

ДЕРЖАВНИЙ ЗАКЛАД „ЛУГАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА”

Навчально - науковий інститут математики та інформаційних технологій

Кафедра математики та інформатики

Лановенко Валерій Юрійович

РОЗРОБКА НОВИННОГО ВЕБ-ПОРТАЛУ

кваліфікаційна робота
здобувача вищої освіти першого (бакалаврського) рівня
освітньої програми «Комп’ютерні науки та інформаційні технології»
за спеціальністю 122 „Комп’ютерні науки ”

Особистий підпис

Валерій ЛАНОВЕНКО

Науковий керівник

Юрій КОЗУБ, д.т.н., професор

В.о.завідувача кафедри

Ліна БОНДАРЕНКО, к.пед.н.

Міністерство освіти і науки України
ДЗ „Луганський національний університет імені Тараса Шевченка”
Навчально-науковий інститут математики та інформаційних технологій

Кафедра

математики та інформатики

Освітній рівень

бакалавр

Спеціальність

122 - Комп'ютерні науки

Галузь знань

12 Інформаційні технології

ЗАТВЕРДЖУЮ

В.о.зав. кафедри_

Ліна БОНДАРЕНКО

(підпис)

(ініціали, прізвище)

“ ____ ” _____ 202 ____ р.

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Лановенко Валерію Юрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи

Розробка новинного веб-порталу

Керівник кваліфікаційної роботи

Козуб Ю.Г., д.т.н., професор

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом по університету

2. Строк подання здобувачем вищої освіти

проекту

03.06.2024

3. Вихідні дані до проекту

Провести аналіз методів створення веб-порталів

(визначаються кількісні або (та) якісні показники, яким повинен відповідати об'єкт розробки)

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Постановка задачі та аналіз предметної області

Аналіз існуючих принципів побудови та засоби розробки веб-порталів

Розробка новинного порталу

(визначаються назви розділів або (та) перелік питань, які повинні увійти до тексту ПЗ)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання « 15 » лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1.	Вибір теми роботи, вивчення наукової літератури, затвердження теми та керівника.	до 1 лютого	
2.	Аналіз літературних джерел за темою роботи. Розробка та апробація методики дослідно-експериментальної роботи. Подання структури теоретичної частини роботи та плану експериментальних досліджень.	другий тиждень лютого	
3.	Робота над теоретичною частиною. Подання теоретичної частини роботи для першого читання науковим керівником.	до 1 квітня	
4.	Усунення зауважень, урахування рекомендацій наукового керівника. Подання теоретичної частини роботи на друге читання.	до 15 квітня	
5.	Проведення експериментальної роботи. Поетапний аналіз та обговорення її результатів. Перевірка стану виконання роботи.	перший тиждень квітня	
6.	Урахування рекомендацій наукового керівника, усунення недоліків, підготовка варіанта роботи до передзахисту. Розробка презентації.	до 20 травня	
7.	Попередній захист роботи на кафедрі	Травень	
8.	Доопрацювання роботи з урахуванням рекомендацій після передзахисту. Подання роботи науковому керівникові та рецензентові на підготовку відгуку та рецензії	За 10 днів до державної атестації	
9.	Подання на кафедру остаточного варіанта роботи, переплетеного та підписаного автором, науковим керівником і рецензентом.	За 5 днів до державної атестації	

Студент

підпис

В.Ю.Лановенко

(ініціали, прізвище)

Керівник роботи

підпис

Ю.Г.Козуб

(ініціали, прізвище)

I. АНОТАЦІЯ

Лановенко В.Ю. Розробка новинного веб-порталу. Кваліфікаційна робота бакалавра. 2024. 69 л.

Бакалаврська робота складається зі вступу, 3-х роздів, списку використаних джерел та додатків, містить 22 рисунок, 1 таблицю.

В аналітичному огляді літератури зі створення новинних порталів розглянуті основні підходи, технології та стандарти, які використовуються як в рамках інфраструктури порталу, так і при реалізації його функціональних модулів.

У роботі представлено функціональне призначення порталу, що полягає в наданні можливості відвідувачеві здійснювати: навігацію по новинному порталу; переглядання статей і новин, як в стоці, так і за категоріями; залишати коментарі. Так само на порталі передбачено блок для розміщення банерів (показу комерційної реклами). Визначено вимоги до розмітки, рядкових елементів новинного порталу. Побудовано функціональну структуру новинного порталу. Визначено вимоги до програмного, технічного та інформаційного забезпечення. Описано перспективність новинного порталу і можливості його розвитку.

Ключові слова: інтернет, база даних, мова програмування, вебсторінка.

II. ABSTRACT

Valery Lanovenko. Development of a news web portal. Bachelor's qualifying work. 2024. 69 pages.

The bachelor thesis consists of an introduction, 3 chapters, a list of used sources and appendices, contains 22 figures, 1 table.

In the analytical review of the literature on the creation of news portals, the main approaches, technologies and standards that are used both within the infrastructure of the portal and in the implementation of its functional modules are considered.

The work presents the functional purpose of the portal, which consists in providing the user with the opportunity to: navigate through the news portal; viewing articles and news, both in stock and by category as well as leave comments. The portal also provides a space for placing banners (displaying commercial advertisements). Requirements for markup, line elements of the news portal are defined. The functional structure of the news portal has been built. Requirements for software, technical and information support are defined. The prospects of the news portal and the possibilities of its development are described.

Keywords: Internet, database, programming language, web page, news portal.

III. ЗМІСТ

I. АНОТАЦІЯ.....	4
II. ABSTRACT.....	5
III. ЗМІСТ.....	6
IV. СПИСОК ВИКОРИСТАНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	9
V. ВСТУП	10
VI. РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	12
1. Аналітичний огляд літератури по створенню новинних порталів	12
2. Основні класифікації порталів:.....	12
1. За типом контенту	12
2. За аудиторією	13
3. За функціональністю.....	13
4. За технологічними характеристиками.....	13
5. За доступом до контенту	13
1. Клієнтський інтерфейс (Frontend)	15
6. Серверна частина (Backend).....	16
7. База даних	16
8. Система кешування.....	16
9. Система управління контентом (CMS)	16
10. Служби доставки контенту (CDN).....	16
11. Аналітика та моніторинг.....	16
12. Служби безпеки	16
VII. Основні цілі створення новинного порталу.....	17
1. Основні характеристики порталів	17
2. Перше покоління	17
3. Друге покоління.....	17

VIII.	Архітектура новинних порталів	18
1.	Клієнтський Інтерфейс (Frontend)	18
2.	API Gateway	18
3.	Мікросервіси	18
4.	Система Кешування	18
5.	База Даних	18
6.	Служби Доставки Контенту (CDN)	18
7.	Служби безпеки	18
IX.	Чого очікувати від порталів в майбутньому?	20
1.	Прогресивні веб-додатки (PWA)	20
2.	Штучний інтелект (AI) та машинне навчання (ML)	20
3.	Віртуальна та доповнена реальність (VR/AR)	20
4.	Мікросервіси	21
5.	Голосові асистенти	21
6.	Безпека та конфіденційність	21
7.	Блокчейн	21
8.	Інтернет речей (IoT)	21
9.	Хмарні обчислення	21
10.	Фреймворки для фронтенду	22
X.	Ринок засобів створення порталів	23
1.	Microsoft SharePoint	24
2.	Oracle WebCenter	26
3.	MangoApps	28
XI.	Висновки до розділу 1	30
XII.	РОЗДІЛ 2. АНАЛІЗ ПРИНЦИПІВ ПОБУДОВИ ТА ЗАСОБИ РОЗРОБКИ WEB-ПОРТАЛІВ	31
1.	Принципи побудови Web-порталів	31
2.	Засоби розробки	31

1. HTML	31
2. PHP	31
3. CSS.....	32
4. JavaScript.....	32
5. Bootstrap	32
6. MySQL.....	33
7. Web-сервер Apache.....	34
8. Codeigniter.....	35
3. Використання бібліотеки JQuery для динамічних вебсторінок	35
1. Огляд та аналіз бібліотеки jQuery, її можливості.....	35
2. Огляд існуючих плагінів під jQuery.....	43
4. JQGrid	45
5. Висновки до розділу 2	47
XIII. РОЗДІЛ 3 ПРОЕКТУВАННЯ ТА РОЗРОБКА НОВИННОГО ПОРТАЛУ NEWS RAPTOR.....	48
1. Постановка задач та функціональні вимоги до проекту	48
2. Реалізація модулів новинного порталу	51
3. Висновки по розділу 3	57
XIV. ВИСНОВКИ.....	58
XV. СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
XVI. ДОДАТКИ	62
Додаток А. Код account.php	62
1. Пояснення коду.....	65
2. Додаткові файли	65
Додаток Б. Код post.php	66
1. Пояснення коду.....	69
2. Додаткові файли	69

IV. СПИСОК ВИКОРИСТАНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

HTML - (HyperText Markup Language) Стандартизована мова розмітки документів у Всесвітній павутині.

HTML-шаблон - готовий дизайн або образ для сайту.

SQL (Structured Query Language) – стандартна непроцедурного мова, яка орієнтована на обробку даних.

PHP (Personal Home Page) - скриптова мова загального призначення, інтенсивно застосовується для розробки веб-додатків.

CSS (Cascading Style Sheets) - формальна мова опису зовнішнього виду документа, написаний з використанням мови розмітки.

JavaScript - прототипна-орієнтована сценарна мову програмування. Є реалізацією мови ECMAScript.

jQuery - бібліотека JavaScript, що фокусується на взаємодії JavaScript і HTML.

ІС - інформаційна система.

ПЗ - програмне забезпечення.

V. ВСТУП

Найбільш споживані ресурси Інтернету - це новини. Новинні портали мають великий успіх у численних відвідувачів Всесвітньої Мережі і розвиваються швидкими темпами. Оперативна і актуальна інформація, отримана з перших рук, забезпечена до того ж численними ілюстраціями подій, незмінно посідає перше місце у всіх рейтингах. Спорт і політика, економіка і бізнес, надзвичайні події та військові дії - якою б теми була новина - вона завжди знаходить свого споживача.

Слід зазначити, що в даний час відбувається активний процес інтеграції внутрішніх інформаційних систем підприємств з додатками, що забезпечують взаємодію з клієнтами і партнерами, а також з Web-додатками, призначеними для відвідувачів Web-сайтів, і число пов'язаних з цим Internet-проектів стрімко зростає. Іншими словами, багато сучасні українські і закордонні компанії зараз активно створюють інфраструктуру, що дозволяє використовувати Internet не тільки як засіб комунікації, але і як один з основних інструментів ведення бізнесу. Створення подібної інфраструктури означає впровадження Internet-додатків, що дозволяють компанії надавати своїм клієнтам і співробітникам актуальну інформацію, а часто і прямий доступ до новинних інформаційних систем та інших бізнес-програм, а також своєчасно обробляти надійшли від них відомості (наприклад, замовлення або пропозиції). Для цієї мети нерідко застосовуються портальні технології, що дозволяють здійснити інтеграцію наявних додатків в рамках єдиного способу доступу до них. В даний час спостерігається стрімке зростання ринку технологій де реалізують цих продуктів.

Високий інтерес і відвідуваність інтернет ресурсів обумовлює актуальність і виведення їх на комерційну основу. Новинний сайт це, хороша можливість отримати нових покупців і клієнтів. Тільки в Україні число користувачів інтернету налічує 20 мільйонів і ця цифра постійно зростає.

Метою написання роботи є розробка новинного порталу.

Поставлена мета визначає **завдання**, структурно повторюють зміст роботи:

- обґрунтувати доцільність розробки новинного порталу;
- провести огляд літератури по створенню новинних порталів;
- розробити основні вимоги для розроблюваного проекту;
- описати алгоритми роботи новинного порталу;
- побудувати план реалізації проекту;

Робота складається з вступу, трьох розділів, висновків, списку використаних джерел та додатків.

У вступі розглянута актуальність обраної теми дипломної роботи, визначено мету і завдання.

Перший розділ дипломної роботи включає обґрунтування актуальності створення новинного порталу, аналітичний огляд літератури за темою дослідження, представлено основні цілі його створення, визначено функціональне призначення новинного порталу, перераховано вимоги до розмітки, рядкових елементів новинного порталу, побудовано

функціональну структуру, визначено вимоги до програмного, технічного, інформаційного забезпечення. Описано перспективність проекту можливості його розвитку.

У другому розділі дипломної роботи описано принципи та засоби побудови веб-порталів. Детально досліджено найбільш використовувані плагіни бібліотеки JQuery, що реалізують динамічне верстання таблиць: DataTables та JQGrid, їх можливості та специфіку роботи з ними.

Третій розділ дипломної роботи присвячено розробці новинного порталу.

У висновках відображено висновки теоретичного та практичного характеру.

Список літератури містить перелік монографій та іншої літератури, вивченої в процесі розробки даної теми. У додатках надано лістинги коду фрагментів новинного порталу.

VI. РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1. Аналітичний огляд літератури по створенню новинних порталів

Портали є таким класом програмних систем, для якого термінологія і класифікація ще остаточно не склалися. Можна зустріти різні визначення поняття «портал» і різні за функціональними можливостями його програмні реалізації.

У літературі найбільш часто зустрічаються такі визначення:

1. портал - це інтегрований і персоніфікований web-інтерфейс для доступу користувачів до інформації, додатків і засобів співпраці;
2. портал - це інструмент для управління інтелектуальною власністю: це може бути різного роду інформація і дані. Портал збирає в собі велику кількість різної розрізненої інформації та організовує її, категоризує і персоналізує, щоб представити її у правильному вигляді, в правильний час, в правильному місці;
3. портали - це web-сайти, орієнтовані на певні аудиторії і спільноти, які забезпечують контент-агрегування - доставку інформації відповідної аудиторії; спільну роботу і сервіси підтримки спільнот (співробітників, клієнтів, акціонерів); сервіси та додатки - доступ цільової аудиторії.

У сучасному розумінні портал має, в порівнянні з Web-сайтом, розширену функціональність і реалізує ідею централізованого доступу спільноти користувачів до необхідної інформації і сервісам.

Якщо Web-сайт - це набір логічно взаємопов'язаних сторінок, доступних через Web-браузер з протоколу HTTP, то портал - це Web-сайт, який має широкий набір функцій.

Класифікувати портали можна за різноманітними критеріями. У наведеній нижче класифікації в якості критеріїв виступають тематика portalу, цільова аудиторія, вирішені порталом задачі і використовувані технології (рис.1.1).

Портали можуть класифікуватися за різними ознаками залежно від їх функціональності, призначення та типу надаваного контенту.

2. Основні класифікації порталів:

1. За типом контенту

- **Новинні портали:** зосереджуються на наданні останніх новин, статей, аналітики (наприклад, BBC, CNN).
- **Інформаційні портали:** надають широкий спектр інформації на різні теми, включаючи енциклопедії та довідники (наприклад, Wikipedia).
- **Навчальні портали:** спеціалізуються на освітніх матеріалах, курсах та тренінгах (наприклад, Coursera, Khan Academy).
- **Розважальні портали:** пропонують контент для розваг, включаючи відео, музику, ігри (наприклад, YouTube, Netflix).
- **Електронна комерція:** портали, що надають можливість здійснювати покупки онлайн (наприклад, Amazon, eBay).

2. За аудиторією

- **Корпоративні портали:** орієнтовані на співробітників, клієнтів та партнерів компанії, часто містять внутрішню інформацію, документи, новини компанії (наприклад, внутрішні портали великих компаній, такі як внутрішня мережа Microsoft або Oracle)
- **Громадські портали:** призначені для широкої публіки та надають загальнодоступну інформацію та послуги (наприклад, Портал державних послуг України)
- **Порти для спеціалізованих груп:** орієнтовані на конкретні професійні або інтересові групи (наприклад, Medscape для медичних професіоналів, Stack Overflow - для програмістів).

3. За функціональністю

- **Пошукові портали:** надають функції пошуку інформації в інтернеті (наприклад, Google, Bing).
- **Соціальні портали:** зосереджені на соціальній взаємодії, обміні інформацією та спілкуванні між користувачами (наприклад, Facebook, Twitter).
- **Порти послуг:** надають різноманітні послуги, такі як бронювання, банківські послуги, комунальні платежі (наприклад, PayPal, Booking.com).

4. За технологічними характеристиками

- **Статичні портали:** містять статичний контент, який рідко оновлюється.
- **Динамічні портали:** використовують бази даних для постійного оновлення контенту та забезпечення інтерактивності.
- **Мобільні портали:** оптимізовані для перегляду та використання на мобільних пристроях (наприклад, мобільні додатки Facebook або Amazon).

5. За доступом до контенту

- **Відкриті портали:** надають доступ до контенту для всіх користувачів без обмежень.
- **Закриті портали:** обмежують доступ до контенту, вимагаючи реєстрації або підписки (наприклад, платні новинні сайти).

Класифікація порталів допомагає зрозуміти їх різноманітність та специфіку. Кожен тип порталу має свої унікальні характеристики та призначення, що дозволяє забезпечити різноманітні потреби користувачів та бізнесів.



Рис.1.1. Класифікація порталів

Підходи, технології та стандарти, які використовуються як в рамках інфраструктури порталу, так і при реалізації його функціональних модулів, є загальновживаними. Це дозволяє описати узагальнену архітектуру порталу, що охоплює потенційно можливу функціональність (рис.1.2).

На рівні інтерфейсу користувача використовується тонкий клієнт (Web-браузер), здатний візуалізувати уявлення інформації, описана на мові HTML. Для використання деяких функціональних можливостей порталу користувач може користуватися і деякими іншими клієнтськими додатками (наприклад, клієнт електронної пошти, RSS-клієнт і тому подібне).

Узагальнена архітектура порталу включає кілька рівнів та компонентів, що забезпечують його функціональність, продуктивність та безпеку. Кожен компонент виконує свою роль, дозволяючи створювати потужні та зручні у використанні платформи для користувачів. Використання сучасних технологій та підходів допомагає забезпечити високий рівень обслуговування та задоволення користувачів.

