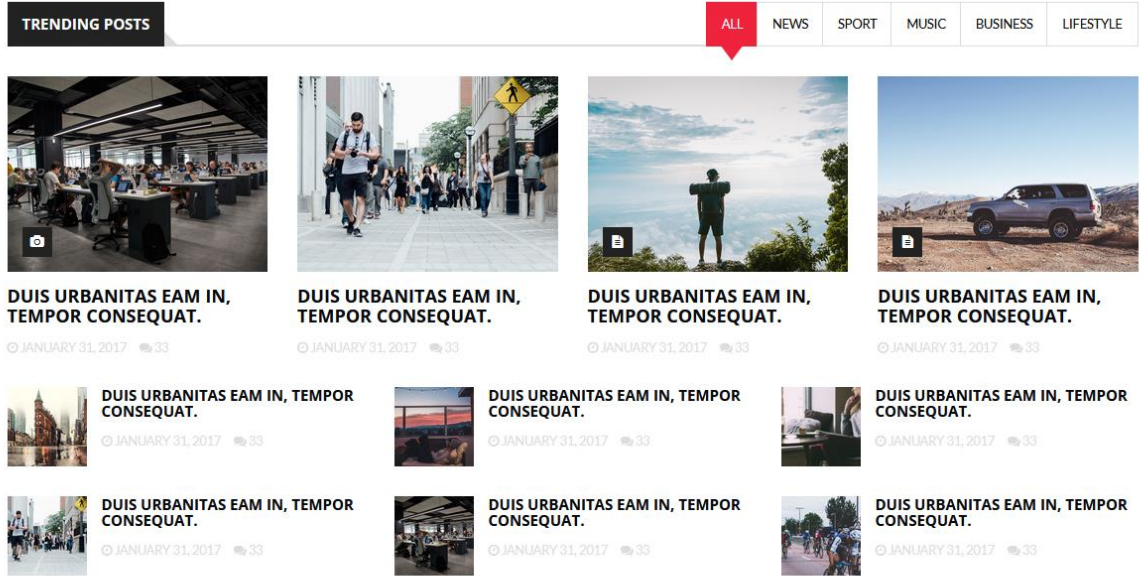


Рисунок 3.8 – Моксип сторінки новин

Форма реєстрації у вебзастосунку (рис. 3.9) є важливим елементом, який дозволяє користувачам створювати облікові записи та отримувати доступ до функціональності системи. Цей процес включає в себе збір та обробку інформації від користувачів, перевірку її достовірності та створення нового облікового запису.

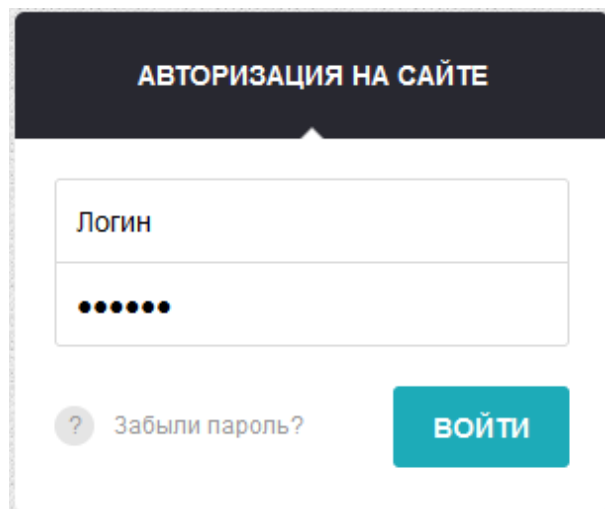


Рисунок 3.9 – Моксип авторизації

Сторінка «Популярне»(рис. 3.10) є однією з ключових сторінок вебзастосунку, яка надає користувачам можливість ознайомитися з найпопулярнішими контентом, продуктами або послугами, що пропонуються в системі. Ця сторінка зазвичай містить список елементів, які користувачі відвідують або використовують найчастіше, на підставі популярності серед інших користувачів.

Основною метою сторінки «Популярне» є надання користувачам швидкого доступу до контенту або продуктів, які є найбільш популярними або рекомендованими в системі. Це може включати найбільш переглядувані статті, відео, рейтинги або відгуки від інших користувачів.

