

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
Фізико-математичний факультет
Кафедра інформатики та прикладної математики

«Допущено до захисту»

В.о. завідувача кафедри

_____ Семеріков С. О.
_____ 2025р.

Реєстраційний № _____ «___»
«___» _____ 2025 р.

РОЗРОБКА САЙТУ-ВІЗИТКИ ДЛЯ АБІТУРІЄНТІВ

Кваліфікаційна робота студента групи І-21
фізико-математичного факультету
спеціальності 014.09 Середня освіта
(Інформатика)
спеціалізації: програмування

Гури Станіслава Андрійовича

Керівник: старший викладач кафедри
інформатики та прикладної математики,
доктор філософії з освітніх,
педагогічних наук

Алека Галина Ігорівна

Оцінка:

Національна шкала _____

Шкала ECTS _____ Кількість балів _____

Члени комісії

_____	_____
(підпис)	(прізвище та ініціали)
_____	_____
(підпис)	(прізвище та ініціали)
_____	_____
(підпис)	(прізвище та ініціали)

ЗАПЕВНЕННЯ

Я, Гура Станіслав Андрійович, розумію і підтримую політику Криворізького державного педагогічного університету з академічної доброчесності. Запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Я не надавав і не одержував недозволену допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають покликання на відповідне джерело. Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Криворізького державного педагогічного університету ознайомлений. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.



ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1.....	
АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ РОЗРОБКИ САЙТІВ	6
1.1. Загальний огляд систем керування вмістом	6
1.2. Порівняння СКВ WordPress із іншими системами керування вмістом	15
1.3. Особливості використання WordPress для створення сайту-візитки .	18
Висновки до розділу 1	22
РОЗДІЛ 2.....	
ПРОЄКТУВАННЯ ТА РОЗРОБКА САЙТУ-ВІЗИТКИ ДЛЯ АБІТУРІЄНТІВ	24
2.1. Формування вимог до функціоналу та дизайну сайту. Створення структури сайту та макету інтерфейсу.....	24
2.2. Реалізація сайту-візитки на основі СКВ WordPress.....	28
2.3. Реалізація власних плагінів для розширення функціональності сайту-візитки.....	36
Висновки до розділу 2.....	59
ВИСНОВКИ	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	63

ВСТУП

Інформаційно-комунікаційні технології (ІКТ) відіграють дуже важливу роль у всіх сферах життя, а сучасний освітній процес з кожним днем все частіше переходить на дистанційну та змішану форми навчання. В умовах цифровізації одним із повсякденних понять є «веб-сайт» або просто «сайт», із яким користувачі взаємодіють щоденно завдяки смартфонам, персональним комп'ютерам та іншим пристроям, що мають вихід у всесвітню мережу Інтернет. Веб-сайти надають платформу для різних ІКТ-сервісів, що підкреслює їх значущість у сучасному інформаційному просторі.

Наявність офіційного сайту закладу освіти є невід'ємною складовою сучасного цифрового суспільства. Для майбутнього абітурієнта такий сайт слугує джерелом достовірної інформації про навчальний заклад, його факультети та спеціальності, терміни подачі документів, терміни складання вступних іспитів тощо. При цьому він може скористатися своїм смартфоном, планшетом, ноутбуком або будь-яким іншим сучасним гаджетом, що має доступ до мережі Інтернет та його веб-ресурсам. Проте, звичайний сайт може мати складну структуру, містити багато інформації й бути не дуже зручним у використанні. У цьому контексті доцільним є створення спеціалізованого ресурсу – сайту-візитки, орієнтованого безпосередньо на абітурієнтів. Тому, створення такого сайту для нашого університету потребує ретельного аналізу функціональних можливостей, з урахуванням вимог до актуальності, зручності навігації та представлення контенту.

Об'єкт дослідження – процес створення інформаційних веб-сайтів, зокрема сайтів-візиток.

Предмет дослідження – технології розробки веб-сайтів із використанням системи керування вмістом WordPress для створення сайту-візитки.

Мета дослідження – розробити інформаційний веб ресурс – сайт-візитку для абітурієнтів спеціальності Середня освіта (Інформатика) на основі

СКВ WordPress, враховуючи сучасні технології веброзробки, вимоги до дизайну та функціональності.

Для досягнення поставленої мети сформульовано такі **завдання дослідження**:

1. Проаналізувати сучасні платформи для створення веб-сайтів.
2. Здійснити порівняльний аналіз СКВ WordPress з іншими системами керування вмістом для виявлення її ефективності та доцільності використання у створенні сайтів-візиток.
3. Вивчити особливості використання WordPress для створення сайту-візитки, зокрема специфіку налаштувань, вибору шаблонів та плагінів.
4. Визначити вимоги до функціональності та дизайну сайту-візитки для абітурієнтів.
5. Реалізувати сайт-візитку на основі системи керування вмістом WordPress із застосуванням вбудованих інструментів і тем.
6. Розробити та інтегрувати власні плагіни з метою розширення базової функціональності сайту-візитки.