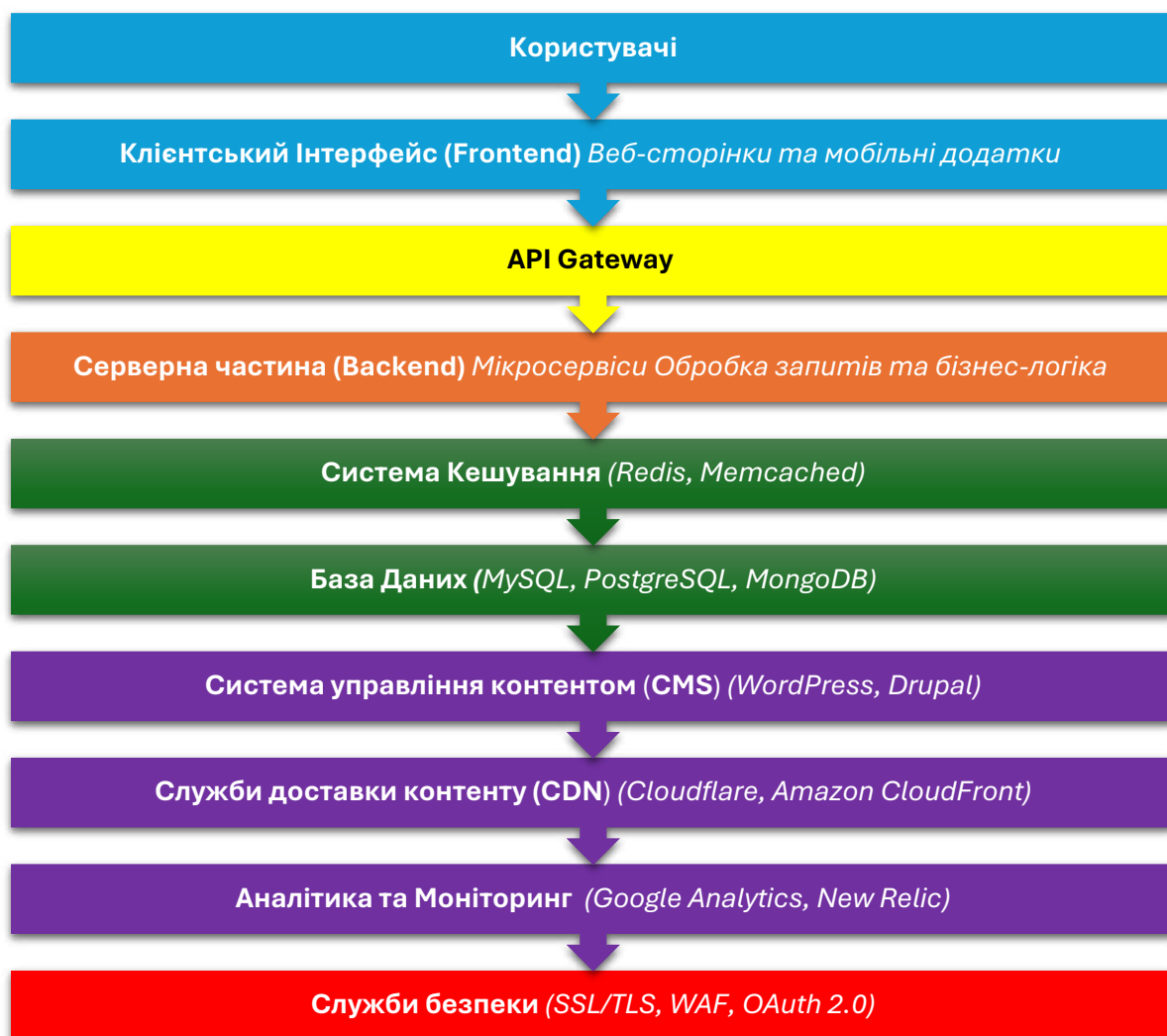


Рис.1.2. Узагальнена архітектура порталу

Для реалізації інфраструктури порталу існує ряд широко використовуваних технологій і серверів додатків. До найбільш поширених серверів додатків можна віднести такі програмні продукти як Microsoft IIS, Apache HTTP Server, Oracle WebLogic Server. Як сховищ даних використовуються реляційні бази даних реляційні (MySQL, PostgreSQL) та NoSQL бази даних (MongoDB, Cassandra) и т.п. Візуальне уявлення описується на мові HTML, який інтерпретується Web-браузером користувача.

Узагальнена архітектура порталу включає кілька рівнів та компонентів, що забезпечують ефективну роботу та інтерактивність. Ось основні компоненти такої архітектури:

1. Клієнтський інтерфейс (Frontend)

- Відповідає за взаємодію з користувачем.
- Включає веб-сторінки та мобільні додатки.
- Використовує HTML, CSS, JavaScript, а також сучасні фреймворки (React, Angular, Vue.js).

6. Серверна частина (Backend)

- Обробляє запити від клієнтів.
- Управляє бізнес-логікою та взаємодією з базою даних.
- Включає фреймворки та платформи (Node.js, Django, Ruby on Rails, Spring Boot).
- Мікросервісна архітектура для розподілу навантаження та полегшення масштабування.

7. База даних

- Зберігає дані користувачів, контенту та налаштувань порталу.
- Використовує реляційні (MySQL, PostgreSQL) та NoSQL бази даних (MongoDB, Cassandra).

8. Система кешування

- Прискорює доступ до часто використовуваних даних.
- Використовує Redis, Memcached.

9. Система управління контентом (CMS)

- Дозволяє редагувати та публікувати контент.
- Популярні CMS: WordPress, Drupal, Joomla.

10. Служби доставки контенту (CDN)

- Забезпечують швидке доставлення статичного контенту (зображення, відео).
- Використовують Cloudflare, Akamai, Amazon CloudFront.

11. Аналітика та моніторинг

- Відстежують поведінку користувачів та продуктивність системи.
- Використовують Google Analytics, Mixpanel, New Relic, Prometheus, Grafana.

12. Служби безпеки

- Захищають дані користувачів та забезпечують безперебійну роботу.
- Використовують SSL/TLS, WAF, OAuth 2.0.

Хоча підходи до розробки та реалізації порталів можна вважати досить опрацьованими з точки зору методів і використовуваних технологій, існує об'єктивна необхідність їх розвитку. Ця необхідність зумовлена розвитком телекомунікаційних технологій, які роблять інформацію більш доступною, а також об'єктивним зростанням обсягу інформації.

Портал є такою інформаційною системою (ІС), яка надає спільноті користувачів уніфікований доступ інформаційного простору, і тому проблема підвищення якості інформаційних процесів при великому обсязі інформації в порталі є особливо актуальною. Одним з підходів до вирішення даної проблеми є перехід на семантичний рівень при зборі, обробці, накопиченні, зберіганні, пошуку і поширенні інформації. Цей підхід розвивається в рамках напряму «Семантичні технології».

VII. Основні цілі створення новинного порталу

Головною метою створення порталу є - залучення більшої аудиторії на власному майданчику, для подальшого розміщення рекламних блоків, в якості отримання прибутку від діяльності.

Основні цілі розробки порталу є:

- Пошук і відбір цінної інформації.
- Зручне спілкування між відвідувачами порталу.
- Публікації своїх статей.
- Розміщення оголошень і анонсів.
- Заовлення і бронювання товарів і послуг.
- Розміщення фото, відео та іншої інформації.
- Інший, індивідуальний функціонал.

1. Основні характеристики порталів

Аналітична компанія Gartner Group в своїх дослідженнях сформулювала основні вимоги до новинних порталів, що характеризують перші два покоління цих продуктів. Згідно з цими дослідженнями, перше покоління новинних порталів має наступні характеристики[14]:

2. Перше покоління

Згідно з цими дослідженнями, перше покоління новинних порталів має наступні характеристики[14]:

- пошук і індексування широкого набору інформаційних репозиторіїв;
- категоризація інформаційного наповнення;
- управління інформаційним наповненням і його агрегація;
- персоналізація;
- високоефективна розробка додатків і можливості інтеграції з іншими додатками.

3. Друге покоління

Для другого покоління новинних порталів, вживаних як складової частини електронного бізнесу, характерні:

- надійна середовище реалізації програм;
- потужні й гнучкі інструменти розробки додатків;
- широкі можливості в області інтеграції додатків;
- відповідність вимогам до інформаційних систем масштабу підприємства;
- підтримка інтеграції з іншими додатками і інформаційними системами партнерів;
- підтримка мобільного / бездротового доступу до даних.

VIII. Архітектура новинних порталів

Архітектура новинного порталу включає різноманітні компоненти, що забезпечують його функціональність, продуктивність та безпеку. Використання сучасних технологій та підходів, таких як мікросервіси, хмарні обчислення та PWA, дозволяє створювати масштабовані та зручні у використанні платформи для надання новинного контенту користувачам. У складі типового новинного порталу умовно можна виділити 7 основних функціональних шаруючи

1. Клієнтський Інтерфейс (Frontend)

- **Фреймворки:** Використання сучасних фреймворків дозволяє створювати динамічні та інтерактивні інтерфейси.
- **Адаптивний дизайн:** Забезпечує зручний перегляд на різних пристроях (ПК, планшети, смартфони).

2. API Gateway

- Центральний вузол для обробки запитів від клієнтів до мікросервісів.
- Забезпечує маршрутизацію, балансування навантаження та безпеку запитів.

3. Мікросервіси

- Розділення функціональності порталу на незалежні сервіси, що спрощує масштабування та обслуговування.
- Кожен мікросервіс виконує певну функцію, наприклад, управління користувачами або контентом.

4. Система Кешування

- Використання кешування для зменшення навантаження на бази даних і прискорення відповіді на запити користувачів.

5. База Даних

- Реляційні та NoSQL бази даних для зберігання структурованих та неструктурованих даних.

6. Служби Доставка Контенту (CDN)

- Забезпечують швидке доставлення статичного контенту (зображення, відео) користувачам по всьому світу.

7. Служби безпеки

- Захищають дані користувачів та забезпечують безперебійну роботу

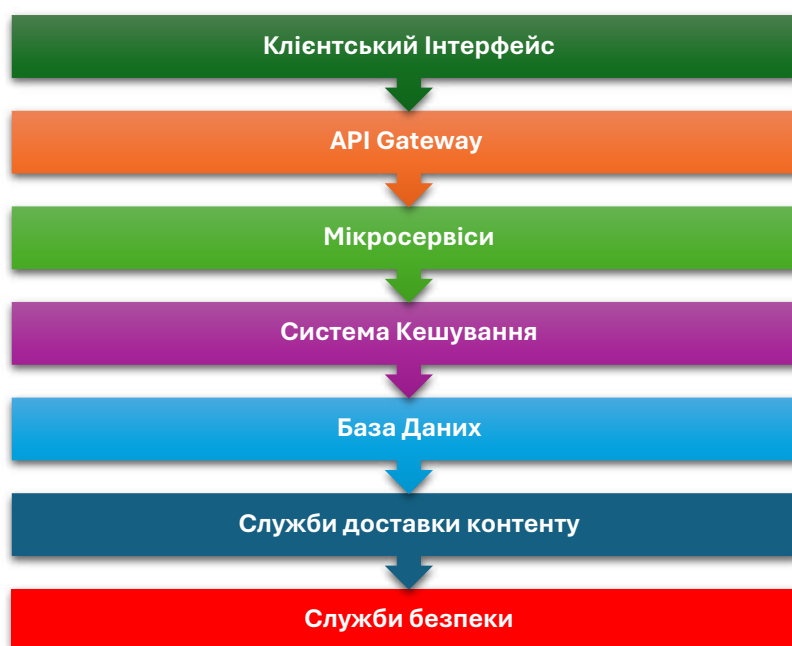


Рис. 1.3. Загальний вид функціональної архітектури порталу

Як правило, портали мають відкриту архітектуру, що дозволяє розширювати їх функціональність за рахунок додавання сторонніх додатків або додаткових компонентів. Відзначимо, що найчастіше такими компонентами виявляються засоби управління інформаційним наповненням, тому зазвичай такі засоби випускаються виробниками порталів або просто входять до складу останніх.

IX. Чого очікувати від порталів в майбутньому?

В даний час портали за своєю суттю стають засобом об'єднання бізнес-процесів усередині підприємства з бізнес-процесами партнерів і клієнтів. Реально майбутні портали зможуть виконувати функцію інтеграції додатків і даних, що необхідно для успішної взаємодії з партнерами та клієнтами. Важливу роль у функціях порталів гратиме і підтримка мобільних пристроїв, що дозволяє користувачам замінити ними традиційні настільні комп'ютери і отримати можливість постійного доступу до послуг, що ними сервісів. Відзначимо також, що однією з основних тенденцій розвитку порталів є застосування в них засобів управління інформаційним наповненням, що дозволяє реалізувати в порталі функцію його агрегації.

У плані технологій слід чекати застосування в порталах можливостей, що представляються сьогодні сучасними серверами додатків і СУБД, особливо підтримки таких технологій інтеграції додатків, як штучний інтелект та машинне навчання. Крім того, можна, мабуть, сподіватися на появу індустріальних стандартів на компоненти порталів.

У сучасних порталах використовуються різноманітні новітні технології для підвищення їх функціональності, продуктивності та безпеки. Ось кілька з них:

1. Прогресивні веб-додатки (PWA)

Прогресивні веб-додатки поєднують переваги веб-сайтів та мобільних додатків, забезпечуючи швидке завантаження, офлайн-режим та push-повідомлення.

- **Service Workers:** Технологія для забезпечення офлайн-режиму та фонові синхронізації даних.
- **App Shell:** Архітектура для швидкого завантаження інтерфейсу користувача.

2. Штучний інтелект (AI) та машинне навчання (ML)

Штучний інтелект використовується для персоналізації контенту, автоматизації обслуговування клієнтів та аналітики.

- **Чат-боти:** Автоматизовані системи для взаємодії з користувачами.
- **Рекомендаційні системи:** Алгоритми, що пропонують контент на основі попередніх взаємодій користувача.

3. Віртуальна та доповнена реальність (VR/AR)

Ці технології використовуються для створення захоплюючих і інтерактивних користувацьких досвідів.

- **Віртуальні тури:** Використовуються в туристичних та нерухомості порталах.
- **AR-фільтри:** Інтерактивні елементи для розважальних порталів.

4. Мікросервіси

Архітектурний стиль, що дозволяє розбити портал на незалежні сервіси, що покращує масштабованість та надійність.

- **Docker:** Платформа для контейнеризації мікросервісів.
- **Kubernetes:** Система для управління контейнерами на основі мікросервісів.

5. Голосові асистенти

Інтеграція голосових асистентів, таких як Google Assistant та Amazon Alexa, для взаємодії з порталом через голосові команди.

- **Розпізнавання мови:** Технології для обробки голосових команд.
- **Синтез мовлення:** Перетворення тексту на мовлення для відповіді користувачам.

6. Безпека та конфіденційність

Новітні методи та стандарти для захисту даних користувачів та забезпечення конфіденційності.

- **OAuth 2.0:** Протокол для безпечної авторизації.
- **GDPR:** Загальний регламент захисту даних для забезпечення конфіденційності користувачів у Європейському Союзі.

7. Блокчейн

Використовується для підвищення безпеки та прозорості транзакцій, особливо в фінансових порталах.

- **Смарт-контракти:** Автоматизація виконання контрактів.
- **Децентралізовані додатки (dApps):** Додатки, що працюють на блокчейн-платформах.

8. Інтернет речей (IoT)

Інтеграція з пристроями IoT для надання користувачам актуальної інформації та можливості керувати розумними пристроями.

- **Сенсори:** Збір даних для аналітики та прийняття рішень.
- **IoT-платформи:** Системи для управління підключеними пристроями.

9. Хмарні обчислення

Використання хмарних сервісів для забезпечення масштабованості, доступності та гнучкості порталів.

- **AWS (Amazon Web Services):** Популярний хмарний провайдер.

- **Google Cloud Platform:** Платформа для хмарних обчислень від Google.

10. Фреймворки для фронтенду

Сучасні фреймворки для розробки інтерфейсу користувача, що забезпечують високу продуктивність та зручність використання.

- **React:** Бібліотека для створення користувацьких інтерфейсів.
- **Vue.js:** Прогресивний фреймворк для створення інтерфейсів.
- **Angular:** Платформа для створення веб-додатків від Google.

Використання новітніх технологій дозволяє створювати потужні, зручні та безпечні портали, які відповідають сучасним вимогам користувачів та бізнесу. Вони покращують взаємодію з користувачами, підвищують ефективність обробки даних та забезпечують високу масштабованість.

Х. Ринок засобів створення порталів

За даними MarketWatch, розмір глобального ринку розробки веб-порталів був оцінений у 56 мільярдів доларів США в 2021 році і, за прогнозами, зросте на CAGR 8,03% протягом прогнозного періоду, в кінцевому підсумку досягнувши 89 мільярдів доларів США до 2027 року.

Відповідно до недавніх досліджень [G2](#), цього року на ринку новинних порталів нарешті з'явилися лідери. G2 оцінює продукти та продавців на основі відгуків, зібраних від спільноти користувачів, а також даних, агрегованих з онлайн-джерел та соціальних мереж. Разом ці бали наносяться на фірмову [G2 Grid®](#), яку можна використовувати для порівняння продуктів та швидкого визначення найкращих продуктів на основі досвіду однолітків. Оцінка G2 програмного забезпечення розраховується за допомогою двох компонентів оцінки: Задоволеність та Присутність ринку.

До лідерів відносяться Microsoft (SharePoint Portal), Oracle (Oracle WebCenter Content), MangoApps і Forest (Forest Admin). З претендентів на лідерство слід також відзначити Ingeniux і Budibase – Продукти цих компаній відносяться до найкращих пропозицій у сфері засобів управління порталами загального призначення.

Що стосується переваг потенційних споживачів засобів створення новинних порталів, явним лідером, згідно з опитуваннями різних аналітичних компаній, є Microsoft (рис. 1.4).



Рис. 1.4. Аналітика споживачів засобів створення порталів

Нижче ми розглянемо засоби створення порталів провідних виробників: Microsoft, Oracle, MangoApps, Forest, Ingeniux і Budibase.

1. Microsoft SharePoint

Microsoft SharePoint — це платформа для співпраці та управління документами, яка дозволяє організаціям ефективно зберігати, організовувати, ділитися та отримувати доступ до інформації. SharePoint забезпечує широкий спектр можливостей для інтеграції з іншими продуктами Microsoft, автоматизації бізнес-процесів та розгортання веб-додатків.