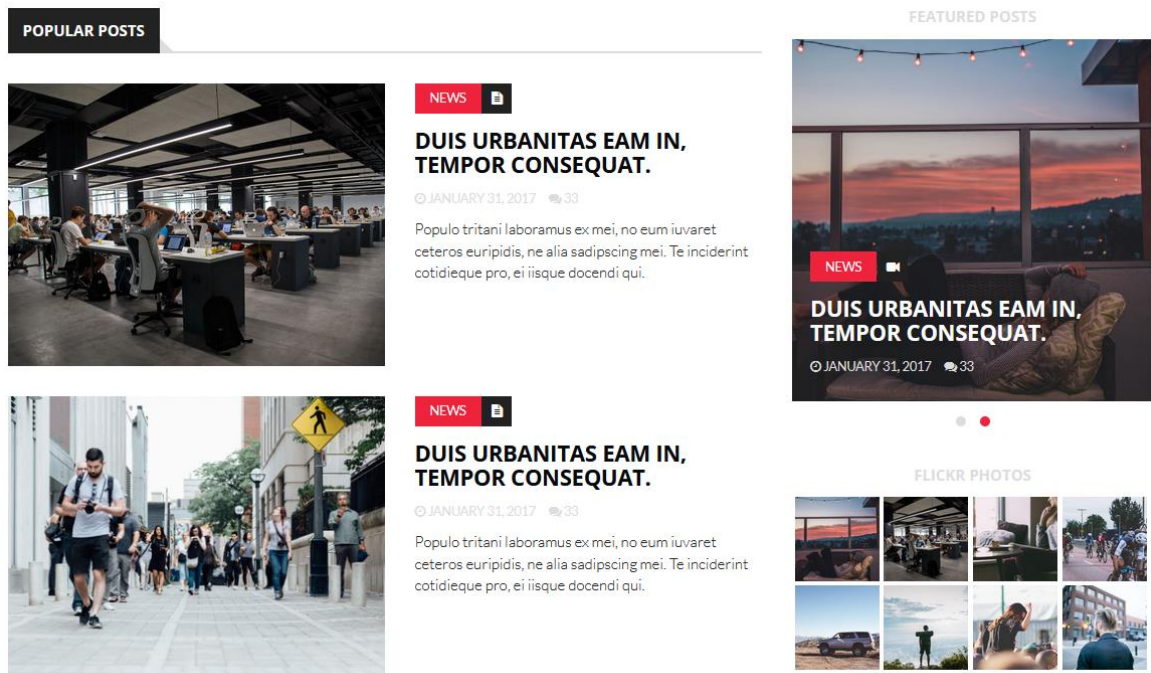


Рисунок 3.10 – Мокер сторінки Популярне

Висновки до розділу 3

У третьому розділі було розглянуто кілька підрозділів, включаючи створення діаграми прецедентів, опис варіантів використання, побудову діаграм діяльності, діаграми станів та переходів, структуру бази даних і проектування інтерфейсу користувача за допомогою мокапів. Ці кроки були важливими для

визначення архітектури системи, функціональних можливостей та способу взаємодії з користувачами.

В ході створення діаграми прецедентів були ідентифіковані основні функціональні можливості системи та їх взаємодія з акторами. Це дозволило чітко визначити, які ролі виконують актори і які дії вони можуть виконувати у системі.

Опис варіантів використання дозволив детальніше розглянути кожен прецедент і визначити послідовність дій, які відбуваються під час взаємодії акторів з системою. Це допомогло краще зрозуміти функціональний обсяг системи та можливі сценарії використання.

Побудова діаграм діяльності зосередилася на моделюванні послідовності дій та контролю потоку в процесах системи. Це було корисно для визначення порядку виконання завдань та уточнення взаємодії між об'єктами.

Діаграми станів та переходів дозволили описати різні стани системи та умови їх зміни. Це було важливо для визначення логіки системи та управління її станами.

Структура бази даних була розроблена для визначення схеми зберігання даних та взаємозв'язків між таблицями. Це дозволило створити ефективну та оптимізовану базу даних для системи.

Проектування інтерфейсу користувача за допомогою мокапів дозволило створити візуальне представлення інтерфейсу та його компонентів. Це допомогло зрозуміти, як користувачі будуть взаємодіяти з системою та які функціональні можливості будуть доступні.

Загальною метою було створення чіткої та докладної моделі системи, яка служитиме основою для подальшої реалізації. Використання різних діаграм та інструментів UML дозволило систематизувати інформацію, встановити взаємозв'язки та зрозуміти функціональні вимоги системи. Такий підхід сприяє кращому розумінню системи всіма зацікавленими сторонами і забезпечує більш точне та ефективне розроблення системи.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ

4.1 Діаграмі класів

На діаграмі класів представлені такі класи (рис. 4.1):

- клас “User”: представляє користувача системи;
- клас “System”: представляє систему інтернет-видання;
- клас “Database”: представляє базу даних, де зберігаються матеріали, коментарі та пов'язані матеріали;
- клас “Material”: представляє матеріал, який може бути відображений користувачу;
- клас “Comments”: представляє коментар, пов'язаний з матеріалом;
- клас “Associated Material”: представляє інший матеріал, який пов'язаний з вибраним матеріалом.