Практичне значення дослідження полягає у розробці та впровадженні сайту-візитки, призначеного для організації та супроводу процесу вступу абітурієнтів.

Структура роботи. Робота складається зі вступу, двох розділів, висновків, списку використаних джерел, що складає 24 найменування.

РОЗДІЛ 1

АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ РОЗРОБКИ САЙТІВ

1.1. Загальний огляд систем керування вмістом

Система керування вмістом (СКВ) – це програмне забезпечення, яке допомагає користувачам створювати, керувати, зберігати та змінювати свій цифровий контент. Це універсальна система для зберігання контенту – додатків, зображень і веб-сайтів – у зручному інтерфейсі, який можна налаштувати відповідно до потреб організації та її співробітників [8; 15, с. 36].

Вміст (контент) – термін з професійної термінології, його пов’язують з розробкою інформаційних веб-систем. Вміст веб-сайту – це інформаційне наповнення, що надається користувачам сайту. Він може бути у вигляді тексту, гіпертекстового посилання, зображення, відео, аудіо, анімації тощо [5; 9, с. 35].

Для розуміння принципів функціонування системи керування вмістом, доцільно розглянути традиційний підхід до розробки веб-сайтів. Створення та адміністрування веб ресурсу у ручному режимі передбачає написання статичних HTML-файлів з нуля та їхнє подальше завантаження на сервер для кожної окремої веб-сторінки.

Ефективною альтернативою цьому підходу є використання систем керування вмістом. Такі системи зазвичай вже мають реалізований бекенд та фронтенд, а користувач взаємодіє лише з інтуїтивно зрозумілим інтерфейсом, який дозволяє швидко й зручно редагувати вміст сайту. Основною метою впровадження систем керування вмістом є спрощення процесу оновлення контенту та підвищення якості користувацького досвіду [9, с. 349; 22].

Структура СКВ визначається завданнями, які можуть бути вирішеними системою для керування вмістом. СКВ складається з таких компонентів (рис. 1.1): модуль редагування (адміністративна частина), що забезпечує виконання адміністративних функцій по редагуванню контенту сайту; контент у структурованому вигляді, що, як правило, включає базу даних для зберігання

HTML-тексту веб-сторінок та сукупність упорядкованих медіа-файлів; модуль подання, ядро системи, що керує логікою подання контенту відвідувачам та забезпечує навігацію по контенту [5; 21, с. 194-195].

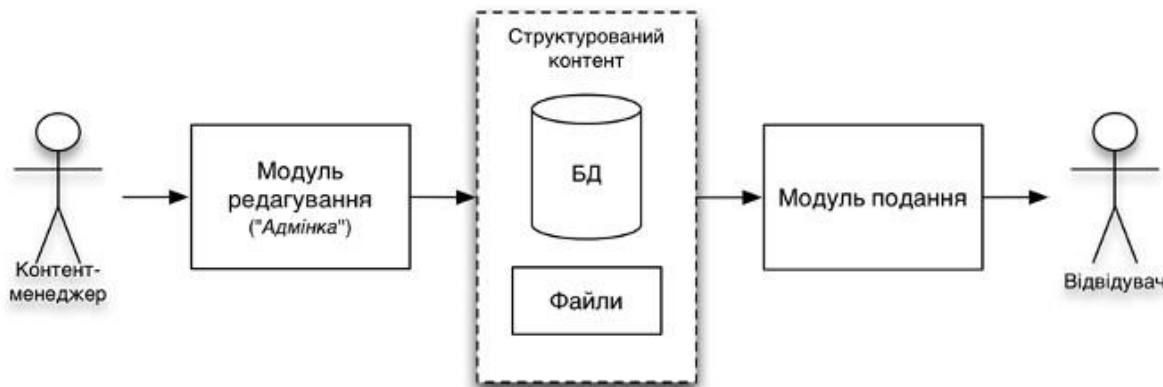


Рис. 1.1. Архітектура СКВ

Системи керування вмістом (СКВ) класифікуються на чотири основні типи залежно від підходів до їх створення та функціонування: конструктори, коробкові системи, фреймворки та самописні платформи [3]:

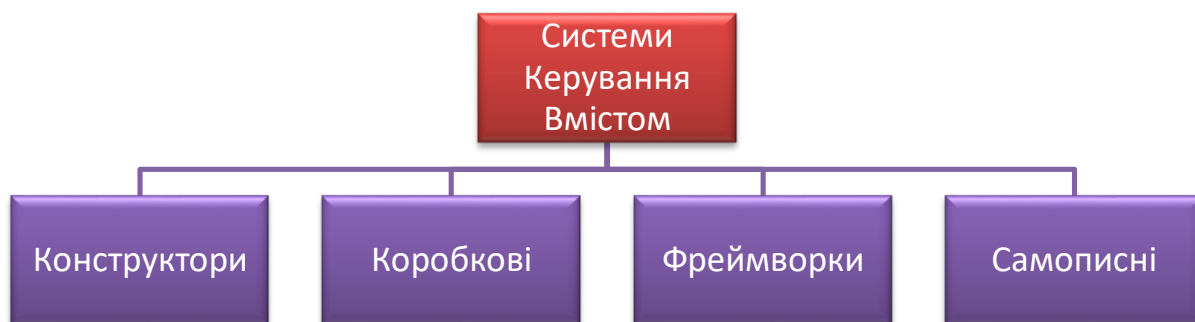


Рис. 1.2. Види СКВ

Конструктори – найкраще рішення для створення й керування сайтом. Їхнє застосування не потребує глибоких знань у сфері програмування, що робить їх зручними для користувачів без технічного досвіду. Основним недоліком більшості конструкторів є відсутність доступу до коду, а також обмежені можливості щодо редагування та кастомізації веб ресурсу.

Унаслідок цього сайти, створені на базі конструкторів, зазвичай мають обмежений потенціал у контексті пошукової оптимізації [3].

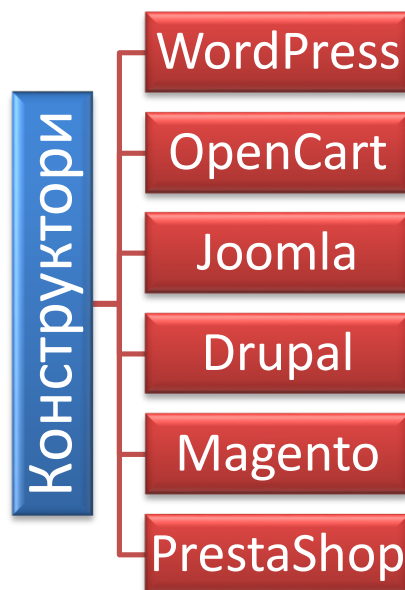


Рис.1.3. Конструктори

WordPress – це відкрита система керування вмістом (Content Management System), спочатку створена для ведення блогів, але за останні 10 років еволюціонувала в універсальну платформу для сайтів будь-якого типу [2; 14, с. 5].

На відміну від інших популярних систем керування вмістом (таких як Joomla чи Drupal), WordPress вирізняється акцентом на інтуїтивно зрозумілий інтерфейс користувача. Такий підхід сприяв швидкому зростанню довіри до платформи як розробників, так і представників малого та середнього бізнесу [2].

Однією з ключових переваг системи WordPress є її доступність і простота використання. Багато хостинг-провайдерів дозволяють встановити СКВ в один клік. Опублікувати новий контент не складніше, ніж друкувати щось в програмі Word. Про роботу з WordPress написано багато статей, знято детальні відео, опубліковано тисячі книг. На будь-яке питання щодо WordPress можна знайти відповідь. Маса безкоштовних можливостей: гнучкі теми для створення унікального дизайну, плагіни для розширення функціоналу – все це

можна знайти, не виходячи з адміністративної панелі сайту. Система постійно вдосконалюється завдяки зусиллям як професійних розробників, так і широкої спільноти волонтерів-ентузіастів [3; 14, с. 6-7].

Попри численні переваги, WordPress має й низку обмежень, які варто враховувати під час вибору цієї системи керування вмістом. Порівняно обмежені можливості налаштувань – це зроблено спеціально, щоб не заплутувати користувачів-початківців. Але просунуті користувачі не завжди можуть реалізувати якусь витончену задумку. Сайт на WordPress легко створити, але так само легко і зіпсувати. Значна частина відео-уроків і довідників з WordPress записана людьми, які дуже далекі від веб-розробки. Наприклад, вони радять як вносити правки в код, але не попереджають (бо не знають), що всі правки зникнуть після поновлення ядра СКВ або теми. «Движок» постійно намагаються зламати. У самій СКВ WordPress практично ніколи не було серйозних дірок в безпеці (окрім кількох невдалих релізів), головна загроза для користувача – тисячі плагінів «движка». Саме через плагіни в більшості випадків зловмисники зламують сайти на WordPress [3].

Отже, WordPress – це універсальна СКВ, яка підходить як для малого бізнесу і стартапів, так і для великих компаній, яким важливі гнучкість, швидкість запуску, інтеграції з локальними сервісами та прозора аналітика [2].

Joomla – система керування вмістом (СКВ), що має багато нагород, яка призначена як для побудови веб-сайтів, так і для створення потужних онлайн додатків. Багато аспектів, включаючи легкість використання та розширюваність, робить Joomla найпопулярнішим програмним забезпеченням для веб-сайтів. Найкраще те, що Joomla – відкритий ресурс, який доступний кожному [8, с. 2-3; 16].

Joomla використовують по всьому світу, щоб керувати веб-сайтами всіх форм та розмірів. Наприклад: корпоративні веб-сайти чи портали, корпоративні приватні мережі(Інтранет) та Екстранет, онлайн магазини, газети та публікації, електронна комерція та онлайн бронювання, державні додатки, сайти для малого бізнесу, некомерційні та комерційні веб-сайти,

громадські портали, шкільні та церковні веб-сайти, особисті та сімейні домашні сторінки [8, с. 3; 16].