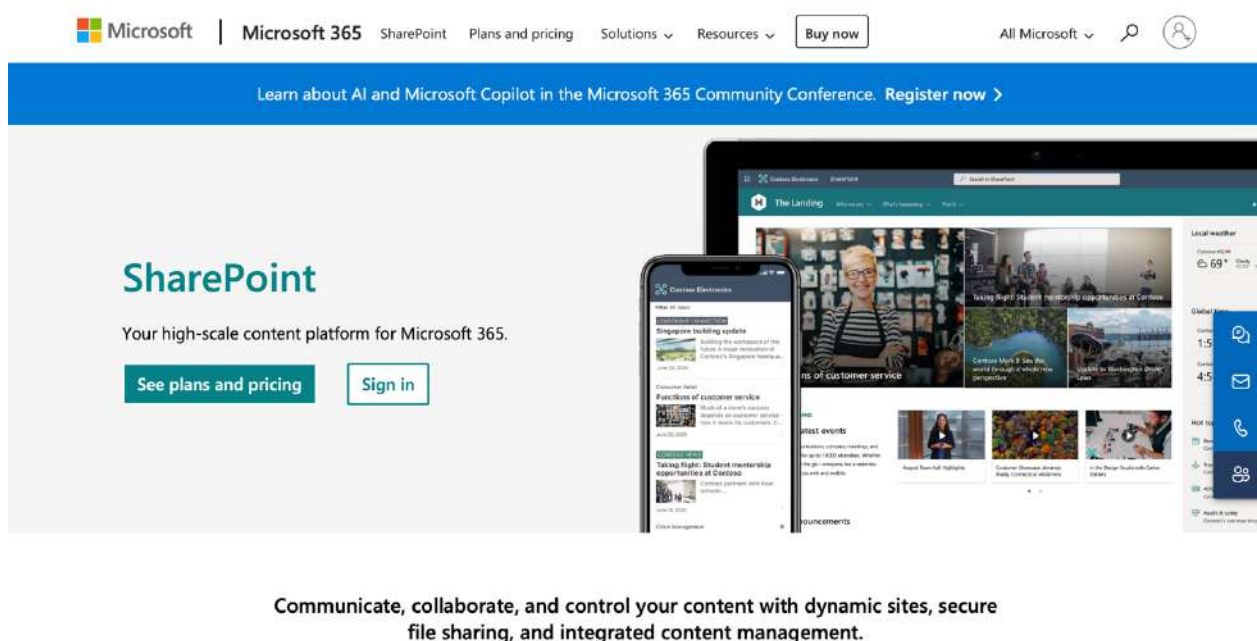


Рис. 1.5. Головна сторінка розробника Microsoft SharePoint

Основні компоненти та функціональність

1. Бібліотеки документів

- **Опис:** Централізоване сховище для зберігання та керування документами.
- **Функції:** Підтримка версійності, керування дозволами, спільний доступ, інтеграція з Microsoft Office.
- **Технічні деталі:** Підтримка метаданих для організації контенту, використання Azure Blob Storage для зберігання великих обсягів даних.

2. Списки SharePoint

- **Опис:** Структуровані колекції даних, які можуть містити текст, числа, дати та інші типи даних.
- **Функції:** Створення списків завдань, контактів, календарів тощо.
- **Технічні деталі:** Використання списків для створення динамічних форм і робочих процесів, підтримка REST API для інтеграції з іншими системами.

3. Сайти групи та сайти комунікації

- **Опис:** Веб-сайти для спільної роботи команд або комунікації з великою аудиторією.
- **Функції:** Публікація новин, ведення блогів, обговорення, обмін документами.
- **Технічні деталі:** Підтримка сучасних шаблонів, використання веб-частин для налаштування сторінок.

4. Пошук

- **Опис:** Потужні інструменти для пошуку контенту на сайтах SharePoint.
- **Функції:** Пошук по документах, списках, сайтах, можливість налаштування пошукових індексів.
- **Технічні деталі:** Використання Microsoft Search для інтеграції з іншими продуктами Microsoft, підтримка запитів на природній мові.

5. Управління контентом (ЕСМ)

- **Опис:** Інструменти для управління життєвим циклом документів та записів.
- **Функції:** Класифікація контенту, зберігання записів, архівування.
- **Технічні деталі:** Використання політик зберігання та утилізації, інтеграція з Microsoft Information Protection для безпеки даних.

6. Автоматизація бізнес-процесів

- **Опис:** Створення та управління робочими процесами для автоматизації завдань.
- **Функції:** Підтримка Microsoft Power Automate для створення робочих процесів без написання коду.
- **Технічні деталі:** Використання шаблонів робочих процесів, інтеграція з Power Apps для створення користувацьких додатків.

7. Безпека та відповідність

- **Опис:** Інструменти для захисту даних та забезпечення відповідності нормативним вимогам.
- **Функції:** Керування дозволами, аудиторські журнали, захист від втрати даних (DLP).
- **Технічні деталі:** Інтеграція з Azure Active Directory для аутентифікації та авторизації, підтримка політик безпеки на основі ролей.

Переваги та можливості інтеграції

- **Інтеграція з Microsoft 365:** Плавна інтеграція з іншими продуктами Microsoft, такими як Teams, OneDrive, Outlook, що забезпечує єдиний користувацький досвід.
- **Розширюваність:** Можливість розширення функціональності за допомогою SharePoint Framework (SPFx), який дозволяє створювати користувацькі веб-частини та розширення.
- **Хмарні та локальні рішення:** SharePoint може бути розгорнутий як у хмарі (SharePoint Online), так і на локальних серверах (SharePoint Server), що забезпечує гнучкість у виборі моделі розгортання.

Приклади використання

1. **Управління проектами:** Використання сайтів групи для управління проектами, зберігання документів, відстеження завдань та строків.

2. **Внутрішні портали:** Створення внутрішніх порталів для корпоративних новин, оголошень, доступу до політик та процедур.
3. **Спільна робота над документами:** Спільне редагування документів в реальному часі, зберігання різних версій, забезпечення контролю доступу.

Додаткові ресурси

- **Офіційна документація:** [Microsoft SharePoint Documentation](#)
- **Блоги та вебінари:** [Microsoft Tech Community](#)

Microsoft SharePoint забезпечує потужні інструменти для співпраці та управління контентом, що робить його важливим компонентом для організацій, які прагнуть оптимізувати свої бізнес-процеси та підвищити ефективність роботи команд.

2. Oracle WebCenter

Oracle WebCenter — це комплексна платформа для управління корпоративним контентом та створення рішень для взаємодії з користувачами. Вона забезпечує інтеграцію бізнес-процесів, соціальну співпрацю, управління документами та створення порталів, що робить її важливим інструментом для цифрової трансформації організацій.

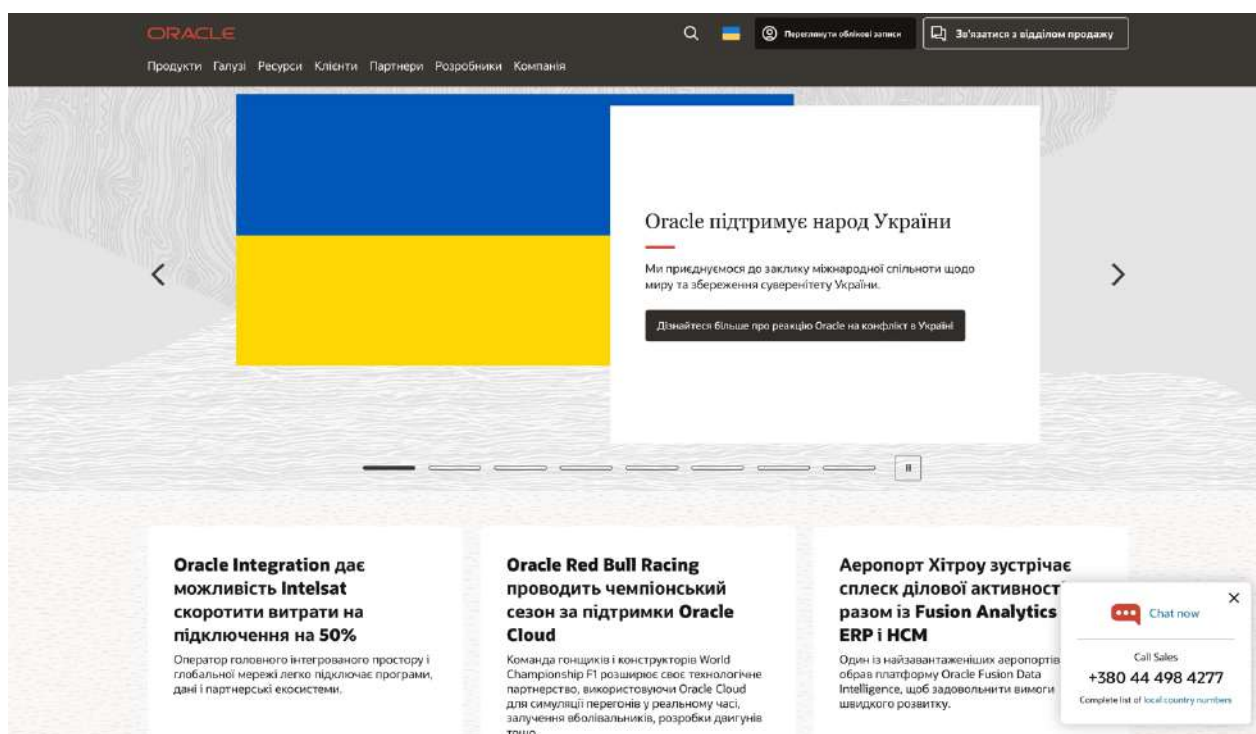


Рис. 1.6. Головна сторінка розробника Oracle WebCenter

Основні компоненти та функціональність

1. Oracle WebCenter Content

- **Опис:** Забезпечує управління корпоративним контентом (ECM), включаючи документи, зображення та відео.

- **Функції:** Управління життєвим циклом контенту, інтеграція з іншими системами, пошук та індексація контенту.
 - **Технічні деталі:** Підтримка метаданих, засоби контролю версій, керування дозволами та політики зберігання.
2. **Oracle WebCenter Portal**
- **Опис:** Платформа для створення інтерактивних порталів, що забезпечують доступ до різноманітних бізнес-додатків та інформації.
 - **Функції:** Підтримка веб-частин, налаштовувані шаблони, інтеграція з соціальними медіа.
 - **Технічні деталі:** Використання Java EE для розробки, підтримка REST API для інтеграції, масштабованість завдяки Oracle Cloud.
3. **Oracle WebCenter Sites**
- **Опис:** Система для управління веб-контентом (WCM), яка дозволяє створювати, редагувати та публікувати контент на веб-сайтах.
 - **Функції:** Інструменти для управління цифровим досвідом, персоналізація контенту, мультиканальне публікування.
 - **Технічні деталі:** Підтримка багатомовності, інтеграція з аналітичними інструментами, можливість роботи з великими обсягами даних.
4. **Oracle WebCenter Social**
- **Опис:** Інструменти для соціальної взаємодії та співпраці в рамках організації.
 - **Функції:** Форум, блоги, обмін повідомленнями, інтеграція з іншими соціальними платформами.
 - **Технічні деталі:** Використання соціальних графів для аналізу взаємодій, підтримка мобільних пристроїв.

Переваги та можливості інтеграції

- **Інтеграція з ERP та CRM:** Можливість інтеграції з Oracle ERP та CRM для забезпечення безшовного доступу до даних та процесів.
- **Підтримка хмарних і локальних рішень:** Гнучкість у виборі моделі розгортання, включаючи Oracle Cloud.
- **Безпека:** Підтримка багаторівневого захисту даних, включаючи шифрування та контроль доступу на основі ролей.

Приклади використання

1. **Корпоративні портали:** Створення порталів для внутрішнього використання, що забезпечують доступ до інформації та інструментів для співпраці.
2. **Управління контентом:** Централізоване управління документацією та цифровими активами організації.
3. **Соціальна співпраця:** Використання соціальних інструментів для підвищення залученості співробітників та покращення комунікації.

Додаткові ресурси

- [Oracle WebCenter Overview](#)
- [Oracle WebCenter Documentation](#)
- [Oracle Cloud](#)

Oracle WebCenter надає потужні інструменти для управління контентом та співпраці, що дозволяє організаціям ефективно реалізовувати свої бізнес-процеси та забезпечувати високий рівень залученості користувачів.

3. MangoApps

MangoApps — це інтегрована платформа для цифрових робочих місць, яка об'єднує можливості корпоративного порталу, управління контентом, соціальної співпраці та багато інших функцій, що дозволяють підвищити ефективність роботи та залученість співробітників.

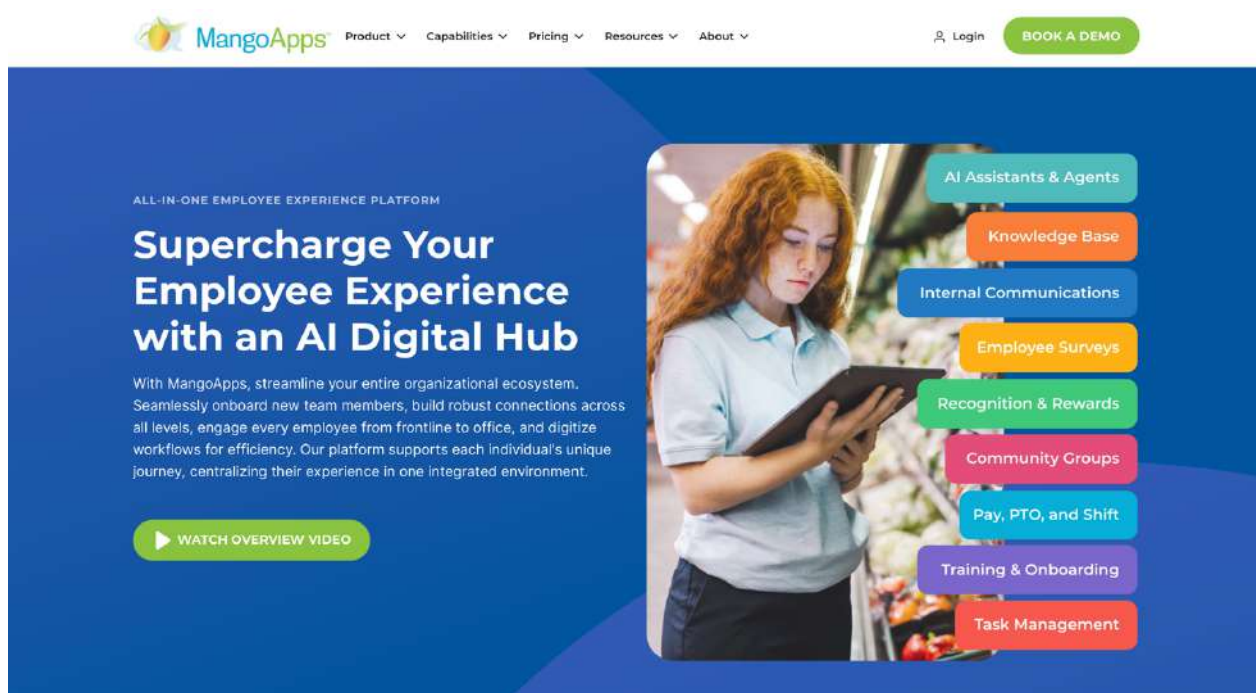


Рис. 1.7. Головна сторінка розробника MangoApps

Основні компоненти та функціональність

1. Корпоративний портал

- **Опис:** Платформа для створення інтерактивних корпоративних порталів.
- **Функції:** Швидке створення та організація сайтів, документів та ресурсів без необхідності залучення ІТ-фахівців. Можливість інтеграції з іншими системами та сервісами.
- **Технічні деталі:** Використання понад 75 віджетів для гнучкого дизайну сторінок, автоматичне створення меню та підменю для полегшення навігації.

2. Управління контентом

- **Опис:** Засоби для управління цифровими активами та контентом організації.
- **Функції:** Створення та публікація контенту, управління життєвим циклом контенту, підтримка багатомовності.
- **Технічні деталі:** Підтримка метаданих та пошуку за контентом всередині документів, інтеграція з AI для покращення пошукових можливостей.

3. Соціальна співпраця

- **Опис:** Інструменти для підвищення співпраці та взаємодії між співробітниками.
- **Функції:** Чати, приватні повідомлення, форуми, блоги, групи за інтересами.
- **Технічні деталі:** Інтеграція з зовнішніми соціальними мережами та іншими корпоративними комунікаційними платформами.

4. Управління проектами

- **Опис:** Інструменти для управління проектами та завданнями.
- **Функції:** Створення робочих просторів для команд, управління завданнями, відстеження прогресу проектів.
- **Технічні деталі:** Використання інтерактивних форм та таблиць для збору інформації та відстеження активностей, автоматизація процесів за допомогою шаблонів робочих процесів.

5. Інтеграції

- **Опис:** Широкі можливості для інтеграції з іншими корпоративними системами.
- **Функції:** Підтримка інтеграції з популярними системами SSO, такими як SAML, OAuth2, AD, LDAP, Office 365 та Google Workspace.
- **Технічні деталі:** Використання OpenAPI для створення кастомних інтеграцій, підтримка понад 200 готових інтеграцій з різними системами.

6. Безпека та відповідність

- **Опис:** Засоби для забезпечення безпеки даних та відповідності нормативним вимогам.
- **Функції:** Захист даних, контроль доступу, аудиторські журнали, шифрування даних.
- **Технічні деталі:** Використання хмарних технологій AWS для забезпечення безпеки, сертифікація HITRUST, SOC 2 Type II та ISO 27001.

7. Аналітика та моніторинг

- **Опис:** Інструменти для моніторингу активності користувачів та продуктивності системи.
- **Функції:** Звіти та аналітика в реальному часі, відстеження взаємодій та активностей.
- **Технічні деталі:** Використання аналітичних панелей для відображення ключових показників, підтримка інтерактивних дашбордів.

Додаткові ресурси

- **Офіційний сайт MangoApps:** [MangoApps](#)
- **Документація MangoApps:** [MangoApps Documentation](#)
- **Інформація про безпеку:** Advanced Security and Compliance

MangoApps надає комплексні інструменти для створення інтерактивних робочих просторів, що дозволяє організаціям ефективно реалізовувати свої бізнес-процеси та забезпечувати високий рівень залученості співробітників.

XI. Висновки до розділу 1

У першому розділі бакалаврської роботи проведено обґрунтування функціоналу новинного порталу, розглянуто класифікацію й узагальнену архітектуру порталу, визначено ринок засобів створення порталів та охарактеризовано можливості інтеграційних продуктів компаній – розробників з бізнес -додатками. Дослідження роботи показує перспективність та актуальність розробляемого проекту.

ХІІ. РОЗДІЛ 2. АНАЛІЗ ПРИНЦИПІВ ПОБУДОВИ ТА ЗАСОБИ РОЗРОБКИ WEB-ПОРТАЛІВ

1. Принципи побудови Web-порталів

Для того щоб Web-портал перетворився в дійсно результативний і надійний маркетинговий інструмент, необхідно привнести в нього всього кілька невеликих, але суттєвих змін.