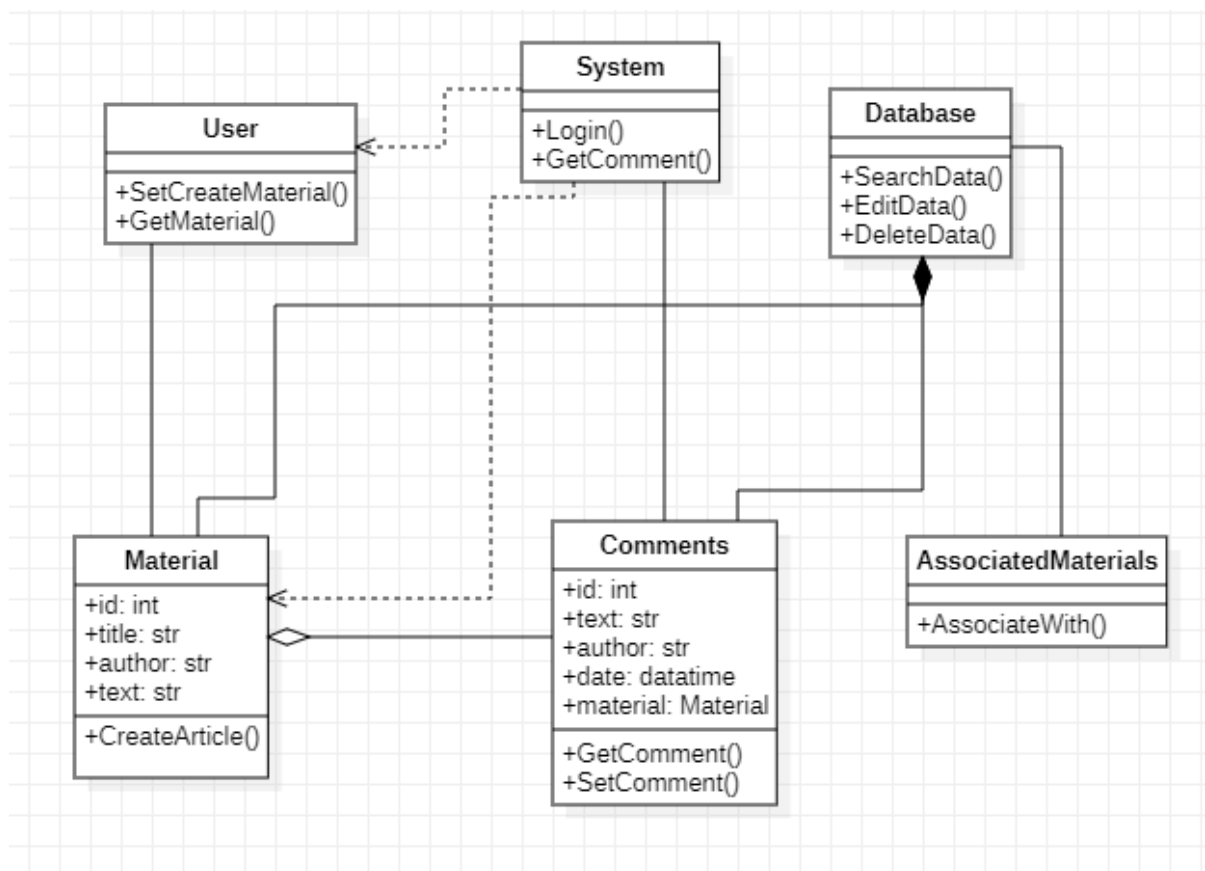


Рисунок 4.1 – Діаграма класів

На діаграмі класів використовуються різні типи відношень для показу зв'язків між класами [24]. Ось деякі додаткові зв'язки, які можуть бути представлені на діаграмі:

– Агрегація: Цей тип зв'язку вказує, що один клас є власником або складовою частиною іншого класу. Наприклад, «Матеріал» може містити кілька коментарів, тому можемо мати зв'язок агрегації між класами «Матеріал» і «Коментар».

– Композиція: Цей тип зв'язку є більш сильним варіантом агрегації, де один клас повністю визначає життєвий цикл іншого класу. Наприклад, «База даних» може містити багато матеріалів і коментарів, тому можемо мати зв'язок композиції між класами «База даних» і «Матеріал», а також між класами «База даних» і «Коментар».

– Інстанціювання: Цей тип зв'язку показує, що один клас створює екземпляр іншого класу. Наприклад, «Система інтернет-видання» може створювати екземпляри класів «Матеріал» і «Користувач», тому ми можемо мати зв'язок інстанціювання між цими класами.

4.2 Діаграма компонентів

На рис. 4.2 наведено діаграму компонентів [25] для системи «Вебзастосунок інтернет-видання».

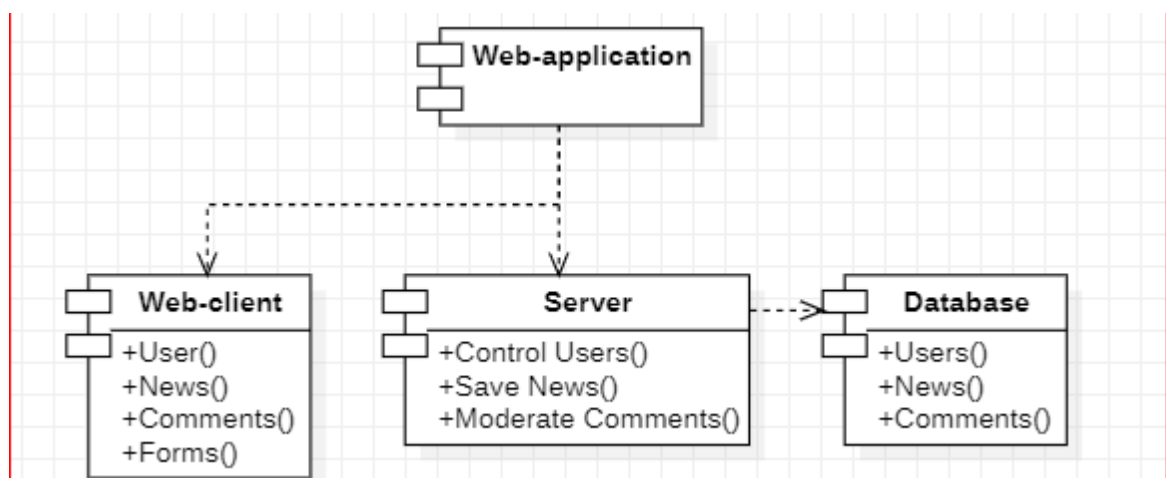


Рисунок 4.2 – Діаграма компонентів

На цій діаграмі ми маємо три компоненти: «Вебклієнт», «Сервер» та «База даних».

Класи пов'язані з цими компонентами розподілені відповідно:

1. «Вебклієнт» містить класи, пов'язані з користувачем, новинами, коментарями та формами;
2. «Сервер» містить класи, відповідальні за керування користувачами, збереження новин та обробку коментарів;
3. «База даних» містить класи, пов'язані з користувачами, новинами та коментарями, що зберігаються в базі даних.