Joomla розроблена так, щоб було легко її встановлювати і налаштовувати навіть недосвідченому користувачу. Багато сервісів веб-хостингу пропонує встановлення в один клік та отримання нового сайту, його запуск за лічені хвилини [8, с. 6-7; 16].

Багато компаній і організацій має вимоги, що перевищують можливості базового пакету Joomla. В цих випадках, могутній Joomla фреймворк для застосувань полегшує створення складних додатків, вони розширюють можливості Joomla практично в необмежених напрямках. Ядро фреймворку Joomla дозволяє розробникам швидко і легко створювати: інвентаризовані системи контролю, вікна звітів із даними, мости застосувань, каталоги продукції на замовлення, інтегровані системи електронної комерції, комплексні бізнес довідники, системи бронювання, засоби комунікації [8, с. 6 - 7; 16].

Головна перевага Joomla – існування зручних багатофункціональних програм для створення інтернет-магазинів різної складності: VirtueMart, JoomlaShopping, Tienda тощо. Створити повнофункціональний інтернет-магазин на Joomla простіше і швидше, ніж на WordPress. Сайт на Joomla можна оптимізувати без додаткових плагінів. Користувач може прописати мета-теги і налаштувати URL (адреси сторінок, які відображаються в рядку браузера). У движку відразу доступно кешування, що пришвидшує завантаження сторінок, підвищує позиції сайту. Сайт можна редагувати не заходячи в адміністративну панель. Доступно так зване фронтенд редагування контенту [3; 8, с. 5].

Joomla має такі ж самі недоліки, як і WordPress – надвелика кількість навчальної інформації, але серед неї є навіть даремні й шкідливі поради та уроки. Наступний недолік – певним чином обмежені можливості. Сюди ж можна віднести пошук слабких місць в коді й намагання зламувати сайти на цій СКВ. У порівнянні з WordPress, можна додати ще один недолік – значно менша кількість додатків, плагінів та тем для Joomla [3].

З того часу, як Joomla базується на PHP та MySQL, побудованими додатками можуть ділитись, а також їх можуть використовувати та підтримувати будь-хто [16].

Drupal – це платформа, яка вимагає від користувача наявності певних навичок. На відміну від Joomla та WordPress, Drupal – це СКВ, яка дозволяє втілити практично будь-який задум. В ній з однаковою успішністю створюються Інтернет-магазини, соцмережі, великі корпоративні сайти [3; 15, с. 1].

Drupal – це гнучка СКВ, заснована на стеку LAMP, з модульним дизайном, що дозволяє додавати та видаляти функції, встановлюючи та видаляючи модулі, а також змінювати зовнішній вигляд веб-сайту, встановлюючи та видаляючи теми. Базове завантаження Drupal, відоме як Drupal Core, містить PHP-скрипти, необхідні для запуску базової функціональності СКВ, кілька додаткових модулів і тем, а також багато JavaScript, CSS і зображень. Багато додаткових модулів і тем можна завантажити з веб-сайту Drupal.org [10; 15, с. 2-3].

Drupal також може працювати на інших технологічних стеках: операційною системою може бути Windows або Mac OS замість Linux, веб-сервером може бути Nginx або IIS замість Apache, база даних може бути PostgreSQL або SQLite замість MySQL, або сумісна з MySQL заміна, наприклад, MariaDB або Percona. Інші операційні системи, веб-сервери та бази даних також можуть працювати; однак скрипти, які використовує програма, написані на PHP, тому їх не можна змінити [10; 15, с. 2-3].

Однією з головних переваг Drupal є високий рівень безпеки, який не потребує складних налаштувань, чи встановлення додаткових модулів. Саме тому розробники вважають її найзахищенішою СКВ, що базується на PHP. Також до переваг можна віднести зручну оптимізацію без додаткових встановлень, а в стандартній комплектації СКВ доступні SEO-налаштування. На відміну від Joomla та WordPress, в СКВ Drupal права користувачів гнучко налаштовуються. Готові потужні й безкоштовні рішення в достатній кількості.

Для електронної комерції, наприклад, є такі платформи, як Ubercart. Вони перетворюють Drupal в зручний багатфункціональний інтернет-магазин і не вимагають складних налаштувань і знань програмування. Адаптація для мобільних пристроїв. Починаючи з 8-ї версії, всі теми для Drupal прекрасно відображаються на всіх популярних мобільних девайсах. Десятки і сотні видів контенту. Без додаткових правок коду можна створити такий тип контенту, як наприклад, опитування, статистичні дані, відео тощо [3; 15, с. 3-5].

До недоліків Drupal можна віднести велику вартість й рівень складності, що для непідготовленого користувача робить важким створення проекту на цій СКВ. Для професійного розробника існує достатній недолік – йому доведеться платити більше, ніж розробнику на інших СКВ. Вартість преміум-тем для Drupal на рівень дорожче, ніж для тем на СКВ WordPress. Окрім того, за швидкістю роботи представлена система значно повільніша, аніж розглянуті аналоги [3].