Кожен кваліфікований Web-розробник повинен пам'ятати про той факт, що в століття інтерактивних технологій на будь-якого користувача мережі Інтернет, що є потенційним споживачем, обрушується маса всілякої інформації, зокрема, повідомлень, що носять рекламний характер. Тому в області маркетингу існує жорстка конкуренція, витримати яку, залучаючи потенційного споживача, можна лише, шляхом створення дійсно якісного Web-сайту. Отже, Web-сайт повинен мати не тільки привабливий, яскравий дизайн, а й містити корисну і інтригуючу потенційного споживача інформацію. Зміст головної сторінки сайту має не тільки стимулювати відвідувача перейти зі стартової сторінки на решту, а в подальшому і зробити покупку, але і відвідувати ресурс згодом, рекомендуючи його своїм близьким.

2. Засоби розробки

1. HTML

Мова розмітки гіпертексту (Hypertext Markup Language), або, як його частіше називають, HTML, - це комп'ютерну мову, що лежить в основі World Wide Web (Всесвітньої Павутини). Завдяки мові HTML будь-який текст можна розмітити, перетворивши його в гіпертекст з подальшою публікацією в Web [1]

Мова HTML має власний набір символів, за допомогою яких Веббраузери відображають сторінку. Ці символи, звані дескрипторами, включають в себе елементи, необхідні для створення гіперпосилань [3, с. 20].

Однією з відмінних рис HTML -документів є те, що сам документ містить лише текст, а всі інші об'єкти вбудовуються в документ в момент його відображення Браузером за допомогою спеціальних тегів і зберігаються окремо. При збереженні HTML-файлу в місці розміщення документа створюється папка, в яку поміщаються супутні йому графічні елементи оформлення [2, с. 14].

2. PHP

В першу чергу PHP використовується для створення скриптів, що працюють на стороні серверу, для цього його, власне, і придумали. PHP здатний вирішувати ті ж завдання, що і будь-які інші CGI -скрипти, в тому числі обробляти дані html-форму, динамічно генерувати html сторінки тощо. Але є й інші області, де може використовуватися PHP.

Друга область - це створення скриптів, що виконуються в командному рядку. Тобто за допомогою PHP можна створювати такі скрипти, які будуть виконуватися, поза за залежності від web-сервера і браузера, на конкретній машині.

І остання область - це створення GUI-додатків (графічних інтерфейсів), що виконуються на стороні клієнта [3, с. 58].

3. CSS

Під терміном CSS ми розуміємо фразу "Каскадні таблиці стилів". Вони впроваджуються в HTML-код і не вимагають ніяких спеціальних редакторів і компіляторів, адже інтерпретуються звичайним браузером.

Каскадні таблиці стилів є описами різних HTML-елементів і створені вони для розширення властивостей останніх. Вперше стилі були запропоновані WWW Consortium'ом в рамках розробки специфікації HTML 3.0, проте реально цю можливість стали підтримувати тільки в 1997 році. Насолодитися CSS мали можливість лише щасливі власники таких браузерів, як Netscape Navigator 4.0 і Internet Explorer 4.0.

CSS відкриває нам нові, раніше невідомі і недоступні межі. З їх допомогою ми отримуємо можливість більш вишукано оформити свій Вебсайт, надати йому нових рис і симпатичне обличчя. Синтаксис CSS досить простий у вивченні, тому освоєння каскадних стилів є справою досить легким, але дуже і дуже корисним. Наприклад. Візьмемо типову таблицю HTML і поставимо атрибут border = "1". Після інтерпретації даного коду браузером ми отримаємо самий звичайний результат, до якого звик наш досвідчений очей. Інша справа, що таблиці подібного роду не сумісні ні з одним мало-мальськи хорошим дизайном. Ну куди годиться ця псевдо-тривимірنا лінія, що представляє собою границю. Ось тут і приходять на допомогу CSS, за засобами яких можна оформити будь-яку таблицю на свій смак.

4. JavaScript

JavaScript можна назвати одним з найбільш використовуваних прототипно-орієнтованих, сценарних мов програмування. Даний код може виконуватися як на стороні клієнта, так і на стороні серверу. Практично не один із сучасних сайтів не обходиться без використання цієї мови програмування. JavaScript була розроблена компанією Netscape для додання інтерактивності веб-сторінок.

Найбільш популярні області застосування JavaScript:

- динамічне створення документа за допомогою сценарію;
- попередня перевірка введених користувачем даних до передачі їх на сервер;
- створення динамічних HTML-сторінок;
- взаємодія з користувачем при вирішенні "локальних" завдань.

5. Bootstrap

Bootstrap також відомий як Bootstrap (Twitter Bootstrap) - вільний набір інструментів для створення сайтів і веб-додатків. Включає в себе HTML і CSS шаблони оформлення для типографіки, веб-форму, кнопок, міток, блоків навігації та інших компонентів веб-інтерфейс, включаючи JavaScript-розширення.

Bootstrap використовує найсучасніші напрацювання в області CSS і HTML, тому часто необхідно бути уважним при підтримці старих браузерів які не підтримують деякі функції.

Переваги Bootstrap

Основною перевагою Bootstrap є економія часу при розробці використовуючи шаблони і класи дизайну.

Динамічні макети Bootstrap масштабуються на різні пристрої і дозволу екрану без яких-небудь змін в розмітці, що скорочує час оптимізації проекту під різні розрощення.

Загальний гармонійний дизайн всіх компонентів платформи Bootstrap.

Використовуються єдиний стиль і шаблони за допомогою центральної бібліотеки. Дизайн і макети веб-сторінок узгоджуються один з одним.

Простота у використанні платформи дозволяє, користувачеві з базовими знаннями HTML і CSS розробляти шаблони сайтів професійно і в найкоротші терміни.

Сумісність з браузерами в Twitter Bootstrap також є незаперечною перевагою. Bootstrap коректно відображається на останніх версіях веббраузерів: Mozilla Firefox, Google Chrome, Safari, Internet Explorer, Microsoft Edge і Opera.

Особливістю Bootstrap є відкритий вихідний код і безкоштовне розповсюдження.

Основні інструменти Bootstrap:

Сітки - заздалегідь задані розміри колонок, які можна відразу ж використовувати, наприклад ширина колонки 140px відноситься до класу `.span2` (`.col-md-2` в третій версії фреймворку).

Шаблони - Фіксований або гумовий шаблон документа. Типографіка -

Описи шрифтів, визначення деяких класів для шрифтів, таких як код, цитати і т. п.

Медіа - Представляє деякий спосіб упорядкування зображень та Відео. Таблиці - Засоби оформлення таблиць, аж до додавання функціональності сортування.

Форми - Класи для оформлення форм і деяких подій, що відбуваються з ними.

Навігація - Класи оформлення для табів, Вкладок, Сторінковий, Меню і панель інструментів.

Алерти - Оформлення діалогових вікон, Підказок і Спливаючих вікон.

6. MySQL

MySQL - це найпопулярніша СУБД з відкритим вихідним кодом. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. Насправді MySQL становить дедалі більшу конкуренцію таким дорогим гігантам, як IBM і Microsoft.

В січні-лютому 2008 Sun Microsystems придбала розробника системи керування базами даних MySQL. Після поглинання у 2009 році Sun Microsystems компанією Oracle Corporation MySQL стала власністю Oracle. Як не дивно, але не знайдете пояснення того, що означає абревіатура MySQL. Що стосується останньої частини, «SQL», тут все ясно: це Structured Query Language (структурована мова запитів) - мова, застосовуваний для взаємодії з більшістю існуючих баз даних. Але ось префікс «Му» ніяк офіційно не розшифровується навіть в самій компанії [3, с. 10]

MySQL - це реляційна система управління базами даних (РСУБД). З технічної точки зору MySQL - програма, що управляє файлами, які складають базу даних, але часто термін «база даних» (БД) застосовується і до адмін програмі, і до цього набору файлів. БД - це просто сукупність взаємопов'язаних даних (текстових, числових, довічних), за зберігання і організацію яких відповідає СУБД.

Є багато видів баз даних: від найпростіших, де дані зберігаються в плоских файлах, до реляційних і об'єктно-орієнтованих. Реляційна база даних містить інформацію про кількох таблицях. Ця концепція була розроблена на початку сімдесятих місто, а до того бази даних

нагадували одну величезну електронну таблицю, де зберігається буквально все. Для проектування реляційної БД потрібно більше зусиль, але це окупається підвищеною надійністю роботи і цілісністю даних.

MySQL - це програма з відкритим вихідним кодом, так само як і PHP і деякі версії системи UNIX & . Це означає, що ви можете безкоштовно встановлювати, запускати програму і модифікувати її вихідний код (Який, як і йому самому, можна запустити з Мережі). Але в деяких випадках все ж таки повинні заплатити за ліцензію на MySQL і, зокрема, тоді, коли отримуєте прибуток від включення MySQL до складу свого продукту. Детальніше про це можна дізнатися в умовах ліцензування MySQL [3, с. 11].

MySQL складається з декількох частин, в тому числі сервера MySQL (Програми MySQL, яка, власне, і керує базою даних), клієнта MySQL (програми MySQL, що надає інтерфейс до сервера) і численних службових утиліт для обслуговування бази даних і інших цілей. Роботу з MySQL можна вести, користуючись багатьма поширеними мовами програмування, включаючи PHP, Perl і Java.

7. Web-сервер Apache

Apache є кросплатформним програмним забезпеченням, підтримуючи такими ОС як Linux, BSD, Mac OS, Microsoft Windows. Основними достоїнствами Apache вважаються надійність і гнучкість конфігурації. Він дозволяє підключати зовнішні модулі для надання даних, використовувати СУБД для аутентифікації користувачів, модифікувати повідомлення про помилки.

Ядро Apache включає в себе основні функціональні можливості, такі як обробка конфігураційних файлів, протокол HTTP і система завантаження модулів. Ядро (на відміну від модулів) повністю розробляється Apache Software Foundation, без участі сторонніх програмістів [3, с. 42].

Теоретично, ядро Apache може функціонувати у чистому вигляді, без використання модулів. Однак функціональність такого рішення вкрай обмежена.

Ядро Apache повністю написано на мові програмування C. Система конфігурації Apache заснована на текстових конфігураційних файлах. Має три умовних рівня конфігурації:

1. Конфігурація сервера (httpd.conf).
2. Конфігурація віртуального хоста (httpd.conf с версії 2.2 extra / httpdvhhosts.conf).
3. Конфігурація рівня директорії (.htaccess).

Має власну мову конфігураційних файлів, заснований на блоках директив. Практично всі параметри ядра можуть бути змінені через конфігураційні файли, аж до управління MPM. Велика частина модулів має власні параметри.

Apache має вбудований механізм віртуальних хостів. він дозволяє повноцінно обслуговувати на одному IP-адреси безліч сайтів (доменних імен), відображаючи для кожного з них власне вміст.

Існує безліч модулів, що додають до Apache підтримку різних мов програмування і систем розробки. До них відносяться:

1. PHP (mod_php).
2. Python (mod_python).

3. Ruby (apache-ruby).
4. Perl (mod_perl).
5. ASP (apache-asp).

Крім того, Apache підтримує механізми CGI і FastCGI, що дозволяє виконувати програми на практично всіх мовах програмування, в тому числі C, C++, Sh, Java.

8. CodeIgniter

CodeIgniter — це потужний PHP-фреймворк з невеликим розміром, створений для розробників, які потребують простого та елегантного інструментарію для створення повнофункціональних веб-додатків. Він відомий своєю легкістю, продуктивністю та відмінною документацією. Розроблений компанією EllisLab, а також Ріком Еллісом (Rick Ellis) та Полом Бурдіком (Paul Burdick). Це готовий каркас для програм, які будуть будуватися на його основі. У цей каркас включені найбільш часто використовувані бібліотеки.

CodeIgniter має дуже невеликий розмір ядра, що робить його одним з найлегших PHP-фреймворків. Це забезпечує високу швидкість завантаження та обробки запитів. Платформа є інструментарієм для тих, хто будує web-програми на PHP. Її мета в тому, щоб дати змогу розробляти програми швидше, ніж якщо б писати код «з нуля», надаючи багатий набір бібліотек для часто використовуваних завдань, а також простий інтерфейс і логічну структуру для доступу до цих бібліотек. CodeIgniter дозволяє творчо зосередитися на проектах, використовуючи мінімальний обсяг коду, необхідний для того чи іншого завдання.

Основне ядро вимагає невеликої кількості бібліотек. Це відрізняє CodeIgniter від більшості фреймворків, які вимагають значно більших ресурсів. Додаткові бібліотеки завантажуються динамічно за запитом, залежно від потреб в той чи інший момент так, що ядро залишається дуже гнучким і швидким.

3. Використання бібліотеки JQuery для динамічних вебсторінок

В наші дні Всесвітня павутина представляє собою динамічну середу, і її користувачі висувають високі вимоги як до оформлення, так і до функціональності сайтів.

Для створення цікавих інтерактивних сайтів розробники використовують бібліотеки JavaScript, такі як jQuery, щоб автоматизувати рішення найбільш типових завдань і спростити рішення більш складних. Однією з причин високої популярності бібліотеки jQuery є її здатність допомагати при вирішенні вельми широкого кола завдань [6].

На перший погляд здається, що складно вибрати, з чого почати, тому що бібліотека jQuery реалізує вельми широкі функціональні можливості. Проте вона має погоджену та симетричну архітектуру; більша частина її концепцій запозичена з HTML і каскадних таблиць стилів (Cascading Style Sheets, CSS). Архітектура бібліотеки швидко освоюється дизайнерами навіть з малим досвідом програмування, оскільки багато веб-розробників, як правило, мають більший досвід роботи із зазначеними технологіями, ніж з JavaScript. У свою чергу досвідченим програмістам буде допомагати і концептуальна цілісність бібліотеки, в чому ми переконуємося пізніше.

1. Огляд та аналіз бібліотеки jQuery, її можливості

Коли люди чують назву jQuery, вони, як правило, вважають, що мова йде про ядро бібліотеки jQuery. Однак під цією назвою також мається на увазі ціла система, що виникла

навколо ядра, яка містить додаткові, супутні бібліотеки, такі як jQuery UI, офіційні розширення і незліченну безліч розширень, створених сторонніми розробниками, які легко можна відшукати в Інтернеті за допомогою будь-якої пошукової системи (Пошук в пошукових системах по фразі «jQuery plugin» принесе вам близько 4 мільйонів посилань).

Причиною всього цього достатку навколо jQuery можна достовірно вважати переваги ядра бібліотеки jQuery.

Мова JavaScript, яка більшу частину свого життя вважалася серед серйозних веб-розробників не настільки серйозною, знайшла нинішній авторитет на хвилі інтересу до повнофункціональних веб-додатків і технології Ajax. Мові довелося дуже швидко розвиватися, так як розробники клієнтських сценаріїв відмовилися від прийому копіювання і вставки програмного коду JavaScript на користь повноцінних бібліотек, які дозволяють вирішувати складні проблеми, пов'язані з відмінностями між браузерами різних типів, і реалізують нові і поліпшені парадигми розробки веб-додатків.

jQuery, що з'явилася в світі бібліотек JavaScript з деяким запізненням, підкорила співтовариство веб-розробників, швидко завоювавши підтримку великих компаній, що використовують веб-додатки для вирішення важливих завдань. У число найбільш помітних користувачів jQuery входять такі компанії, як IBM, Netflix, Amazon, Dell, Best Buy, Twitter, Bank of America і безліч інших відомих компаній.

Компанія Microsoft навіть включила бібліотеку jQuery до складу інструментом, що поставляються разом із середовищем розробки Visual Studio.

І це зовсім не випадково.

На відміну від інших інструментів, які сконцентровані на застосуванні складних методик JavaScript, jQuery прагне змінити уявлення веб-розробників про принципи створення повнофункціональних веб-додатків. Замість того щоб витрачати час на жонгливання непростими можливостями мови JavaScript, розробники отримали можливість за допомогою каскадних таблиць стилів (Cascading Style Sheets, CSS), розширеної мови розмітки гіпертексту (eXtensible Hypertext Markup Language, XHTML) і, звичайно ж, JavaScript безпосередньо маніпулювати елементами сторінки і втілити мрію про швидкій розробці веб-додатків.

Для початку дізнаємося, що насправді може дати jQuery при розробці вебсторінок.

a) Можливості бібліотеки jQuery

Бібліотека jQuery надає багатоцільовий рівень абстракції для вирішення типових задач розробки веб-додатків і тому може застосовуватися практично в будь-яких ситуаціях. Бібліотека має розширювану архітектуру; оскільки постійно з'являються нові розширення і додаються нові можливості, зараз просто не можливо охопити всі функції і допустимі випадки використання jQuery.

Однак базові можливості дозволяють вирішувати наступні завдання:

b) Доступ до елементів документа.

Щоб виконати обхід дерева об'єктної моделі документа (Document Object Model, DOM) і відшукати певні фрагменти HTML без застосування бібліотеки JavaScript, довелося б написати безліч рядків програмного коду. Бібліотека jQuery пропонує надійний та

ефективний механізм селекторів, який дозволяє витягувати необхідні фрагменти документа для подальшого аналізу та модифікації.

c) Зміна зовнішнього вигляду сторінки.

Каскадні таблиці стилів (CSS) пропонують потужний механізм визначення зовнішнього вигляду документа, але він виявляється марним у випадку використання веб-браузерів, які не підтримують єдині стандарти. За допомогою jQuery розробники можуть заповнити цей недолік, спираючись на стандарти, підтримувані всіма браузерами. Крім того, бібліотека jQuery дозволяє змінювати класи або окремі стильові властивості, застосовувані до фрагмента документа, навіть після того, як він буде відображений [16].

d) Зміна вмісту документа.

Бібліотека jQuery дозволяє не тільки виконувати прості косметичні зміни документа, але і дає можливість модифікувати його вміст. За допомогою одного зручного у використанні прикладного програмного інтерфейсу (Application Programming Interface, API) можна змінювати текст, вставляти або змінювати зображення, змінювати порядок списки і навіть повністю змінювати і розширювати структуру документа HTML.

e) Відгук на дії користувача.

Навіть найбільш ретельно розроблені і потужні реалізації поведінки стають даремними, якщо відсутня можливість управління моментом, коли вони повинні запускатися. Бібліотека jQuery пропонує елегантний спосіб, що дає можливість перехоплювати найрізноманітніші події, такі як клацання мишею на посиланні, і не захаращувати при цьому код розмітки HTML оброблювачами подій. Крім того, прикладний інтерфейс механізму обробки подій ліквідує існуючі між браузерами суперечності, які часто викликають почуття досади у веб-розробників.

f) Відтворення анімаційних ефектів в документі.