4.3 Діаграма взаємодії

Діаграма взаємодії (Collaboration Diagram) для варіанта використання «Перегляд сторінок видань» наведена на рис. 4.3. У даному варіанті взаємодіють головний актор «Користувач» і об'єкти «Система інтернет-видання», «Сторінка видання», «Стаття», «Коментар».

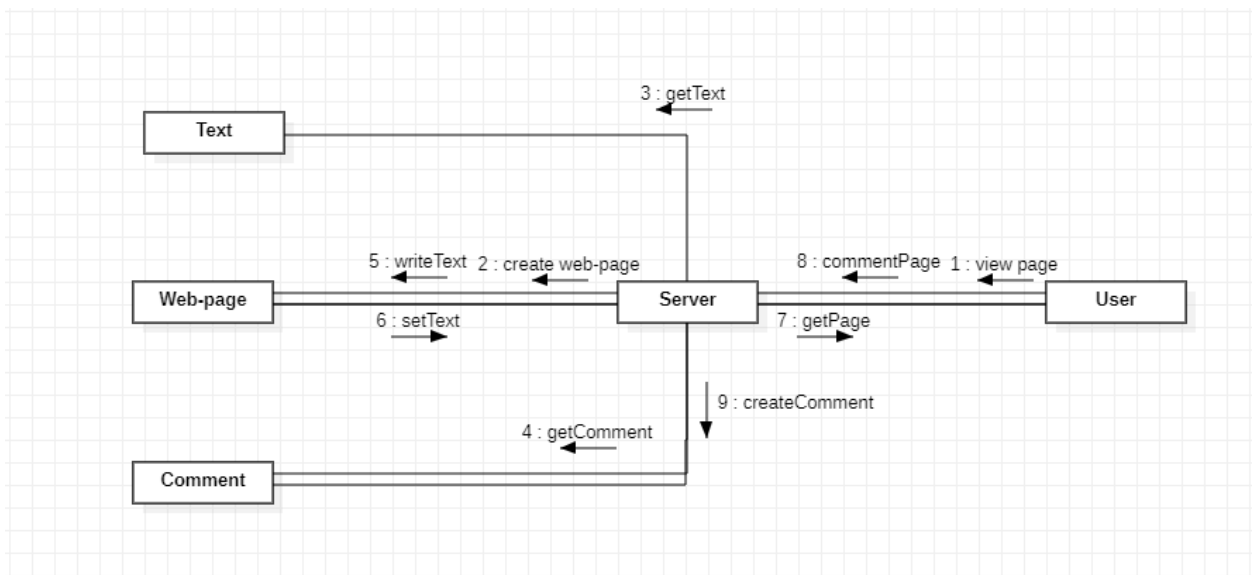


Рисунок 4.3 – Перегляд сторінок видань

Повідомлення:

1. користувач надсилає запит на перегляд сторінки видання;

2. система інтернет–видання створює об'єкт «Сторінка видання»;
3. система інтернет–видання отримує дані сторінки видання;
4. система інтернет–видання запитує коментарі до статті;
5. система інтернет–видання створює об'єкт «Стаття»;
6. система інтернет–видання отримує дані статті;
7. система інтернет–видання передає сторінку видання разом із списком статей і коментарів користувачеві. Користувач отримує сторінку видання та переглядає її;
8. користувач надсилає запит на додавання коментаря до статті;
9. система інтернет–видання створює об'єкт «Коментар».

Ця Collaboration Diagram ілюструє взаємодію між актором «Користувач» та об'єктами системи «Система інтернет–видання», «Сторінка видання», «Стаття» та «Коментар» в процесі виконання варіанту використання «Перегляд сторінок видань».

Варіант діаграми взаємодії для сценарію «Пошук матеріалів за ключовими словами».

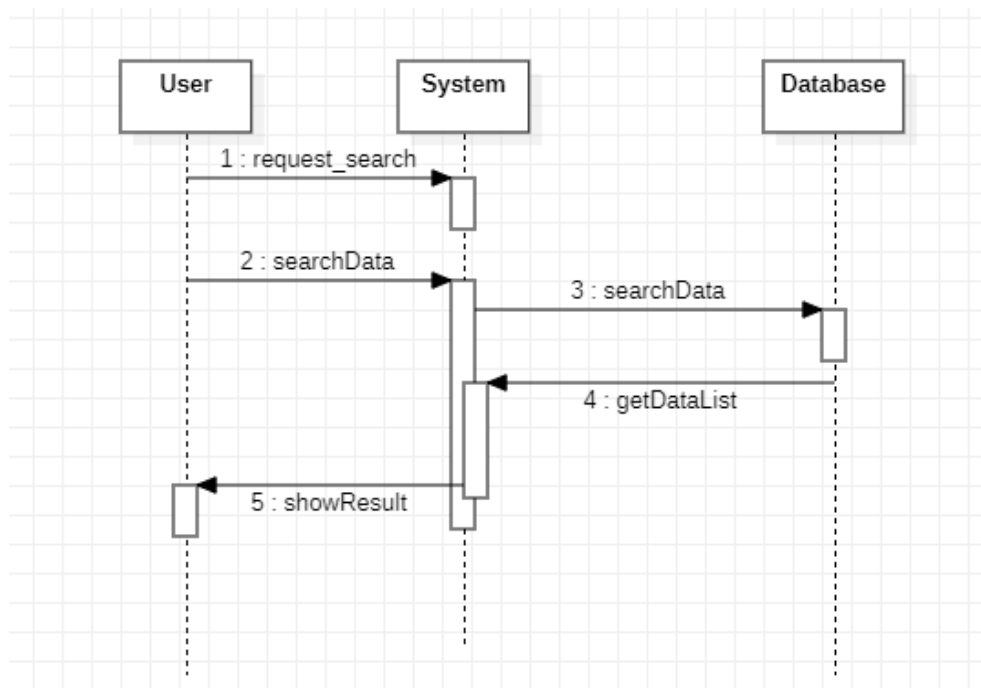


Рисунок 4.4 – Пошук матеріалів за ключовими словами

Діаграма включає наступні додаткові повідомлення:

1. користувач надсилає запит на пошук матеріалів за ключовими словами до Системи інтернет-видання.;
2. система інтернет-видання виконує пошук даних та звертається до Бази даних з запитом на пошук матеріалів за ключовими словами;
3. база даних обробляє запит і повертає Системі інтернет-видання список матеріалів, які відповідають запиту;
4. система інтернет-видання передає користувачу результати пошуку (список матеріалів).