Коробкові СКВ, на відміну від самописних – це відомі на ринку повноцінні системи управління вмістом. Для розробки, просування або підтримки такого сайту не складно знайти досвідченого виконавця. Коробкові СКВ мають технічну документацію для розробників. Наявність готових модулів з можливістю їх доопрацювання та кастомізації теж викликає довіру в розробників [3].



Рис. 1.4. СКВ з коробки

СКВ на фреймворку – окремий тип движків. Такі СКВ також мають технічну документацію й для роботи з ними можна залучати сторонніх фахівців. СКВ на фреймворку мають найвищий рівень гнучкості для створення абсолютно будь-яких модулів для роботи сайту [3].

СКВ на фреймворку має дві основні функції: робота на серверній стороні (бекенд) і робота на клієнтській стороні (фронтенд). Із зовнішньою частиною програми пов'язані фронтенд-фреймворки, тобто вони відповідають за інтерфейс, а за внутрішню частину додатку, тобто логіку додатку відповідають бекенд-фреймворки [6, с. 40].

Серверні веб-фреймворки (СВФК) мають певні обмеження щодо створення веб додатків із розвиненим графічним інтерфейсом. Вони забезпечують створення простих сторінок і різних форм, маючи обмеження в своїй функціональності. Також, СВФК здатні відповідати за безпеку в разі атак й забезпечувати формування вихідних даних. Все це, безумовно, спрощує процес розробки. Серверні фреймворки, найчастіше, відповідають за окремі критично важливі частини програми, без яких неможливе стале функціонування. На Рис. 1.5 можна побачити найбільш поширених представників даного класу фреймворків з відповідними мовами програмування [6, с. 40]:

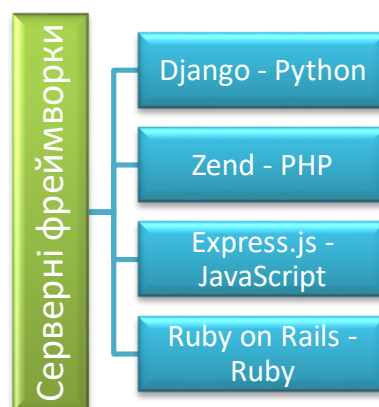


Рис. 1.5. Серверні фреймворки

Клієнтські веб-фреймворки (КВФК), на відміну від серверних, ніяк не пов'язані з логікою програми. Такий тип фреймворків призначений для роботи в браузері. Впровадження нових користувацьких інтерфейсів з їх

поліпшенням – це одна з можливостей КВФК. Створення різних анімацій й односторінкових додатків теж відноситься до можливостей фронтенд-фреймворків. Всі клієнтські фреймворки можна відрізнити за функціональністю та використанням. До найбільш поширених фреймворків такого типу виділимо наступні:

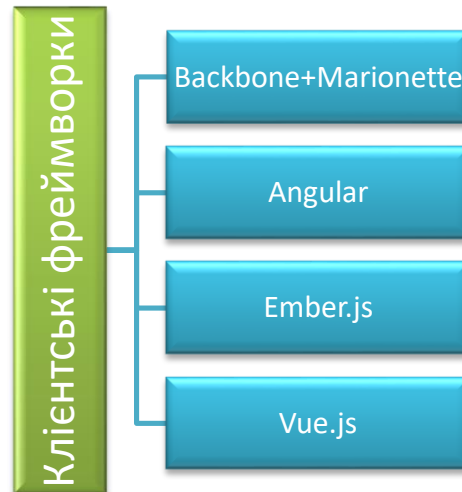


Рис. 1.6 Клієнтські фреймворки

Головна риса всіх цих фреймворків – це використання мови програмування – JavaScript [6, с.40].

Також, значного поширення набули багатofункціональні фреймворки. Їх застосовують для комплексного розв’язання завдань програмування. Meteor відноситься до фул-стек веб-фреймворків. Надзвичайно популярним його робить той факт, що він здатний задовольнити майже всі потреби як з боку клієнта, так і з боку сервера. Використовуючи такий фреймворк ми забезпечуємо заощадження часу. У якості прикладу можна навести заощадження часу на налагодження взаємодії між двома ВФК через інтерфейс RESTAPI. ВФК Meteor, при цьому, надає можливість забезпечення виконання відповідних процедур і прискорення процесу розробки. Існує також можливість створення і використання для них одного коду, оскільки обидві сторони – серверна та клієнтська – працюють на одній мові. Однією з особливостей даного ВФК є «режим реального часу» – можливість внесення змін в одному інтерфейсі, що обумовлює зміни і в інших. Як приклад,

додавання коментарів або зміна змісту документу чи таблиці із загальним доступом, коли інші користувачі одночасно будуть бачити ці зміни [6, с. 40].