Для ефективної взаємодії користувача з документом дизайнер повинен забезпечити зворотний візуальний зв'язок. Бібліотека jQuery сприяє вирішенню цього завдання, надаючи безліч анімаційних ефектів, таких як розчинення і стирання елементів, а також зручні інструментальні засоби для реалізації нових ефектів.

g) Отримання інформації з боку сервера без повного оновлення сторінки.

Цей шаблон програмування відомий як асинхронний JavaScript і XML (Asynchronous JavaScript And XML, AJAX) і допомагає веб-розробникам створювати повнофункціональні і швидко реагують сайти. Бібліотека jQuery приховує складнощі, пов'язані з несумісністю браузерів, дозволяючи розробникам сконцентруватися на реалізації функціональності на стороні сервера [13].

h) Спрощення вирішення типових завдань програмування на JavaScript.

На додаток до всіх можливостей, пов'язаних з документами, бібліотека jQuery надає розширення до базових конструкцій JavaScript, таких як обхід масивів в циклі і маніпулювання ними.

i) Фактори успіху jQuery

Новий сплеск інтересу до динамічного HTML привів до виникнення великої кількості платформ JavaScript.

Одні з них є вузькоспеціалізованими інструментами, які приділяють увагу тільки одній або двом із зазначених вище завдань. Інші намагаються запропонувати повний спектр можливостей і анімаційних ефектів і являють собою комплекти всього, що тільки може знадобитися. Щоб забезпечити широке коло можливостей, позначених вище, і залишитися при цьому досить компактною, бібліотека jQuery використовує кілька стратегій:

j) Використання знань про CSS.

Завдяки тому, що в основу механізму пошуку елементів сторінки були покладені селектори CSS, бібліотека jQuery успадкувала короткий і зрозумілий спосіб вираження структури документа. Для дизайнерів, які прагнуть забезпечити інтерактивність своїх сторінок, бібліотека jQuery стала відправною точкою, оскільки знання синтаксису CSS є необхідною і обов'язковою умовою професійної веб-розробки.

к) Підтримка розширень.

Щоб уникнути «розповзання» функціональних можливостей бібліотека jQuery залишає реалізацію особливих випадків за модулями розширення. Порядок створення нових модулів простий і докладно описаний у документації, що сприяє розробці різноманітних винахідливих і корисних модулів. Навіть багато базових функціональних можливостей бібліотеки jQuery реалізовано з використанням механізму розширень і вони можуть видалятися з бібліотеки по мірі необхідності, що дозволяє ще сильніше зменшувати її розмір [10].

l) Абстрактний спосіб обходу несумісностей браузерів.

Суворі дійсності веб-розробки така, що кожен браузер має свої відхилення від загальноприйнятих стандартів. Значну частину будь-якого вебдодатку становить реалізація функціональних особливостей для кожної з платформ. Внаслідок неможливості реалізовувати деякі особливості однаково для всіх браузерів, бібліотека jQuery додає рівень абстракції, що дозволяє уніфікувати рішення типових завдань, зменшити обсяг програмного коду і істотно спростити його.

м) Завжди працює з наборами.

Коли ми наказуємо бібліотеці jQuery відшукати всі елементи з класом collapsable і приховати їх, нам не потрібно проводити обхід всіх елементів в циклі, тому що методи, такі як .hide(), автоматично працюють не з окремими об'єктами, а з їх наборами. Завдяки цьому прийому, званому неявною ітерацією, багато конструкцій циклу стають непотрібними, що призводить до істотного скорочення обсягу програмного коду.

п) Дозволяє виконувати безліч операцій в одному рядку.

Щоб уникнути надмірного використання тимчасових змінних або непотрібних повторів бібліотека jQuery для основних своїх методів використовує шаблон програмування, що зветься зчепленням. Завдяки цьому результатом більшості операцій над об'єктом є сам об'єкт, готовий до виконання наступної операції [25].

Застосування цих стратегій дозволило зберегти малий обсяг бібліотеки (стиснений файл займає менше 20 Кбайт) і в той же час забезпечити компактність спеціалізованого програмного коду, що використовує цю бібліотеку.

Елегантність бібліотеки забезпечується частково архітектурними рішеннями, а частково еволюційним процесом її розвитку, що підтримуються енергійним співтовариством, яке

виникло навколо цього проекту. Користувачі jQuery обговорюють не лише розробку модулів розширення, але і поліпшення самого ядра бібліотеки.

Незважаючи на значні зусилля, необхідні для розробки настільки гнучкої і надійної системи, кінцевий продукт залишається безкоштовним для будь-яких видів використання. Цей відкритий проект поширюється під подвійний ліцензією: GNU Public License (як і багато інших відкритих проектів) і MIT License (з метою сприяння використанню jQuery в створенні пропрієтарного програмного забезпечення).

о) *Ненав'язливий JavaScript*

До появи CSS, веб-розробники були змушені змішувати в HTML-сторінках стильову розмітку зі структурою документа. Додавання CSS в арсенал інструментів веб-розробника дозволяє відокремити стилістичну інформацію від структури документа і проводити на заслужений відпочинок такі теги, як ``. Відділення стилю від структури не тільки спрощує управління документами, а й надає їм гнучкість, дозволяючи добитися повної зміни стилю відображення сторінки простою заміною таблиці стилів.

До цих пір звичайної залишається розмітка на зразок цієї:

```
<button                                type="button"
onclick="document.getElementById('xyz').style.col
or='red';">
    Click Me
</button>
```

Тут ми можемо побачити, що стиль елемента-кнопки, включаючи шрифт напису на ній, визначається не міткою `` або іншої небажаної розміткою стилю, а задається правилами CSS, діючими в межах сторінки.

У цьому оголошенні не змішується розмітка стилю і структура, однак тут у наявності змішування поведінки структури за рахунок включення коду JavaScript, виконуваного по клацанню по кнопці, в код розмітки елемента `<button>` (в даному випадку клацання по кнопці викликає фарбування в червоний колір деякого елемента об'єктної моделі документа (Document Object Model, DOM), атрибут `id` якого має значення `xyz`).

Розглянемо, як можна було б поліпшити цю ситуацію.

р) *Відділення поведінки від розмітки*

З тих же самих причин, по яких бажано відокремлювати стиль від розмітки HTML-документа, також бажано було б відокремити поведінку елементів від їх розмітки.

В ідеалі HTML-сторінка повинна мати структуру, як показано на рис. 1.2, де розмітка, інформація про стилі та реалізація поведінки знаходяться в своїх окремих нішах.

Рис. Відділення елементів, що визначають структуру, стиль і поведінку компонентів сторінки, підвищує читабельність вихідного коду і спрощує його супровід.

Ця стратегія, відома під назвою ненав'язливий JavaScript, була запропонована розробниками jQuery і тепер підтримується всіма великими бібліотеками JavaScript, допомагаючи авторам сторінок досягати цього відділення при розробці сторінок. Бібліотека

jQuery зробила цю стратегію популярною, і її ядро вибудовано так, що ненав'язливий JavaScript підтримується дуже органічно.

Ненав'язливий JavaScript припускає, що присутність будь-яких висловів чи інструкцій на мові JavaScript в тезі <body> HTML-сторінок помилкова - як в атрибутах HTML-елементів (наприклад, onclick), так і в блоках сценаріїв в тілі сторінки [28].

q) *Відділення сценарію*

Замість того щоб вбудовувати визначення поведінки кнопки в її розмітку, можна помістити його в блок сценарію в розділі <head> сторінки за межами тіла документа, як показано нижче:

```
<script type="text/javascript"> window.onload =  
function()  
document.getElementById('testButton').onclick =  
function()  
document.getElementById('xyz').style.color =  
'red';  
};  
};  
</script>
```

У обробнику події сторінки onload ми пов'язується функцію з атрибутом onclick елемента-кнопки.

Обробник події onload (а не вбудований код) використовується щоб елемент-кнопка вже існував до того моменту, коли розробник спробує маніпулювати ним.

Ненав'язливий JavaScript - це потужна методика, що дозволяє розділити обов'язки в веб-додатках, але за це доводиться платити.

Можливо, ви вже звернули увагу, що для досягнення поставленої мети нам знадобилося додати трохи більше рядків коду, ніж у випадку, коли код JavaScript містився безпосередньо в код розмітки. Ненав'язливий JavaScript не тільки може привести до збільшення обсягу програмного коду, але також вимагає певної дисципліни та застосування добре зарекомендували себе шаблонів програмування клієнтських сценаріїв.

Але в цьому немає нічого поганого - все, що спонукає веб-розробника писати свій клієнтський код так само уважно і акуратно, як зазвичай пишеться код, що розміщується на сервері, йде йому тільки на користь.

Однак якщо не використовувати jQuery, то у вас з'являється зайва робота.

Як вже говорилося, розробники jQuery зосередили свої зусилля на те, щоб спростити нам програмування сторінок із застосуванням методики ненав'язливого JavaScript, не сплачуючи за це зайвими зусиллями і зайвим програмним кодом. Ефективне використання jQuery дозволяє нам привнести більше можливостей в наші сторінки, створюючи сценарії меншого обсягу.

А тепер розглянемо, як бібліотека jQuery дозволяє розширювати функціональність сторінок без зайвих зусиль.

r) Основи jQuery

Сутність jQuery в тому, щоб відбирати елементи HTML-сторінок і виконувати над ними деякі операції. Якщо ви знайомі з CSS, то добре розумієте, наскільки зручні селектори, які описують групи елементів, об'єднані за якимись атрибутами або по місцю розташування в документі. Завдяки jQuery ви зможете, використовуючи свої знання, значно спростити код JavaScript.

Бібліотека jQuery в першу чергу забезпечує несуперечливу роботу програмного коду у всіх основних типах браузерів, вирішуючи такі складні проблеми JavaScript, як очікування завантаження сторінки, перед тим як виконати будь-які операції.

На той випадок, якщо в бібліотеці виявиться недолік функціональності, розробники передбачили простий, але дуже дієвий спосіб її розширення. Багато починаючих програмістів jQuery виявляють цю гнучкість на практиці, розширюючи можливості бібліотеки в перший же день.

Подивимось, як знання CSS допомагає створити короткий, але потужний програмний код.

s) Обгортка jQuery

З введенням CSS в веб-технології з метою відокремити подання від вмісту знадобився спосіб, що дозволяє посилатися на групи елементів сторінки з зовнішніх таблиць стилів. В результаті був розроблений метод, заснований на використанні селекторів, які представляють елементи на основі їх атрибутів або розташування в HTML-документі.

Селектори CSS є досить короткими і вважаються досить простими в розумінні.

Наприклад, селектор `p` а посилається на всі посилання (елементи `<a>`), вкладені в елементи `<p>`.

Бібліотека jQuery використовує ті ж самі селектори і підтримує не тільки звичайні селектори, які застосовуються сьогодні в CSS, але й інші, ще не повністю реалізовані в більшості браузерів, включаючи селектори, які визначаються специфікацією CSS3 [7].

Щоб відібрати групу елементів, достатньо передати функції jQuery простий селектор:

`$(selector)`

чи `jQuery(selector)`

Функція `$ ()`, на перший погляд, незвичайна, але більшість користувачів швидко починають застосовувати її завдяки її стислості. Наприклад, отримати групу посилань, вкладених в елементи `<p>`, можна наступним способом:

`$("p a")`

Функція `$ ()` (псевдонім функції `jQuery ()`) повертає спеціальний об'єкт JavaScript, який містить масив елементів DOM, відповідних вказаному селектору. У цього об'єкта багато зручних зумовлених методів, здатних впливати на групу елементів. На мові програмування такого роду конструкція називається обгорткою (*wrapper*), тому що вона «обгортає»

відібрані елементи додатковою функціональністю. Ми будемо використовувати термін обгортка jQuery, або обгорнутий набір, посилаючись на групи елементів, управляти якими дозволяють методи, визначені в jQuery.

Припустимо, нам потрібно реалізувати поступове зникнення всіх елементів `<div>` з класом CSS `notLongForThisWorld`. jQuery дозволяє зробити це так:

```
$ ("div.notLongForThisWorld").hide();
```

Особливість багатьох з цих методів, часто званих методами обгортки jQuery, полягає в тому, що по завершенні своїх дій (наприклад, дії, що забезпечує зникнення) вони повертають ту ж саму групу елементів, готову до виконання іншої операції.

Припустимо, що після того як елементи зникнуть, до них потрібно додати клас CSS `removed`. Записати це можна так:

```
$ ("div.notLongForThisWorld").hide().addClass("removed");
```

Такий ланцюжок (chain) методів jQuery можна продовжувати до нескінченності. Можна без зусиль відшукати в Інтернеті приклади ланцюжків jQuery, що складаються з десятків методів. А оскільки кожна функція працює відразу з усіма елементами, відповідними вказаним селектору, не потрібно виконувати обхід масиву елементів в циклі. Все потрібне відбувається у фоновому режимі [14].

t) Хронологія розвитку проекту jQuery

Головна мета бібліотеки - забезпечити простий спосіб пошуку елементів веб-сторінки і маніпулювання ними - залишається незмінною в ході розробки, але деякі особливості синтаксису і можливості змінюються від версії до версії. У наведеному далі короткому огляді хронології розвитку проекту описуються найбільш важливі зміни, що відбувалися від версії до версії.

Етап громадської розробки: у серпні 2005 року Джон Резіг (John Resig) вперше оголосив про вдосконалену бібліотеку «Behaviour», засновану на бібліотеці Prototype. Офіційно нова платформа була випущена 14 січня 2006 під назвою jQuery.

jQuery 1.0 (серпень 2006): ця перша стабільна версія бібліотеки вже мала надійну підтримку селекторів CSS, технології AJAX і механізму подій.

jQuery 1.1 (січень 2007): у цій версії була проведена істотна модернізація прикладного програмного інтерфейсу (API). Багато рідко використовувані методи були об'єднані, що призвело до зменшення числа методів, які необхідно вивчати і документувати.

jQuery 1.1.3 (липень 2007): у цій версії була істотно підвищена продуктивність механізму селекторів jQuery. Починаючи з цієї версії, продуктивність бібліотеки jQuery досягла рівня продуктивності споріднених бібліотек JavaScript, таких як Prototype, Mootools і Dojo.

jQuery 1.2 (вересень 2007): з цієї версії був прибрааний синтаксис XPath вибору елементів, оскільки він став надлишковим при наявності синтаксису CSS. У цій версії стало більш гнучким налаштування ефектів, а завдяки додаванню логіки управління подіями у просторах назв спростилося розробка розширень.

jQuery UI (вересень 2007): цей новий набір розширень був випущений, щоб замінити популярний, але застаріваючий модуль Interface. У нього була включена багата колекція

готових віджетів, а також ряд додаткових інструментів для створення складних елементів, таких як інтерфейси буксирування елементів мишею (drag-and-drop).

jQuery 1.2.6 (травень 2008): до складу основної бібліотеки були включені функціональні можливості популярного модуля розширення Dimensions, створеного Бренденом Аароном (Brandon Aaron).

jQuery 1.3 (січень 2009): значна модернізація механізму селекторів (Sizzle) забезпечила гігантський приріст продуктивності бібліотеки. Офіційно було оголошено про підтримку делегування подій.

jQuery 1.4 (січень 2010): оптимізація обробки подій, реалізація. Closest (Array). Швидкодія основних функцій збільшено більш ніж у 3 рази.

jQuery 1.5 (січень 2011): переписаний модуль AJAX, додана можливість створення відкладених (deferred) об'єктів, метод. Sub () для клонування jQuery, велика швидкість виконання методів. Children (), Prev () і Next ().

jQuery 1.6 (лютий 2011): виправлення 4-х помилок; Збільшена продуктивність методу data (); Зміни, «ламають» сумісність (змінені методи

`data ()`, `attr ()`, `prop ()`)

jQuery 1.8.3 (листопад 2012): виправлення помилок.

jQuery 1.9 (січень 2013): видалення застарілих інтерфейсів та чистка коду.

jQuery 2.0 (квітень 2013): припинення підтримки IE 6-8 для збільшення продуктивності та зменшення розміру файлів [28].

2. Огляд існуючих плагінів під jQuery.

Бібліотека jQuery написана так, що дозволяє легко додавати до неї нові функціональні можливості. Модулі, що додають нові функціональні можливості, називаються розширеннями (plug-in).

Велику кількість таких плагінів можна відшукати на сайті <http://plugins.jquery.com>.

Розширення для бібліотеки jQuery є звичайними файлами з програмним кодом на мові JavaScript. Для того, щоб задіяти розширення у своїх вебсторінках, достатньо просто підключити їх за допомогою елемента <script>, так само як будь-яку іншу бібліотеку на мові JavaScript.

Звичайно, розширення повинні підключатися після підключення самої бібліотеки jQuery.

В цьому розділі ми розглянемо два найбільш використовуваних плагіна JQuery, які реалізують для інтерактивного верстання таблиць: DataTables та JQGrid; DataTables.

Це дуже гнучкий інструмент, заснований на основі прогресивного поліпшення, який додає розширені можливості управління взаємодією будь-якої HTML-таблиці.

Основні можливості:

- Змінна довжина нумерації сторінок;
- Фільтрація на ходу;
- Сортування за кількома стовпцями з визначенням типу даних;

- Смарт обробка ширини стовпців;
- Відображення даних практично з будь-якого джерела даних;
- DOM, JavaScript масив, Ajax файлів і обробки на стороні сервера (PHP, C #, Perl, Ruby, AIR, зубчасті передачі тощо);
- Прокрутка варіантів перегляду таблиці;
- Підтримка JQuery UI ThemeRoller;
- Можливість приховувати стовпці;
- Динамічне створення таблиць;
- Ajax автоматичне завантаження даних;
- Альтернативні види нумерації сторінок;
- Сортування стовпців та їх підсвічування;
- Сортування, визначення типу, API функцій, нумерація сторінок та фільтрація.

DataTables працює за принципом прогресивного розвитку, в результаті чого таблиця буде представлена більш міцною і інтерактивною для кінцевого користувача, якщо їх браузер має необхідні можливості. При ініціалізації об'єкта `jQuery.dataTable`, інформація про таблицю зчитується безпосередньо зі сторінки HTML. У поєднанні зі значеннями за замовчуванням для функції DataTables, це робить її легко інтегрованою безпосередньо в сайт або вебдодатки. За бажанням, можна використовувати параметри ініціалізації для завантаження даних з інших місць, окрім DOM, таких як серверний сценарій обробки або отриманий за допомогою Ajax JSON-файл.