Цей розширений варіант діаграми демонструє більш детальну послідовність повідомлень між актором «Користувач», об'єктом «Система інтернет-видання» та об'єктом «База даних» під час виконання сценарію «Пошук матеріалів за ключовими словами».

Варіант діаграми для сценарію «Перегляд матеріалів та коментування» (рис. 4.5).

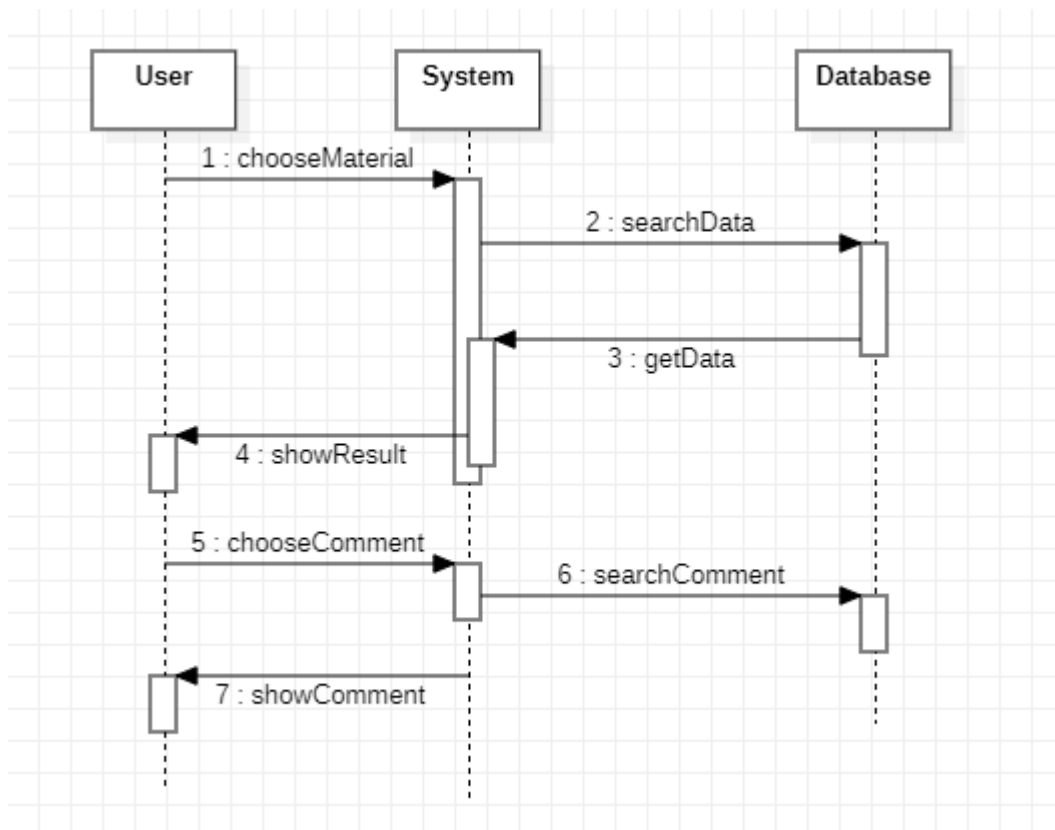


Рисунок 4.5 – Перегляд матеріалів та коментування

Включає такі повідомлення:

1. користувач вибирає матеріал для перегляду в Системі інтернет-видання;
2. система інтернет-видання звертається до Базу даних з запитом на вибір матеріалу за його ідентифікатором;
3. база даних повертає Системі інтернет-видання інформацію про вибраний матеріал;
4. система інтернет-видання відображає матеріал користувачу;
5. після відображення матеріалу, користувач запитує Базу даних про коментарі, пов'язані з цим матеріалом;
6. база даних повертає Системі інтернет-видання список коментарів для відображення;
7. система інтернет-видання відображає коментарі користувачу.

Наступна діаграма ілюструє взаємодію між акторами та об'єктами системи в процесі виконання сценарію «Пошук матеріалів за ключовими словами» (рис. 4.6).