1.2. Порівняння СКВ WordPress із іншими системами керування вмістом

Почнемо з того, що об'єднує всі СКВ. Три найпопулярніші СКВ в інтернеті мають багато спільного з точки зору технологій, філософії та спільноти. WordPress, Joomla та Drupal безкоштовні та мають відкритий вихідний код. Усі три движки написані переважно на PHP. Усі вони підтримують MySQL як систему керування базами даних. WordPress підтримує виключно MySQL, тоді як Joomla та Drupal підтримують додаткові системи керування базами даних. Усі три з них використовують теми та шаблони для швидкого оформлення сайтів, а також плагіни, модулі чи розширення для збільшення функцій [1].

Також, як програмне забезпечення з відкритим кодом, WordPress, Joomla та Drupal є проектами спільноти. Ці системи розробляє та підтримує велике товариство розробників та користувачів з усього світу. Люди роблять внесок у розвиток відповідних систем у вигляді нових функцій, плагінів, тем, виправлень помилок та загального розвитку проекту. Чим більша спільнота, тим легше працювати з движком. В спільноті користувачі обмінюються знаннями, вирішують проблеми та підтримують один одного, що сприяє подальшому розвитку та покращенню цих СКВ. Хоча WordPress, Joomla та Drupal мають багато подібностей, вони також і відрізняються в багатьох аспектах [1].

Joomla – одна з найпопулярніших СКВ для створення динамічних веб-сайтів. Joomla є сумісною з базами даних MySQL, SQL Server та PostgreSQL. Однією з ключових особливостей Joomla є те, що вона пропонує найширший спектр функціональних можливостей, таких як галереї зображень, форуми, чати, блоги, новини тощо. Аналогічно, Drupal також спрямований на створення динамічних веб-сайтів. Він має високу сумісність з різними базами

даних. Безпека, швидке завантаження та широке розмаїття користувацьких ролей є основними особливостями Drupal. Як приклад останньої характеристики, Drupal дозволяє обмежити доступ конкретного користувача таким чином, що він може змінювати лише властивості й лише певного функціоналу, і навіть змінювати лише певні параметри цього функціоналу [17, с. 3].

З іншого боку, WordPress спочатку був орієнтований на створення блогів, але еволюціонував до створення веб-додатків та рішень для електронної комерції. Однією з ключових особливостей WordPress є його чудове позиціонування для пошукової оптимізації (SEO). Причиною цього є те, що WordPress має багато плагінів для швидкого включення в пошукові системи, порівняно з іншими СКВ. Крім того, можна безкоштовно створити простий блог під субдоменом платформи (.wordpress.com). Розширення та модулі також можна використовувати в Joomla та Drupal для досягнення кращого SEO, але ці можливості не мають такого високого впливу, як WordPress. Всі платформи СКВ пропонують покращити SEO ручним способом [17, с. 3-4].

Існує багато відмінностей між трьома обраними СКВ. Є багато можливих сценаріїв, але, як правило, веб-сайти починаються зі зменшеного розміру і не включають занадто багато характеристик, модулів або контенту. З часом, зі збільшенням кількості відвідувачів сайту, нового розробленого контенту, обсягу продажів тощо, виникає потреба у додаванні нових модулів та розширень на сайт, що надають йому більше можливостей. WordPress пропонує багато розширень для додавання певного функціоналу, але він не має такого широкого спектру функціональних можливостей, як Joomla. Крім того, WordPress передбачає можливість інтеграції додаткових розширень у середовище керування. Нарешті, Drupal є найскладнішою СКВ з точки зору розширення своїх можливостей, оскільки певний функціонал може мати залежності від різних бібліотек або функцій, що робить процес розширення довгим і виснажливим. Те ж саме стосується і функціоналу, де Joomla і

WordPress мають велику кількість функцій для розширення сайту. На відміну від них, ця можливість більш обмежена в реалізації Drupal. Репозиторії дуже важливі для оновлення СКВ, впровадження нових функцій та покращення існуючих. Ці репозиторії можуть бути розподіленими, як у випадку з Joomla і WordPress, або централізованими, як у випадку з Drupal. Найбільші спільноти користувачів належать до Joomla і WordPress, а отже, вони мають дуже активну спільноту користувачів і відмінну документацію. З одного боку, у випадку з Drupal спільнота користувачів є відносно обмеженою, що зумовлено підвищеною складністю впровадження та адміністрування цієї системи. Як наслідок, обсяг наявної документації також є менш повним порівняно з іншими системами керування контентом. У таблиці нижче наведено основні характеристики трьох обраних СКВ. Порівняння було здійснено на основі доступної офіційної документації та попередніх досліджень [17, с. 4].

Таблиця 1.1.