Для того, щоб DataTables, мав можливість правильно функціонувати, HTML-код для цільової таблиці, повинен бути розташований з оголошенням секцій THEAD і TBODY". Наприклад:

```
<table id="table_id">
  <thead>
    <tr>
      <th>Column 1</th>
      <th>Column 2</th>
      <th>etc</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Row 1 Data 1</td>
      <td>Row 1 Data 2</td>
      <td>etc</td>
    </tr>
```

```
<tr>
  <td>Row 2 Data 1</td>
  <td>Row 2 Data 2</td>
  <td>etc</td>
</tr>
</tbody>
</table>
```

DataTables може приймати дані, які він буде відображати, з багатьох різних джерел. Це означає, що нам не обов'язково обмежувати DataTables одним конкретним шляхом, що забезпечує більшу гнучкість.

Існують чотири основні методи надання даних DataTables:

1. DOM (тобто HTML таблиці на сторінці)
2. JavaScript масив
3. Ajax Джерело - серверний файл, із збереженням вихідного форматування JSON
4. Обробка на стороні сервера - сервер, де буде мати справу з нумерацією сторінок, фільтрацію і т.д.

DOM. На базовому рівні ми можемо дати DataTables посилання на таблицю, яка вже існує у нашій HTML сторінці. DataTables буде читати всю інформацію про таблицю зі сторінки (DOM) і додавати такі функції, як фільтрація, розбиття та сортування.

JavaScript масив. Це дає можливість дати DataTables інформацію, яку ви хочете відобразити в таблиці як 2D масив JavaScript(тобто масив масивів). Це корисно, коли ваші дані обробляються JavaScript або при динамічному додаванні таблиці в сторінку.

Ajax Джерело. Якщо дані, які ми хочемо відобразити, доступний з сервера і ще не в браузері, ми можемо використати DataTables, щоб витягнути дані назад з сервера для відображення. Також це корисно, якщо виникає ситуація, коли потрібно показувати “живу” інформацію, яку можна було б періодично оновлювати.

Обробка на стороні сервера. Якщо веб-розробник має справу з великими обсягами даних (наприклад, 20 мільйонів рядків) веб-браузер просто не справляються з кількістю обробки, яка потрібна для DataTables. Замість цього можна обробляти дані у процесі, спеціально розробленому для цього - а саме використовувати бази даних SQL (або будь-яке інше джерело даних). Серверний процес буде робити все - нумерацію сторінок, сортування, фільтрацію і т.д., в той час як DataTables просто покаже результати обробки взаємодії з користувачем.

4. JQGrid

JQGrid підтримує AJAX JavaScript управління та пропонує рішення для представлення та управління табличними даними в мережі. Оскільки таблиця на стороні клієнта завантажується динамічно через зворотні виклики Ajax, вона може бути інтегрована з будь-якою серверної технологією, в тому числі PHP, ASP, Java-додатками, JSP, ColdFusion, і Perl.

JQGrid є дуже гнучким плагіном. Ось деякі з його особливостей:

JQGrid послідовно були розроблені з простотою використання на увазі.

Користувач повинен відчувати себе вільно від установки до адміністрації.

Підтримка CSS-тем. JQuery сумісний з UI Теми оформлення. Розробники можуть змінити тему таблиці, за допомогою користувацького інтерфейсу CSS Framework.

Швидкість. Новий движок підвищує швидкість завантаження в деяких випадках до 5-10 разів швидше, ніж попередні релізи.

Перегортання сторінок. Ця функція дозволяє розробнику відображати дані сторінка за сторінкою. Дані не будуть отримані з сервера, поки кінцевий користувач не зайде на цю сторінку. Розробники можуть контролювати, скільки рядків відображаються одночасно.

Змінювані стовпці. Ширина стовпчика може бути змінена користувачем.

Сортування за різними типами даних. Кінцевий користувач може відсортувати записи, натиснувши на заголовок стовпця. Розробники можуть вказати сортування для різних типів, включаючи рядки, числа, логічні, посилання та інші.

Обробники подій і користувацького API. Обробники подій і різні методи надають велику гнучкість без великої кількості кодування.

Автоматичне завантаження даних при прокручуванні. Ця функція дозволяє завантажувати дані без підкачки і з використанням тільки вертикальної смуги прокручування.

Робота з локальними даними. JQuery може працювати з локальними даними, визначаючи їх, як масив.

Інтеграція з будь-якими рішеннями на стороні сервера, такими як ASP, PHP і Perl.

Підтримка декількох мов. В даний час JQuery підтримує більше 20 мов.

Підтримка XML, JSON і масивів в якості джерел даних. Ще одна цікава особливість - це можливість визначити свої власні типи даних.

SubGrids. Використання підтаблиці - це найпростіший спосіб відображення даних з дочірніх записів.

Метод форматування має зумовлені типи, такі як: ціле число, номер, дата, валюта, посилання, прапорці і т.д.

Розробник може визначати свої власні типи форматування.

Підтримку тексту, текстової області, прапорця, варіантів вибору, зображення, радіокнопки.

Перевірка введених даних на стороні клієнта.

Навігація з клавіатури вкладці зрушення вкладки, вліво, вправо, вгору і вниз.

Повний контроль над формою від створення до закриття.

Панель інструментів пошуку.

Складна форма пошуку з критеріями, визначеними користувачем.

Підтримка форматів XML, JSON.

Можливість імпорту конфігурації і даних.

5. Висновки до розділу 2

Мова JavaScript знайшла нинішній авторитет на хвилі інтересу до повнофункціональних веб-додатків

Щоб автоматизувати рішення найбільш типових завдань і спростити рішення більш складних, розробники використовують бібліотеки JavaScript, такі як jQuery для розробки інтерактивних елементів сторінки.

Бібліотека jQuery реалізує вельми широкі функціональні можливості.

На відміну від інших інструментів, які сконцентровані на застосуванні складних методик JavaScript, jQuery прагне змінити уявлення веброзробників про принципи створення повнофункціональних веб-додатків.

Бібліотека jQuery надає багатоцільовий рівень абстракції для вирішення типових задач розробки веб-додатків і тому може застосовуватися практично в будь-яких ситуаціях. Бібліотека має розширювану архітектуру; оскільки постійно з'являються нові розширення і додаються нові можливості, зараз просто не можливо охопити всі функції і допустимі випадки використання jQuery.

Елегантність бібліотеки забезпечується частково архітектурними рішеннями, а частково еволюційним процесом її розвитку.

Головна мета бібліотеки - забезпечити простий спосіб пошуку елементів веб-сторінки і маніпулювання ними - залишається незмінною в ході розробки, але деякі особливості синтаксису і можливості змінюються від версії до версії.

Бібліотека jQuery написана так, що дозволяє легко додавати до неї нові функціональні можливості. Модулі, що додають нові функціональні можливості, називаються розширеннями (plug-in).

Ми розглянули найпопулярніші засоби, що використовують при розробці веб-порталів. Для створення бакалаврського дослідження обрали ще два найбільш використовуваних плагіна бібліотеки JQuery, що реалізують динамічне верстання таблиць: DataTables та JQGrid, їх можливості та специфіку роботи з ними

XIII. РОЗДІЛ 3 ПРОЕКТУВАННЯ ТА РОЗРОБКА НОВИННОГО ПОРТАЛУ NEWS RAPTOR

1. Постановка задач та функціональні вимоги до проекту

Весь процес проектування сайту розділено на кілька послідовно розставлених етапів:

- провести аналіз дослідження та розробити макет новинного порталу News Raptor;
- розробити діаграму прецедентів (Додаток В);
- розробити ER діаграму бази даних та створити базу даних на основі отриманих діаграм;
- провести верстку шаблону сайту на основі розробленого макета;
- розробити інтерфейс для клієнтської частин сайту;
- провести тестування.

Детальніше розглянемо кожен з представлених етапів. Успішність готового проекту залежить безпосередньо від етапу проектування, а успіх в проектуванні залежить в першу чергу від правильно сформульованих і поставлених завдань. Для виконання проекту виділено основні завдання:

1. Провести аналіз основних вимог щодо розроблюваного сайту;
2. Сформулювати основні функціональні характеристики;
3. Опрацювати usability сайту;
4. Провести SEO оптимізацію проекту.

Провівши аналіз завдань поставлених перед розроблювальним проектом, прийшли до наступного, веб-портал повинен забезпечувати можливість виконання наступних функцій:

- ведення списків користувачив;
- введення нової і корекцію поточної інформації;
- зберігання інформації;
- отримання відомостей про первинні новини.

Сайт повинен передбачати базовий захист від основних видів атак: міжсайтового скриптинга (XSS), SQL -ін'єкцій, CSRF-вразливостей, а так само захищену авторизацію адміністратора сайту засобами СМС пароля.

Для програмної реалізації бакалаврського дослідження використані сучасні технології (див. таблицю 3.1), у другому розділі було докладно описано можливості обраних середовищ.

MVC фреймворк	CodeIgniter 4.5.2
СУБД	MySQL
CSS Бібліотеки	Bootstrap 5.3.3 (CDN) FontAwesome 6.5.2 (CDN) Google Fonts (CDN)
JS Бібліотека	JQuery 3.7.1 (CDN)
Бібліотека Email-розсилки	PHPMailer 6.9.1
PDF Бібліотека	TCPDF 6.7.5
Сервіс збереження аватарів	Gravatar

Таблиця – 3.1. Використані технології

Новинний портал News Raptor надає інформацію за допомогою основних прототипізованих веб-сторінок, що утворюють розділи. Веб-додаток спирається на сторонні бібліотеки які виконанні за кращими стандартами розробки ПЗ для доступної і простої подальшої модифікації. PHP Mailer та TCPDF необхідні для розсилки email та створення PDF-файлів відповідно. Також додаток завантажує CSS та JS ресурси з хмарного сховища для frontend частини.

Далі наведена схема процесу інтеграції компонентів веб-додатку:

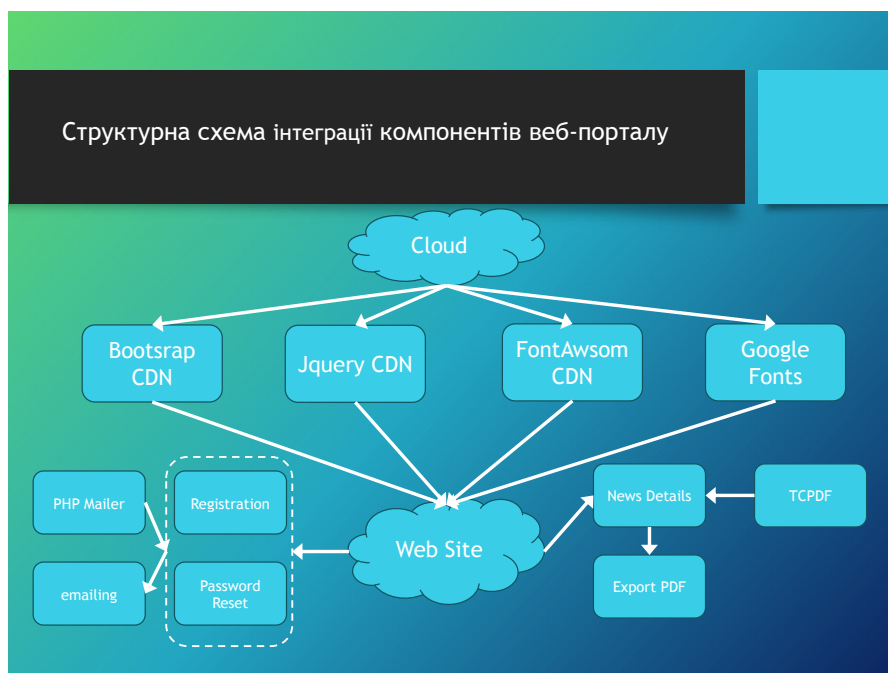


Рис. 3.1 Структурна схема інтеграції компонентів веб-порталу

В даний час більшість веб проєктів розробляється відповідно до якоїнебудь методологією розробки ПО як наслідок, розробникам потрібно інструмент для моделювання даних на етапах аналізу і проектування. Та ким інструментом є ER - діаграми (Entity-Relationship, «Сутність-Зв'язок»).

Фактично їх використання є обов'язковим при розробці ІС, систем прийняття рішень, систем електронної торгівлі і B2B - більшості бізнес орієнтованих систем. ER - діаграми дозволяють будувати моделі логічної структури даних предметної області, а так само робити моделювання фізичної структури систем зберігання даних.

На рисунку 3.2 представлена ER-діаграма логічної моделі даних проєкту з деталізацією до рівня сутностей. Вона несе інформацію про склад об'єктів даних і існуючих між ними зв'язків.

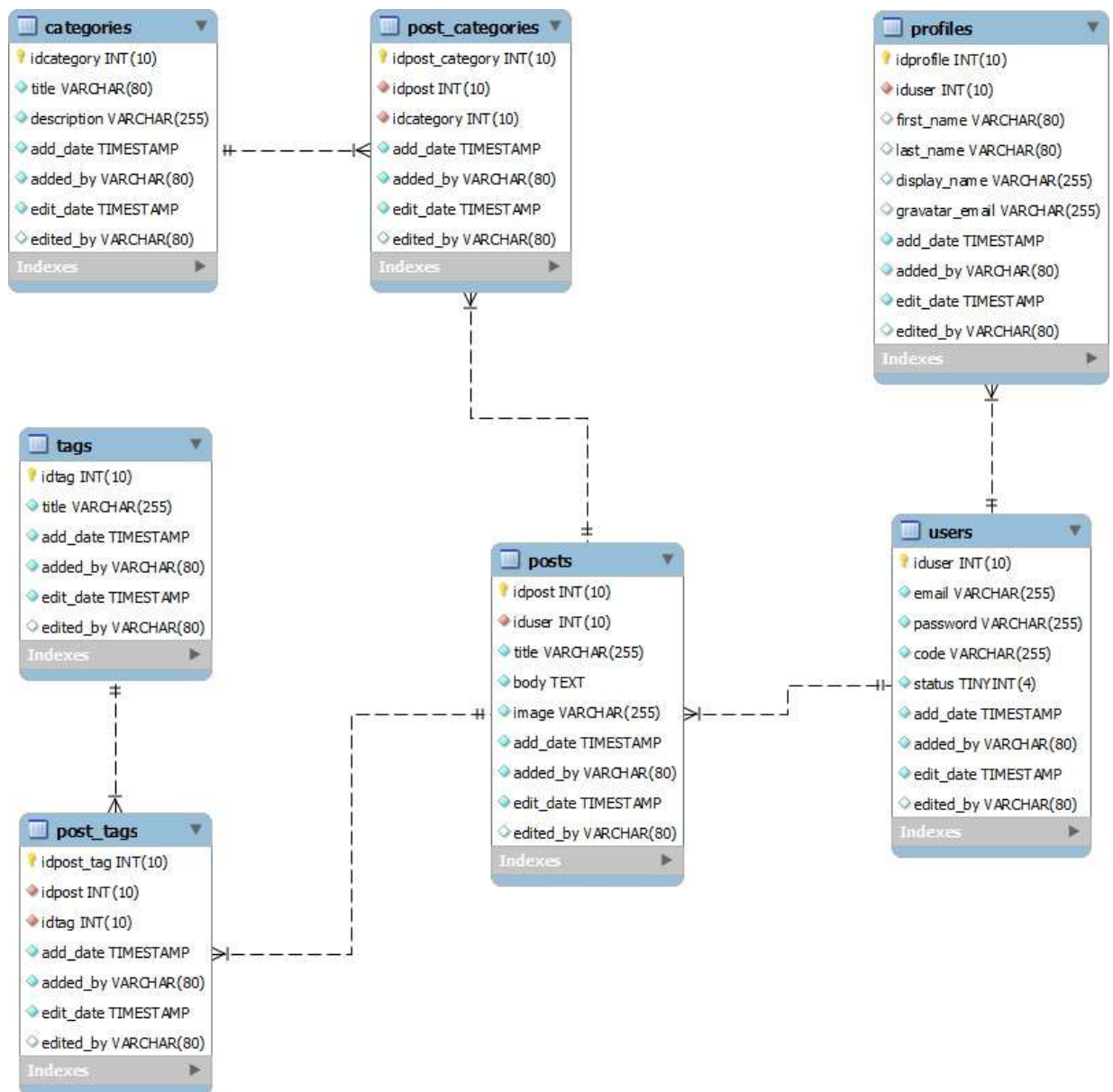


Рис. 3.2 Структура бази даних

Розробка веб-системи починається з проектування бази даних та видів інформації, що зберігається. Для виконання проектування структури бази даних та створення був обрано інструмент phpMyAdmin. PhpMyAdmin — вебдодаток з відкритим кодом на мові PHP із графічним веб-інтерфейсом для адміністрування бази даних MySQL.

База даних для додатку складається з 7 таблиць які зв'язані між собою зовнішніми ключами.

categories	-	таблиця категорій;
post_categories	-	таблиця використання категорій;
tags	-	таблиця тегів;
post_tags	-	таблиця використання тегів;
posts	-	таблиця постів;
profile	-	таблиця профілів;
users	-	таблиця користувачів.

2. Реалізація модулів новинного порталу

При реалізації проекту було використано функції, які є загальними для усіх скриптів новинного веб-порталу. Ці функції винесені в окремі модулі, які підключаються при запуску порталу, щоб не переписувати постійно код. Вхід при запуску з локального сервера здійснюється за адресою [http:// news_portal /index.php](http://news_portal/index.php).

Модуль index.php формує головну сторінку (рис.3.3) новинного порталу News Raptor: відкриває сесію, підключає головну сторінку, модулі та в залежності від ролі користувача відкриває сторінку адміністрування чи головну сторінку новинного порталу.