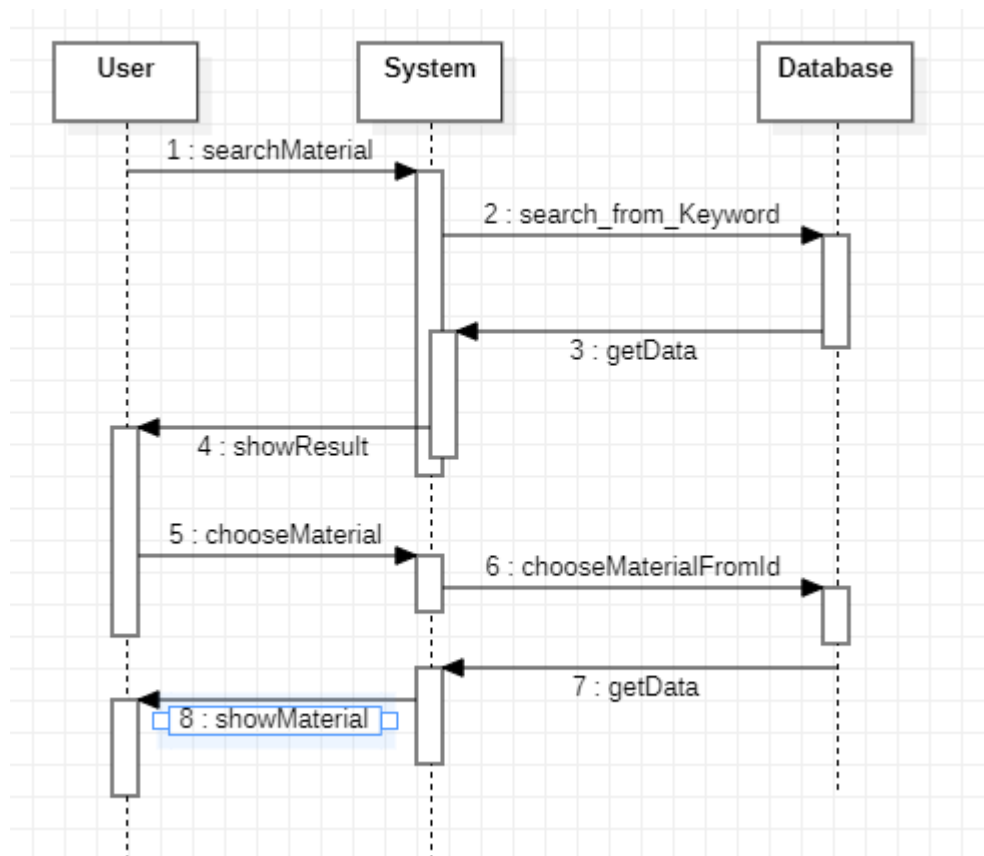


Рисунок 4.6 – Пошук матеріалів за ключовими словами

На діаграмі присутні такі елементи:

1. Актори:

– користувач: Виконує пошук матеріалів за ключовими словами.

2. Об'єкти:

– система інтернет-видання: Приймає запити користувача і здійснює пошук матеріалів за ключовими словами;

– база даних: зберігає інформацію про матеріали та їх ключові слова.

3. Повідомлення:

– користувач відправляє запит на пошук матеріалів за ключовими словами до Системи інтернет-видання;

– система інтернет-видання звертається до Базы даних і запитує інформацію про матеріали, що містять ключові слова;

– база даних повертає список матеріалів, що відповідають запиту;

– система інтернет-видання передає список матеріалів користувачу.

Ця діаграма демонструє послідовність повідомлень між актором «Користувач» та об'єктами «Система інтернет-видання» та «База даних» під час виконання сценарію «Пошук матеріалів за ключовими словами»

4.4 Інструкція користувача

Реєстрація нового користувача:

Під час реєстрації нового користувача вебзастосунку інтернет-видання, вам буде необхідно заповнити обов'язкові поля, що включають ім'я, електронну пошту та пароль. Перед початком реєстраційного процесу, переконайтеся, що ви маєте доступ до своєї електронної пошти, оскільки для завершення реєстрації буде вислано підтверджувальне повідомлення.

Кроки для реєстрації:

1. відкрийте головну сторінку вебзастосунку інтернет-видання;

2. знайдіть кнопку «Зареєструватися», яка веде до реєстраційної форми.

Натисніть на посилання або кнопку, щоб відкрити реєстраційну форму;

3. у реєстраційній формі заповніть всі необхідні поля. Це включає ім'я, електронну пошту та пароль;
4. перевірте правильність введених даних і натисніть кнопку «Зареєструватися», щоб відправити свої дані для реєстрації;
5. після успішної відправки даних, ви отримаєте підтверджувальне повідомлення на вашу електронну пошту;
6. відкрийте свою електронну пошту та знайдіть повідомлення про реєстрацію;
7. відкрийте це повідомлення і натисніть на посилання для підтвердження реєстрації;
8. після підтвердження вашого облікового запису ви будете перенаправлені на сторінку вебзастосунку, де зможете увійти за допомогою свого електронного поштового адресу та паролю.

Аутентифікація та вхід в систему:

Після успішної реєстрації у вебзастосунку, ви зможете виконати вхід до свого облікового запису за допомогою облікових даних, які ви вказали під час реєстрації. Аутентифікація дозволяє вам отримати доступ до всіх функцій та можливостей вебзастосунку, які призначені для авторизованих користувачів.

Кроки для входу в систему:

1. відкрийте головну сторінку вебзастосунку;
2. знайдіть кнопку «Увійти», яка веде до сторінки входу в систему;
3. натисніть кнопку, щоб відкрити сторінку входу в систему;
4. у формі входу в систему введіть свій електронний поштовий адрес та пароль, які ви вказали під час реєстрації;
5. перевірте правильність введених даних і натисніть кнопку «Увійти» щоб виконати вхід до системи.

Після успішного входу в систему ви будете перенаправлені на основну сторінку вебзастосунку, де зможете користуватися усіма його функціями та можливостями, які доступні авторизованим користувачам.

Відновлення пароля:

Якщо ви забули свій пароль до облікового запису, ви можете скористатися функцією відновлення пароля, щоб відновити доступ. Процес відновлення пароля передбачає відправку вам спеціального посилання.