Якісне порівняння Joomla, WordPress та Drupal

Характеристика	Joomla	Drupal	WordPress
Основний тип контенту	Веб-сайти, онлайн-додатки	Блог	Блог, електронна комерція, онлайн-додатки
Доступність розширення	Висока	Середня	Висока
Діапазон функціональності	Високий	Середній	Високий
Репозиторій розширень	Розподілений	Централізований	Розподілений
Документація	Чудово	Добре	Чудово
Спільнота користувачів	Дуже активна	Обмежена	Дуже активна
Простота використання	Простий	Складний	Простий
Персоналізація ролі користувача	Середній	Дуже високий	Середній
Ручне SEO-позиціонування	Так	Так	Так
Автоматичне SEO-позиціонування	Розширення	Модулі	Плагіни та інструменти

Підсумовуючи результати порівняння, слід зазначити, що за п'ятьма ключовими характеристиками – доступністю розширень, діапазоном

функціональності, якістю документації, активністю користувацької спільноти та простотою використання – найоптимальнішою системою керування контентом є WordPress. Саме з огляду на вказані переваги платформу WordPress було обрано для реалізації сайту-візитки.

1.3. Особливості використання WordPress для створення сайту-візитки

В процесі створення сайту-візитки засобами СКВ WordPress можна виділити певні кроки.

Збір необхідної інформації для створення сайту-візитки на WordPress. Загальні відомості про організацію: час створення організації/початку роботи в заявленій сфері, місію та мету, основні напрямки діяльності. Контактні відомості: телефон, адреса, e-mail, групи або професійні профілі в соцмережах, П.І.Б. відповідальних осіб тощо [4].

Створення прототипу майбутньої «візитки» на WordPress. Прототип – це схема майбутнього сайту. Його можна створити в текстовому редакторі, програмі для створення презентацій або на папері. Прототип вважається готовим, якщо: зрозуміло, скільки всього сторінок буде на сайті, і які це сторінки (наприклад, головна, сторінка з новинами, сторінка портфоліо); зрозуміло, з яких блоків складається перша сторінка, в якій послідовності ці блоки розташовані; для кожного блоку написано текст і підібрано фото, зображення, якщо це необхідно; для інших сторінок складено такі самі схеми, як для головної; зрозумілі переходи між сторінками (наприклад, якщо на головній сторінці є блок із кількома відгуками та сторінка з всіма відгуками, тоді має бути перехід із блока) тощо [4].

Вибір імені для сайту й перевірка його на зайнятість. У кожного сайту є своя адреса в мережі. Для створення такої адреси необхідно доменне ім'я, наприклад, site.com. Після того, як сформулювали 2-3 варіанти доменного імені, перевіряємо, чи вільні вони. Для цього достатньо в пошуковій системі написати запит «перевірка доменного імені» й вибрати будь-який ресурс. За

результатами перевірки залишаємо одне доменне ім'я [4; 14, с. 11-13; 23, с. 86].

Вибір й покупка хостингу, реєстрація доменного імені. Усі сайти розташовуються в мережі на великих комп'ютерах – серверах. Для цього власник сайту купує місце на сервері – хостинг. Під час вибору хостингу обираємо його за критеріями: відгуки та рекомендації (особливо щодо роботи технічної підтримки), відповідними тарифами, можливістю зареєструвати доменне ім'я, наявністю автоматичного встановлення WordPress, наявністю автоматичного встановлення безкоштовного SSL-сертифіката, а також наявністю зручного інтерфейсу для керування акаунтом хостингу (як правило, C-Panel) [4; 14, с. 11-13; 23, с. 86].

Встановлення WordPress. Встановлюємо WordPress. Багато хостингових компаній пропонують хостинг WordPress з установкою за нас. Якщо ми хочемо використовувати WordPress, найпростіше рішення – обрати одну з цих хостингових компаній. Вибрана хостингова компанія проведе нас через процес встановлення, запропонувавши вибрати тему і будь-які функції плагінів, які ми хотіли би додати.

Увійдемо в систему. Після налаштування WordPress ми побачимо шаблон за замовчуванням, коли зайдемо на свій сайт.

На інформаційній панелі натиснемо зліва «Зовнішній вигляд». Натиснемо «Додати нову тему», щоб переглянути каталог тем. Ми можемо шукати за ключовим словом (наприклад, «музика»), а також за допомогою фільтра функцій вибрати тему вашого блогу, функції, які вона включає, і тип макета, який вона використовує. Уявімо, як виглядатиме тема з нашим власним вмістом, особливо з нашими фотографіями. Рекомендується зосередитися на адаптивних темах, які добре працюють на всіх пристроях.

Натиснемо на тему, щоб переглянути її. Коли ми знайдемо тему, яка нам подобається, натиснемо, щоб встановити її. Це додасть її до нашої СКВ WordPress, але не змінить зовнішній вигляд нашого сайту. Щоб почати використовувати тему, натиснемо зліва «Зовнішній вигляд». Ми можемо

побачити всі встановлені теми, включно з їхніми маленькими зображеннями. Активуємо одну з них, щоб почати використовувати її на своєму сайті.