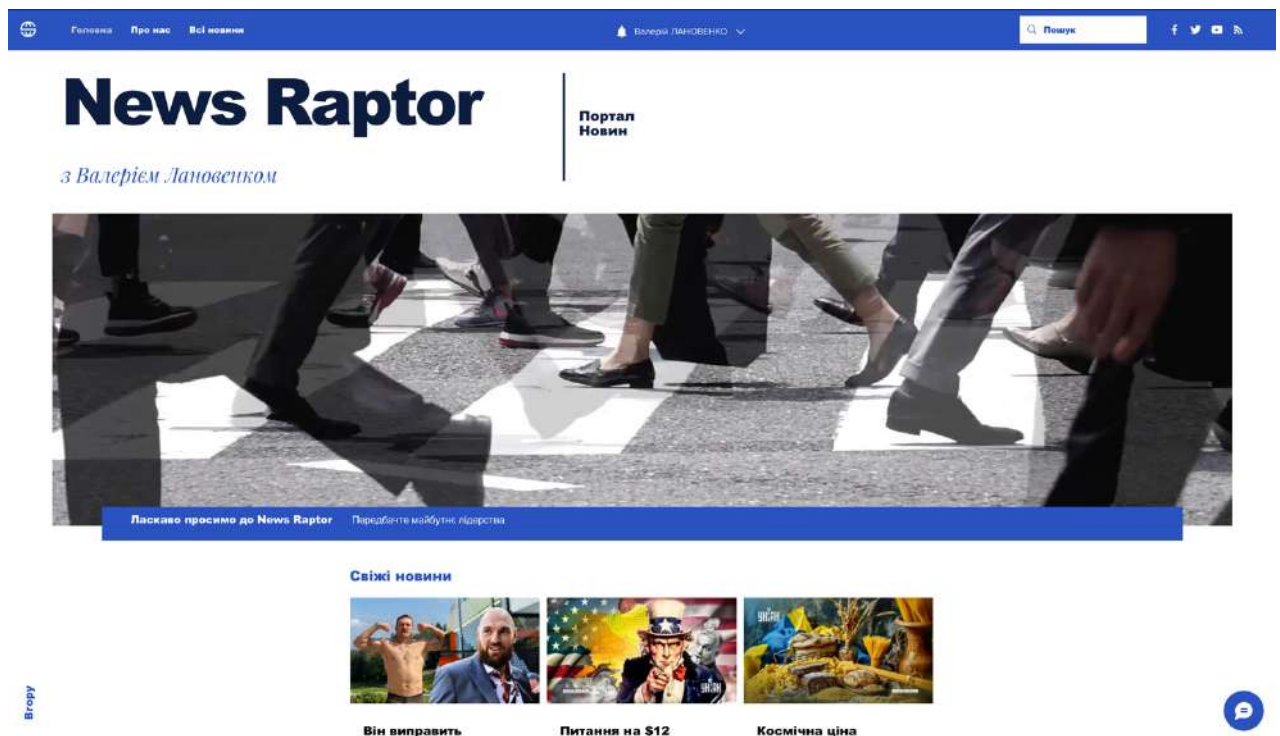


Рис. 3.3. Головна сторінка News Raptor (спочатку найсвіжіша новина)

Основні розділи порталу:

- головна сторінка;
- Всі новини;
- керування профілем;
- Про нас.

Кожен розділ має верхню і нижню незмінну частину.

Відвідувачі порталу обмежені переглядом не більше ніж 10 новин. Вони можуть передивлятись останні 10 новин, підписатись на RSS-канал або експортувати у PDF дозволені новини. Для доступу до розширених опцій таких як фільтр по категорії або приміткам, пошук і т.д. необхідно зареєструватись. На рис.3.4 надано схему робочого процесу користувачів. Тільки зареєстровані користувачі можуть вивішувати новини.

На даному етапі відсутня адміністративна панель. У подальшій розробці можливо додати адміністративну зону де користувачі зможуть керувати.



Рис.3.4 Схема робочого процесу користувачів

Навігаційне меню однаково підходить для повної і мобільної версії новинного порталу (рис.3.5). Навігаційне меню служить для переходу між розділами порталу.

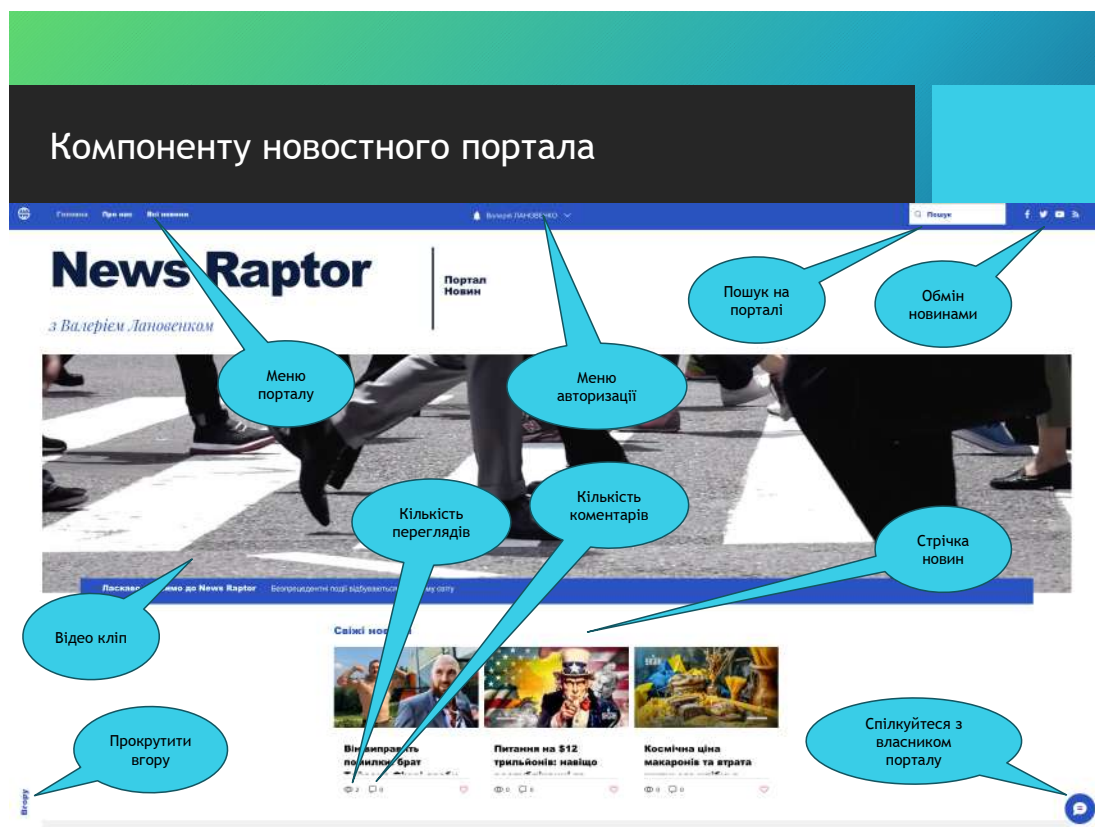


Рис. 3.5. Компоненту новостного портала: навігаційне меню

У нижній частині порту, під рискою розташовується авторський підпис розробника (рис.3.6) сайту і посилання для підписки на RSS-стрічку



Рис. 3.6. Компонент новинного порталу: нижнє меню

Службовий розділ передбачається тільки в повній версії порталу. Стрічка новин (головна сторінка) порталу містить список новин, розташованих в порядку їх додавання (актуальності новини). Для повної версії список новин на одну сторінку обмежується 10 новинами. Отримання інших новин здійснюється за допомогою посторінкової навігації (рис.3.7).



Рис.3.7. Елемент сторінки стрічки новин: посторінкова навігація

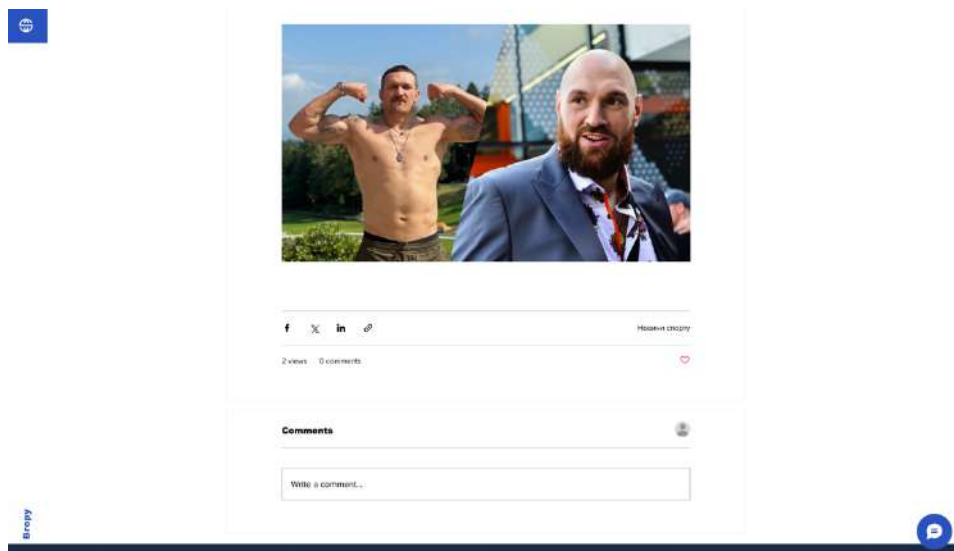


Рис.3.8. Список коментарів новини, залишених користувачами

До кожної новини користувач порталу може здійснювати коментарі та пости, обмінюючись думками про новину з іншими зареєстрованими користувачами.

Коментар користувач може написати через спеціальну форму (рис.3.9) для цього потрібно зареєструватися (рис.3.10),

 A screenshot of the 'News Raptor' website's account settings page. The top navigation bar is blue with links for 'Головна', 'Про нас', and 'Всі новини'. A user profile 'Валерій ЛАНОВЕНКО' is shown. The main header features the 'News Raptor' logo and the tagline 'з Валерієм Лановенком'. The page title is 'Портал Новин'. The 'Account Settings' section has tabs for 'My Account', 'My Addresses', 'My Wallet', and 'Settings'. The 'Account' section includes a 'Discard' button and an 'Update Info' button. The 'Display Info' section has a text input for 'Display name' (filled with 'Валерій ЛАНОВЕНКО') and a 'Title' input. The 'Personal Info' section has inputs for 'First name', 'Last name', and 'Phone', each with a lock icon. There are 'Discard' and 'Update Info' buttons at the bottom right.

Рис.3.10. Форма додавання реєстрації

Схему етапів процесу реєстрації показано на рисунку 3.11:

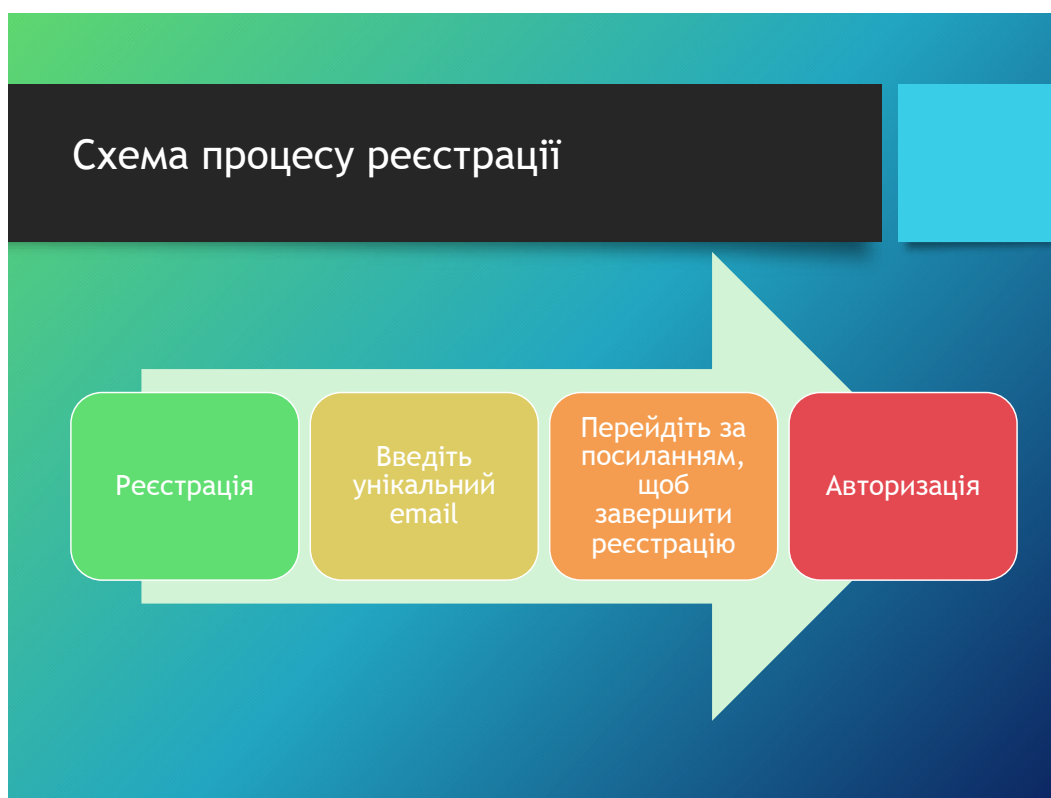


Рис. 3.11. Схема процесу реєстрації

Для реєстрації створюється особистий акаунт, заповнюється форма акаунта та сторінка керування профілем, де указуються унікальна е-адреса, прізвище та ім'я (рис.3.12-3.13).

The screenshot shows a web form titled "Sign Up". Below the title is a link: "Already a member? Log In". The form contains two input fields: "Email" and "Password". Below these fields is a CAPTCHA section with a green checkmark and the text "I'm not a robot", followed by a CAPTCHA image and the text "Введіть код". At the bottom of the form is a blue button labeled "Sign Up".

Рис. 3.12. Форма створення акаунта

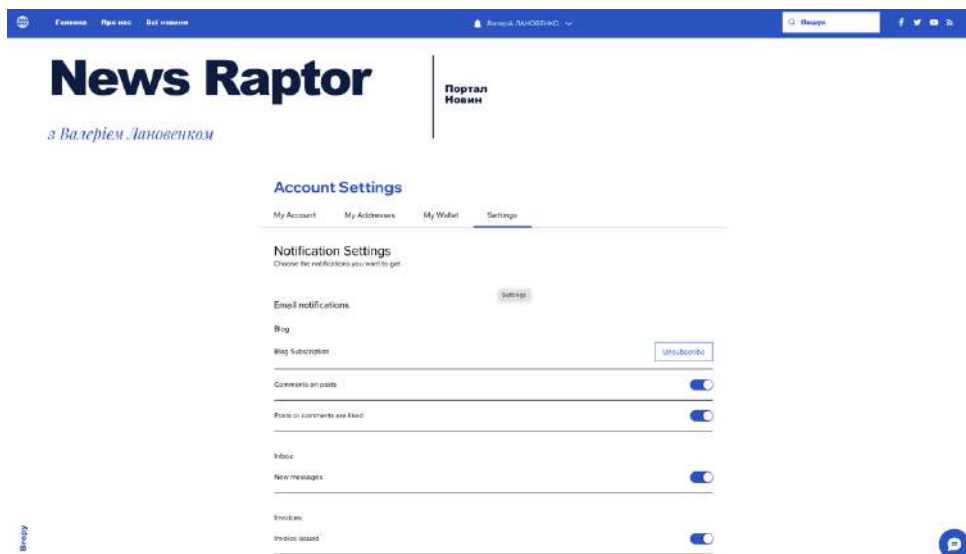


Рис. 3.13. Сторінка керування профілем

Після введення ім'я користувача і пароля, користувач отримує доступ до зміни змісту сторінки, шляхом додавання, редагування або видалення вмісту поточної сторінки, про що користувач отримує повідомлення під навігаційним меню (рис. 3.9).

Редагування поста новини здійснюється шляхом переходу за посиланням Редагувати. Після чого користувач перенаправляється на сторінку по роботі з вмістом новини, де, в разі успішно переданого ключа новини, здійснюється заповнення відповідних полів, з метою редагування. Кількість особистих публікацій можливо передивитись на сторінці списку публікацій.

Щоб застосувати внесені зміни в новини, користувач повинен натиснути кнопку Ок, після чого він буде перенаправлений на попередню в формі сторінку.

При переході на сторінку новини, користувач має можливість видалити небажану запис користувача, шляхом переходу за посиланням Видалити, поруч із залишеним коментарем. Після видалення коментаря модератор буде перенаправлено на сторінку з останніми новинами (рис.3.14).

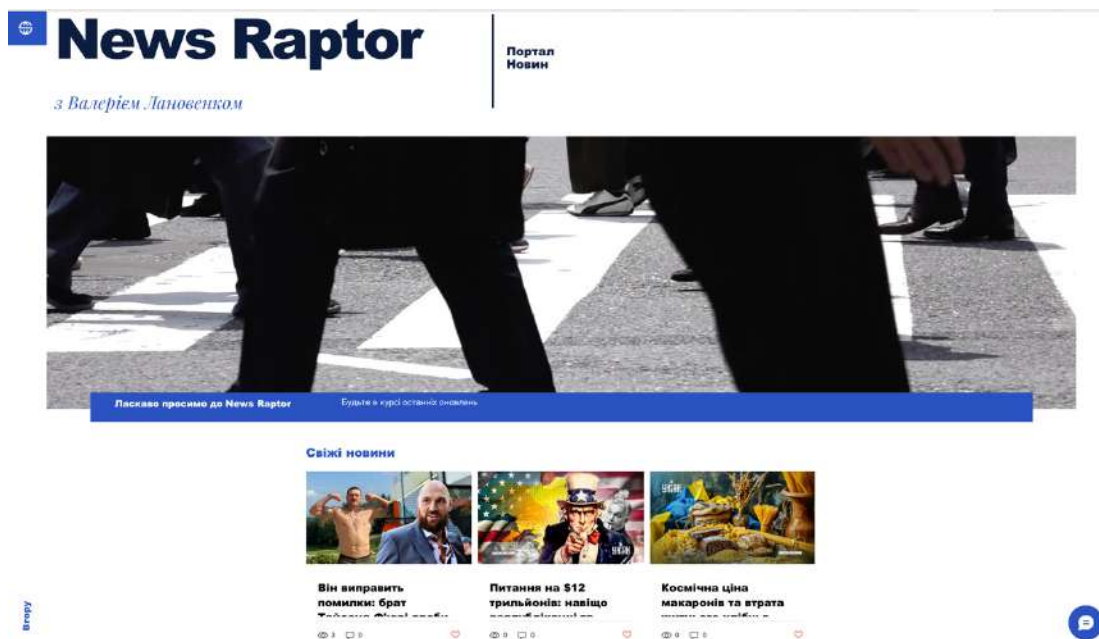


Рис. 3.14. Головна сторінка

3. Висновки по розділу 3

Розроблено новинний портал News Raptor. Враховуючи з проведеного аналізу існуючих систем, при проектуванні веб-порталу створено структуру порталу, який надає можливість перегляду зразу 10 новин, підписатись на RSS-канал або експортувати у PDF дозволені новини. Для доступу до розширених опцій таких як фільтр по категорії або приміткам, пошук і т.ін. необхідно зареєструватись. Описано структуру та етапи розробки новинного порталу.

Розглянуто реалізацію бази даних порталу, його логічну структуру та структури таблиць.

XIV. ВИСНОВКИ

Метою дипломної роботи була розробка новинного порталу News Raptor.

На початку роботи було обґрунтовано доцільність розробки новинного порталу, яка полягає в залученні великої аудиторії на власному майданчику, для подальшого розміщення рекламних блоків, як отримання прибутку від діяльності. На даний момент існує діюча група в одній із соціальних мереж, що надає обмежений доступ, і не дає можливості виставлення рекламних повідомлень.