Кроки для відновлення пароля:

1. на сторінці входу в систему вебзастосунку знайдіть посилання «Забули пароль?»;
2. натисніть на посилання, щоб відкрити сторінку відновлення пароля;
3. у формі відновлення пароля введіть свій електронний поштовий адрес, який ви використовували при реєстрації;
4. перевірте правильність введеного електронного поштового адресу і натисніть кнопку «Відновити пароль»;
5. після успішної відправки запиту на відновлення пароля, ви отримаєте електронне повідомлення з інструкціями щодо подальших дій;
6. відкрийте свою електронну пошту та знайдіть повідомлення з інструкціями щодо відновлення пароля;
7. виконайте вказані в повідомленні кроки для зміни вашого пароля;
8. після успішної зміни пароля ви зможете увійти до системи з новим паролем і отримати повний доступ до вебзастосунку.

Головна сторінка та навігація:

Після успішного входу в систему ви потрапите на головну сторінку. Головна сторінка відображає важливу інформацію, новини, статистику, а також навігаційні елементи, що дозволяють вам легко перейти до різних функціональних модулів та розділів вебзастосунку.

Навігація вебзастосунком виконується за допомогою різних елементів керування, таких як меню, підменю, кнопки та посилання. Вони дозволяють вам швидко переміщатися між різними розділами, сторінками та функціями вебзастосунку.

Наприклад, у верхній частині головної сторінки розташовується головне меню з основними розділами вебзастосунку, такими як «Головна», «Статті», «Категорії», «Користувачі» тощо. Підменю можуть містити додаткові пункти меню, що розкриваються при натисканні на них.

Завдяки зручній навігації ви зможете швидко знайти потрібну інформацію, перейти до бажаного розділу або виконати певні дії в межах вебзастосунку.

Створення та редагування статей:

Однією з основних функцій вебзастосунку є можливість створювати та редагувати статті. Статті є основною одиницею контенту вебзастосунку і представляють собою тексти, що містять інформацію, новини, огляди тощо.

Кроки для створення нової статті:

1. натисніть на пункт меню «Створити статтю», що відповідає за створення нової статті;
2. відкрийте редактор статей, де ви зможете ввести заголовок, текст статті, форматувати його, додавати зображення та інші медіафайли;
3. введіть необхідну інформацію в поля статті, такі як заголовок, категорія, ключові слова тощо;
4. збережіть статтю після завершення редагування.

Кроки для редагування існуючої статті:

1. знайдіть статтю, яку потрібно відредагувати, наприклад, на сторінці зі списком статей або за допомогою функції пошуку;
2. клацніть на статтю або відповідну кнопку, щоб відкрити редактор статті;
3. внесіть необхідні зміни до тексту, заголовка, категорії або інших атрибутів статті;
4. після завершення редагування збережіть оновлену версію статті.

Таким чином, вебзастосунок інтернет-видання надає зручність в реєстрації, входу в систему, відновленні пароля, навігації по різним розділам, створенні та редагуванні статей.

Висновки до розділу 4

У четвертому розділі було розглянуто ключові аспекти, що включали побудову діаграми класів, створення діаграми компонентів, побудову діаграм взаємодії та створення інструкції користувача.

Під час побудови діаграми класів було представлено зв'язки між різними класами, що використовуються в системі. Ця діаграма надає загальний огляд класів і їхніх взаємозв'язків, допомагаючи нам краще розуміти архітектуру нашого вебзастосунку. Це надало чітке уявлення про архітектурний склад нашого вебзастосунку та допомогло забезпечити ефективне управління даними та зберігання інформації.

Під час створення діаграми компонентів було проаналізовано компоненти, що складають наш вебзастосунок. Це дало змогу краще організувати функціональність системи шляхом відокремлення компонентів та визначення їх взаємодії, що сприяє зручності та ефективності використання системи користувачами.

Побудова діаграм взаємодії була проведена з метою детального аналізу процесів та взаємодії між різними класами та компонентами системи. Ця діаграма дає можливість глибше розуміння послідовності подій та операцій, які відбуваються в процесі взаємодії з системою.

Була розроблена детальна інструкція, яка допомагає користувачам ознайомитися з функціоналом та особливостями вебзастосунку. Ця інструкція містить послідовність кроків для реєстрації, входу в систему, навігації по різних розділах, створення та редагування статей та інші важливі дії. Вона спрощує використання вебзастосунку та забезпечує зручність користувачів у взаємодії з системою.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було розроблено вебзастосунок для інтернет-видання з метою поліпшення його функціональності та доступності для користувачів. Для досягнення цієї мети були вирішені наступні завдання:

1. Проведено аналіз предметної області та існуючих інтернет-видань для визначення їх основних функціональних та дизайнерських характеристик.

2. Визначено ключові проблеми та недоліки існуючих інтернет-видань, які вимагали вдосконалення та оптимізації.

3. Визначено необхідний функціонал для вебзастосунку інтернет-видання, зокрема, можливість перегляду статей, коментування, пошуку та сортування матеріалів, тощо.

4. Розроблено дизайн та інтерфейс користувача, забезпечуючи зручну та привабливу навігацію, читабельність текстів, графічні елементи та структуру сторінок.

5. Використано сучасні веб-технології та програмні рішення, такі як HTML, CSS, JavaScript, бази даних та фреймворки, для реалізації функціоналу вебзастосунку.

6. Проведено тестування вебзастосунку, щоб перевірити його працездатність, надійність та відповідність вимогам.