В аналітичному огляді літератури зі створення новинних порталів розглянуті основні підходи, технології та стандарти, які використовуються як в рамках інфраструктури порталу, так і при реалізації його функціональних модулів.

Основними цілями розробки новинного порталу стали:

- концентрація і систематизація інформації новин;
- своєчасне забезпечення всіх користувачів, в тому числі і мобільних, актуальною і достовірною інформацією (новинами);
- створення новинного порталу для презентації інформації для ефективного використання різномірних і часто хаотичних новин і даних;
- створення окремого новинного порталу, для збільшення кількості переглядів;
- інтенсифікація внутрішнього обміну думок (коментарів) по новинах;
- створення умов для розміщення рекламних банерів, з метою отримання прибутку від діяльності.

У роботі представлено функціональне призначення порталу, що полягає в наданні можливості відвідувачеві здійснювати: навігацію по новинному порталу; переглядання статей і новин, як в стоці, так і за категоріями; залишати коментарі. Так само на порталі передбачено блок для розміщення банерів (показу комерційної реклами).

Визначено вимоги до розмітки, рядкових елементів новинного порталу. Побудовано функціональну структуру новинного порталу. Визначено вимоги до програмного, технічного та інформаційного забезпечення. Описано перспективність новинного порталу і можливості його розвитку.

XV. СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кренш, М., Робінс, Дж. (2019). Веб-дизайн для початківців: Керівництво з HTML, CSS, JavaScript та веб-графіки. Видавництво O'Reilly.
2. Ньюман, С. (2015). Побудова мікросервісів: Проектування високодеталізованих систем. Видавництво O'Reilly.
3. Андорецкая, И.В. Интернет программирование. - СПб.: BHV, - 2000. - 224 с.
4. Буковецкая, О.А. Дизайн текста: шрифт, эффекты, цвет. - М.: ДМК, 2012. - 304 с.
5. Васильев, И.А. Методы и инструментальные средства построения семантических web-порталов: диссертация на соискание ученой степени кандидата технических наук: спец. 05.13.11 / И. А. Васильев; Томский политехнический университет; науч. рук. В. З. Ямпольский. - Защищена 14.12.2005 г. - Томск: Б.и., 2005. - 185 с.
6. Васильев И. А. Выбор средства представления знаний для их использования в работе информационного портала организации // Материалы XLII Международной научной студенческой конференции «Студент и научно-технический прогресс»: Информационные технологии. - Новосибирск: Изд-во НГУ, 2004. - С. 56-60.
7. Веллинг, Л., Томсон, Л. Разработка веб-приложений с помощью PHP и MySQL, 4-е издание: перевод с англ. - М.: ООО «И.Д. Вильямс», 2011. - 848 с.
8. Вигерс К.И. Разработка требований к программному обеспечению, М, 2011.
9. Герасимов, В.В., Гридина Е.Г., Кривошеев Л.О., Курмышев Н.В., Попов С.В. Учебный курс «Технологии построения интернет-порталов».
10. Электронный ресурс. [URL: <http://ict.edu.ru/ft/005543/279-306.pdf>]
11. Зандстра, М. PHP: объекты, шаблоны и методики программирования, 2е издание: перевод с англ. - М.: ООО «И.Д. Вильямс», 2009. - 480 с.
12. Зотов, В.В. Терминологический словарь по автоматике, информатике и вычислительной технике / В.В. Зотов, Ю.Н. Маслов, А.Е. Пядочкин. 3-е издание, - М.: Высшая школа, 2009. - 191 с.
13. Макфаланд, Д. Большая книга CSS. 2-е издание. - СПб.: Питер, 2011. - 560 с.
14. Методические указания к выполнению организационно-экономической части дипломных проектов для студентов специальности «Программное обеспечение вычислительной техники и автоматизированных систем»
15. Надежин, А. В. Анализ размеров экранов Электронный ресурс.
16. Оплата за размещение баннера на своем сайте. Данные баннерной рекламы Rotaban. Электронный ресурс. [URL:
17. <http://www.brimz.ru/skolko-deneg-brat-za-razmeschenie-bannera-na-svoemsayte.html>]
18. Web-портали: призначення та основні характеристики Электронный ресурс. <http://easy-code.com.ua/2011/03/web-portali-priznachennya/>
19. Советов, Б.Я. Информационные технологии: учебник для вузов / Б.Я. Советов, В.В. Цехановский. - М.: Высшая школа, 2012.

- 20.Цеховой, В.А. Web-дизайн и коммерция. – М.: Наука и техника, 2013. - 192 с.
- 21.Шатунова, О.В. Информационные технологии: Учебное пособие / О.В. Шатунова. - Елабуга: Изд-во ЕГПУ, 2010. - 77 с.
- 22.Методические указания по выполнению преддипломной практики, специальность 230105.65 «Программное обеспечение вычислительной техники и автоматизированных систем». Составитель: Архипов Игорь Олегович, к.т.н., доцент, заведующий кафедрой «Программное обеспечение»
- 23.Савельева Н.В. Основы программирования на PHP [Электронный ресурс]: курс лекций. Учебное пособие для студентов вузов, обучающихся по специальностям в области информационных технологий/ Савельева Н.В.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2005.— 264 с.
- 24.Ларри Ульман Основы программирования на PHP [Электронный ресурс]/ Ларри Ульман— Электрон. текстовые данные.— М.: ДМК Пресс, 2007.— 286 с.
- 25.Ларри Ульман MySQL [Электронный ресурс]/ Ларри Ульман— Электрон. текстовые данные.— М.: ДМК Пресс, 2007.— 352 с.
- 26.Панфилов К.С. Создание веб-сайта от замысла до реализации [Электронный ресурс]/ Панфилов К.С.— Электрон. текстовые данные.—
- 27.М.: ДМК Пресс, 2009.— 440 с.
- 28.Клеппман, М. (2017). *Проектування дано-інтенсивних додатків: Основні ідеї для надійних, масштабованих та підтримуваних систем*. Видавництво O'Reilly.
- 29.Круг, С. (2014). *Не змушуйте мене думати: Практичний підхід до зручності використання веб-сайтів*. Видавництво New Riders.
- 30.Норман, Д. (2013). *Дизайн звичайних речей*. Видавництво Basic Books.
- 31.Галворсон, К., Рач, М. (2012). *Стратегія контенту для вебу*. Видавництво New Riders.
- 32.Кіссейн, Е. (2011). *Елементи стратегії контенту*. Видавництво A Book Apart.
- 33.Веселкова Т.В. Эффективная эксплуатация сайта [Электронный ресурс]: практическое пособие/ Веселкова Т.В., Кабанов А.С.— Электрон.
- 34.текстовые данные.— М.: Дашков и К, Ай Пи Эр Медиа, 2011.— 176 с.
- 35.Загуменнов А.П. Как раскрутить и разрекламировать Web-сайт в сети Интернет [Электронный ресурс]/ Загуменнов А.П.— Электрон. текстовые данные.— М.: ДМК Пресс, 2007.— 384 с.
- 36.Мациевский Н.С. Реактивные веб-сайты. Клиентская оптимизация в алгоритмах и примерах [Электронный ресурс]: учебное пособие/ Мациевский Н.С., Степанищев Е.В., Кондратенко Г.И.— Электрон. текстовые данные.— М.: БИНОМ. Лаборатория знаний, Интернет-Университет Информационных Технологий (ИНТУИТ), 2010.— 336 с.

- 37.Молочков В.П. Основы работы в Adobe Photoshop CS5 [Электронный ресурс]/ Молочков В.П.— Электрон. текстовые данные.— М.: ИнтернетУниверситет Информационных Технологий (ИНТУИТ), 2016.— 261 с.
- 38.Бердышев С.Н. Искусство оформления сайта. 2-е изд. [Электронный ресурс]: практическое пособие/ Бердышев С.Н.— Электрон. текстовые данные.— М.: Дашков и К, Ай Пи Эр Медиа, 2012.— 101 с.
- 39.МакДональд, М. (2020). *Веб-безпека для розробників: Реальні загрози, практичний захист*. Видавництво No Starch Press.
- 40.Грігорік, І. (2013). *Високопродуктивні браузерні мережі: Що повинен знати кожен веб-розробник про мережі та продуктивність вебу*. Видавництво O'Reilly.
- 41.Хавербеке, М. (2018). *Елоквентний JavaScript: Сучасний вступ до програмування*. Видавництво No Starch Press.
- 42.Сімпсон, К. (2014-2015). *Ти не знаєш JS* (серія книг). Видавництво O'Reilly.

XVI. ДОДАТКИ

Додаток А. Код account.php

Ось приклад коду для account.php, який може бути використаний для управління користувацьким акаунтом на веб-порталі. Цей код включає основні функції, такі як реєстрація, вхід, редагування профілю та вихід.

```
<?php
session_start();
require 'db_connection.php'; // Підключення до бази даних
require 'functions.php'; // Файл з додатковими функціями

// Перевірка дії (login, register, update, logout)
$action = isset($_GET['action']) ? $_GET['action'] : 'login';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    switch ($action) {
        case 'register':
            $username = $_POST['username'];
            $email = $_POST['email'];
            $password = password_hash($_POST['password'], PASSWORD_DEFAULT);
            $query = "INSERT INTO users (username, email, password) VALUES
            (?, ?, ?)";

            $stmt = $conn->prepare($query);
            $stmt->bind_param('sss', $username, $email, $password);
            if ($stmt->execute()) {
                header('Location: account.php?action=login');
            } else {
                echo "Помилка реєстрації!";
            }
            break;

        case 'login':
            $email = $_POST['email'];
            $password = $_POST['password'];
            $query = "SELECT * FROM users WHERE email = ?";
            $stmt = $conn->prepare($query);
            $stmt->bind_param('s', $email);
            $stmt->execute();
            $result = $stmt->get_result();
```

```

        $user = $result->fetch_assoc();
        if ($user && password_verify($password, $user['password'])) {
            $_SESSION['user_id'] = $user['id'];
            header('Location: profile.php');
        } else {
            echo "Невірний email або пароль!";
        }
        break;

    case 'update':
        $username = $_POST['username'];
        $email = $_POST['email'];
        $user_id = $_SESSION['user_id'];
        $query = "UPDATE users SET username = ?, email = ? WHERE id = ?";
        $stmt = $conn->prepare($query);
        $stmt->bind_param('ssi', $username, $email, $user_id);
        if ($stmt->execute()) {
            header('Location: profile.php');
        } else {
            echo "Помилка оновлення!";
        }
        break;
    }
} elseif ($action == 'logout') {
    session_destroy();
    header('Location: account.php?action=login');
}
?>

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Користувацький акаунт</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <?php if ($action == 'register'): ?>
        <h2>Реєстрація</h2>

```

```

        <form action="account.php?action=register" method="POST">
            <input type="text" name="username" placeholder="Ім'я користувача"
required>
            <input type="email" name="email" placeholder="Email" required>
            <input type="password" name="password" placeholder="Пароль"
required>
            <button type="submit">Зареєструватися</button>
        </form>
        <p>Вже є акаунт? <a href="account.php?action=login">Увійти</a></p>

<?php elseif ($action == 'login'): ?>
    <h2>Вхід</h2>
    <form action="account.php?action=login" method="POST">
        <input type="email" name="email" placeholder="Email" required>
        <input type="password" name="password" placeholder="Пароль"
required>
        <button type="submit">Увійти</button>
    </form>
    <p>Немає акаунту?
href="account.php?action=register">Зареєструватися</a></p>

<?php elseif ($action == 'update' && isset($_SESSION['user_id'])): ?>
    <h2>Редагувати профіль</h2>
    <?php
    $user_id = $_SESSION['user_id'];
    $query = "SELECT * FROM users WHERE id = ?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param('i', $user_id);
    $stmt->execute();
    $result = $stmt->get_result();
    $user = $result->fetch_assoc();
    ?>
    <form action="account.php?action=update" method="POST">
        <input type="text" name="username" placeholder="Ім'я користувача"
value="<?php echo $user['username']; ?>" required>
        <input type="email" name="email" placeholder="Email" value="<?php
echo $user['email']; ?>" required>
        <button type="submit">Оновити</button>
    </form>
    <p><a href="account.php?action=logout">Вийти</a></p>

```



```

        <?php else: ?>
            <p>Будь ласка, <a href="account.php?action=login">увійдіть</a> або <a
href="account.php?action=register">zareestruytesya</a>.</p>
        <?php endif; ?>
    </body>
</html>

```

1. Пояснення коду

1. **Підключення до бази даних:** `require 'db_connection.php';` та `require 'functions.php';` - підключення файлів для роботи з базою даних та додаткових функцій.
2. **Сесії:** `session_start();` - запуск сесії для зберігання інформації про користувача.
3. **Перевірка дії:** `isset($_GET['action']) ? $_GET['action'] : 'login';` - визначення дії (реєстрація, вхід, редагування профілю або вихід).
4. **Обробка POST-запитів:** залежно від дії, виконується відповідна функція (реєстрація, вхід, оновлення даних).
5. **Форми:** різні HTML-форми для реєстрації, входу та редагування профілю користувача.
6. **Редагування профілю:** завантаження даних користувача з бази для відображення в формі редагування.

2. Додаткові файли

- **db_connection.php:** файл для підключення до бази даних.
- **functions.php:** файл з додатковими функціями, наприклад, для перевірки вводу даних, захисту від SQL-ін'єкцій тощо.

Цей код є базовим прикладом і може бути розширений додатковими функціями та безпековими заходами.

Додаток Б. Код post.php

Ось приклад коду для post.php, який можна використовувати для управління створенням, редагуванням та видаленням постів на вашому веб-порталі. Цей код включає основні функції, такі як створення нового поста, редагування існуючого поста та видалення поста.

```
<?php
session_start();
require 'db_connection.php'; // Підключення до бази даних
require 'functions.php'; // Файл з додатковими функціями

// Перевірка дії (create, edit, delete)
$action = isset($_GET['action']) ? $_GET['action'] : 'view';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    switch ($action) {
        case 'create':
            $title = $_POST['title'];
            $content = $_POST['content'];
            $author_id = $_SESSION['user_id'];
            $query = "INSERT INTO posts (title, content, author_id) VALUES (?,
?, ?)";

            $stmt = $conn->prepare($query);
            $stmt->bind_param('ssi', $title, $content, $author_id);
            if ($stmt->execute()) {
                header('Location: post.php');
            } else {
                echo "Помилка створення поста!";
            }
            break;

        case 'edit':
            $post_id = $_POST['post_id'];
            $title = $_POST['title'];
            $content = $_POST['content'];
            $query = "UPDATE posts SET title = ?, content = ? WHERE id = ?";
            $stmt = $conn->prepare($query);
            $stmt->bind_param('ssi', $title, $content, $post_id);
```

```

        if ($stmt->execute()) {
            header('Location: post.php');
        } else {
            echo "Помилка редагування поста!";
        }
        break;

    case 'delete':
        $post_id = $_POST['post_id'];
        $query = "DELETE FROM posts WHERE id = ?";
        $stmt = $conn->prepare($query);
        $stmt->bind_param('i', $post_id);
        if ($stmt->execute()) {
            header('Location: post.php');
        } else {
            echo "Помилка видалення поста!";
        }
        break;
    }
} elseif ($action == 'view' && isset($_GET['id'])) {
    $post_id = $_GET['id'];
    $query = "SELECT * FROM posts WHERE id = ?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param('i', $post_id);
    $stmt->execute();
    $result = $stmt->get_result();
    $post = $result->fetch_assoc();
}
?>

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Пост</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>

```

```

<?php if ($action == 'create'): ?>
    <h2>Створити новий пост</h2>
    <form action="post.php?action=create" method="POST">
        <input type="text" name="title" placeholder="Заголовок" required>
        <textarea name="content" placeholder="Вміст поста"
required></textarea>
        <button type="submit">Створити</button>
    </form>

<?php elseif ($action == 'edit' && isset($_GET['id'])): ?>
    <h2>Редагувати пост</h2>
    <form action="post.php?action=edit" method="POST">
        <input type="hidden" name="post_id" value="<?php echo $post['id'];
?>">
        <input type="text" name="title" placeholder="Заголовок"
value="<?php echo $post['title']; ?>" required>
        <textarea name="content" placeholder="Вміст поста" required><?php
echo $post['content']; ?></textarea>
        <button type="submit">Оновити</button>
    </form>

<?php elseif ($action == 'view' && isset($_GET['id'])): ?>
    <h2><?php echo $post['title']; ?></h2>
    <p><?php echo $post['content']; ?></p>
    <p><a href="post.php?action=edit&id=<?php echo $post['id'];
?>">Редагувати</a></p>
    <form action="post.php?action=delete" method="POST" onsubmit="return
confirm('Ви впевнені, що хочете видалити цей пост?');">
        <input type="hidden" name="post_id" value="<?php echo $post['id'];
?>">
        <button type="submit">Видалити</button>
    </form>

<?php else: ?>
    <h2>Пости</h2>
    <?php
$query = "SELECT * FROM posts ORDER BY created_at DESC";
$result = $conn->query($query);
while ($post = $result->fetch_assoc()) {
    echo
    <h3><a href='post.php?action=view&id={$post['id']}'>{$post['title']}</a></h3>";
}

```

```

        echo "<p>" . substr($post['content'], 0, 100) . "...</p>";
    }
    ?>
    <p><a href="post.php?action=create">Створити новий пост</a></p>
<?php endif; ?>
</body>
</html>

```

1. Пояснення коду

1. **Підключення до бази даних:** `require 'db_connection.php';` та `require 'functions.php';` - підключення файлів для роботи з базою даних та додаткових функцій.
2. **Сесії:** `session_start();` - запуск сесії для зберігання інформації про користувача.
3. **Перевірка дії:** `isset($_GET['action']) ? $_GET['action'] : 'view';` - визначення дії (створення, редагування, видалення або перегляд поста).
4. **Обробка POST-запитів:** залежно від дії, виконується відповідна функція (створення нового поста, редагування існуючого поста, видалення поста).
5. **Форми:** різні HTML-форми для створення, редагування та видалення постів.
6. **Перегляд постів:** відображення списку всіх постів, кожен з яких можна переглянути детально або відредагувати/видалити.

2. Додаткові файли

- **db_connection.php:** файл для підключення до бази даних.
- **functions.php:** файл з додатковими функціями, наприклад, для перевірки вводу даних, захисту від SQL-ін'єкцій тощо.

Цей код є базовим прикладом і може бути розширений додатковими функціями та безпековими заходами.