Результатом виконання кваліфікаційної роботи є розроблений вебзастосунок інтернет-видання, який забезпечує поліпшену функціональність та зручний інтерфейс для користувачів. Вебзастосунок дозволяє залучити більше читачів, забезпечує зручний доступ до інформації та сприяє взаємодії користувачів з контентом. В цілому, розробка вебзастосунку інтернет-видання виявилась ефективним способом покращення якості та доступності інформаційних ресурсів в онлайн-середовищі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Коцарев О. О. Інтернет-сайти: функціонально-змістова типологія. URL: http://www-philology.univer.kharkov.ua/katedras/prof_sites/kotsarev/kots_article_3.pdf (дата звернення: 08.04.2023).
2. Дементьева І. В. Інтернет–видання як соціально–комунікаційний простір. *Наукові праці Вінницького національного технічного університету*. 2018. № 4 (62). С. 43–49.
3. Городенко Л.М. Типологія Інтернет-видань. *Друкарство*. 2004. №5. С. 25–28.
4. Крейг Р. Інтернет-журналістика. Видавничий дім «Києво-Могилянська академія», 2007. 323 с.
5. Архітектура клієнт–сервер: вебсайт. URL: <http://inter.ptngu.com/> (дата звернення: 05.06. 2023).
6. Антонюк А. С., Пасічник Н. Р. Характеристики якісного веб-сайту. *Сучасні комп'ютерні інформаційні технології: матеріали V Всеукраїнської школисемінару молодих вчених і студентів*. Тернопіль, 22–23 травня, 2015 р. С. 81–82.
7. Хансем Г., Хансем Дж. Бази даних: розробка та управління. М: Біном, 2010. 704 с.
8. Web-розробка: вебсайт. URL: <http://fcit.tneu.org/web-rozrobka/> (дата звернення: 14.03. 2023).
9. Суслов Д. А., Винник Д. Ю., Борзенко М. І. Розробка інформаційних систем на базі вебтехнологій. Харків: Видавничий дім «Інженер», 2016.
10. Перська І. В., Гулько Є. М. Системи електронного документообігу: технології створення та впровадження. Київ: Центр учбової літератури, 2019. 448с.
11. Academy С. Р. Php: the ultimate crash course to learn PHP with practical computer coding exercises. Independently Published, 2019. 227 с.
12. The research of PHP development framework based on MVC pattern / W.Cui

and others. *Fourth international conference on computer sciences and convergence information technology*. Seoul, Korea, November 24–26, 2009. URL: <https://doi.org/10.1109/iccit.2009.130> (дата звернення: 01.06.2022).

13. Kromann F. M. *Web services. Beginning PHP and MySQL*. Berkeley, CA, 2018. P. 443–460. URL: https://doi.org/10.1007/978-1-4302-6044-8_18 (дата звернення: 01.06.2022).

14. Куренков Ю. В., Лапін В. В. Створення вебдодатків за допомогою фреймворка Ruby on Rails. *Інформаційні технології в освіті, науці і техніці*. 2019. № 4 (10). С. 31–41.

15. Meier A., Kaufmann M. *NoSQL databases. SQL & NoSQL Databases*. Wiesbaden, 2019. 229. p. URL: https://doi.org/10.1007/978-3-658-24549-8_7 (дата звернення: 01.06.2023).

16. Mavro P. *MariaDB High Performance*. Packt Publishing, 2014. 298 p.

17. Prokofyeva N., Boltunova V. Analysis and practical application of PHP frameworks in development of web information systems. *Procedia computer science*. 2017. Vol. 104. P. 51–56. URL: <https://doi.org/10.1016/j.procs.2017.01.059> (дата звернення: 01.06.2022).

18. MySQL-Строкові функції життя: вебсайт. URL: <http://www.codenet.ru/db/mysql/mystring4> (дата звернення: 12.03.2023).

19. Моделювання Adapting user interfaces with model-based reinforcement learning / K. Todi and others. *CHI '21: CHI conference on human factors in computing systems*: online virtual conference. Yokohama, Japan, Apr. 6, 2021. URL: <https://doi.org/10.1145/3411764.3445497> (дата звернення: 01.06.2023).

20. AutoCRUD: Automatic generation of CRUD specifications in interaction flow modelling language / R. Rodriguez-Echeverria and others. *Science of computer programming*. 2018. Vol. 168. P. 165–168. URL: <https://doi.org/10.1016/j.scico.2018.09.004> (дата звернення: 01.06.2023).

21. UML – діаграм: вебсайт. URL: <https://www.diagrams.net/> (дата звернення: 05.06.2023).

22. Gnesi S., Latella D., Massink M. Model checking UML Statechart diagrams using. *JACK. HASE 99: 4th IEEE International Symposium on High-Assurance Systems Engineering*. Washington, DC, USA, Nov. 17, 1999. URL: <https://doi.org/10.1109/hase.1999.809474> (дата звернення: 01.06.2023).
23. More P., Phalnikar R. Generating UML diagrams from natural language specifications. *International journal of applied information systems*. 2012. Vol. 1. №8. P. 19–23. URL: <https://doi.org/10.5120/ijais12-450222> (дата звернення: 01.06.2023).
24. Li D., Chen Y. Data migration research between relational database and nonrelational database. *Multimedia, communication and computing application*. 2015. P. 77–81. URL: <https://doi.org/10.1201/b18512-19> (дата звернення: 01.06.2023).
25. Turan M. Integrating software metrics with UML class diagrams. *Lecture notes on software engineering*. 2015. Vol. 3. № 3. P. 220–224. URL: <https://doi.org/10.7763/lmse.2015.v3.194> (дата звернення: 01.06.2023).