Щоб налаштувати тему, натиснемо «Налаштувати» зліва у розділі «Зовнішній вигляд». Для різних шаблонів тут будуть показані різні варіанти. У розділі «Зовнішній вигляд» є посилання на меню, за допомогою якого ми можемо створити власне меню.

Натиснемо «Редактор файлів теми» (у розділі «Зовнішній вигляд» зліва), і ми зможемо редагувати шаблони CSS і PHP [4; 14, с. 13; 19, с. 33-35; 21, с. 196; 23, с. 84-85].

Додавання сторінок. Для додавання нової сторінки необхідно увійти до адміністративної панелі WordPress і в меню зліва вибрати розділ «Сторінки». За замовчуванням у системі вже наявний приклад сторінки, яку можна переглянути або відредагувати, натиснувши на її назву. Для створення нової сторінки слід натиснути кнопку «Додати нову».

Увійдемо до WordPress і натиснемо «Сторінки» зліва. За замовчуванням у нас є зразок сторінки. Тут ми можемо побачити будь-які інші сторінки, які ми додали, а також натиснути на назву сторінки, щоб відредагувати її. Натиснемо кнопку «Додати нову».

Введемо назву нашої сторінки в першому полі, де написано «Додати назву». Вона буде використовуватися як назва посилання на цю сторінку в навігаційній панелі, тому зробимо її короткою і змістовною.

Використовуємо редактор блоків WordPress, щоб додати вміст сторінки. Кожна частина нашого контенту (наприклад, абзац або зображення) є блоком. Якщо ми хочемо додати абзац, почнемо вводити його. В іншому випадку, натиснемо «/», щоб відкрити меню для додавання зображення, заголовка, списку або іншого популярного блоку. Ми також можемо натиснути знак «+» у верхньому лівому куті, щоб відкрити меню всіх блоків зліва. Коли ми натискаємо блок або виділяємо в ньому текст, параметри форматування з'являються у спливаючому меню [14, с. 36-37; 21, с. 198].

Додавання дописів. Дописи зазвичай використовуються для контенту, який ми плануємо регулярно оновлювати. Наприклад, кожна новина або новий запис у блозі буде постом. Окремі дописи не додаються на панель навігації, але дописи датуються, і відвідувачі можуть переглядати їх за місяцями або категоріями дописів. Існує багато спільного в тому, як ми керуємо сторінками і публікаціями, але є й деякі відмінності. Нижче наведено алгоритм додавання нового допису.

Увійдемо до WordPress і натиснемо «Публікації» зліва. Ми побачимо список дописів, і, як і у випадку зі сторінками, ми можемо редагувати допис, натиснувши на його назву. Натиснемо «Додати новий», щоб створити новий пост.

Додамо статтю так само, як ми це робимо для сторінок, створюючи її з блоків контенту.

Праворуч натиснемо «Дописати», і ми зможемо додати категорії для своїх дописів. Це важливий інструмент навігації, оскільки він організовує дописи в групи, які люди можуть легко переглядати. Допис може з'являтися в декількох категоріях. Для оптимального юзабіліті обмежимо свій сайт невеликим вибором категорій.

Додамо теги до своїх дописів праворуч. Це короткі описові ключові слова або фрази, які використовуються відвідувачами для пошуку пов'язаних дописів. Класифікація за тегами є гнучкішою, ніж категоризація. Наприклад, якщо публікація повністю присвячена викладацькому складу, її доцільно віднести до категорії «Викладачі». Водночас цій самій публікації можуть бути призначені теги, як-от «науковий ступінь», «досвід», «курси», навіть якщо ці аспекти лише побіжно згадуються в тексті.

Ми можемо зберігати чернетку під час роботи. Коли ми закінчимо, натиснемо кнопку «Попередній перегляд», щоб вичитати наш допис перед тим, як він буде опублікований на весь світ.

Натиснемо «Опублікувати», коли все буде готово. WordPress зробить його першим дописом на вашому сайті і автоматично додасть його до архіву, навігації за категоріями і тегами [4; 14, с. 40-42; 21, с. 199; 23, с. 90-91].

Керування коментарями. WordPress дозволяє приймати коментарі до ваших публікацій і сторінок, що може бути цінним способом отримання зворотного зв'язку від вашої аудиторії.

Коли ми ввійдемо в свою панель керування WordPress, ми побачимо скільки коментарів очікують схвалення на навігаційній панелі. Натиснемо «Коментарі», щоб переглянути їх.

Прочитаємо коментарі. При наведенні миші на коментар з'являються посилання для схвалення коментаря або переміщення його до папки «Спам» або «Кошик». Використовуємо ці посилання, щоб підтвердити або видалити коментарі, якщо у нас їх лише кілька. Ми також можемо редагувати коментарі, але це рідко використовується. Якщо ми редагуємо чийсь коментар, чітко вказуємо, що саме ми змінили, щоб не спотворити його зміст.

Якщо нам потрібно обробити багато коментарів, встановимо прапорець навпроти тих, які ми хочемо затвердити. Потім клацнемо меню «Групові дії» і оберемо «Затвердити». Натиснемо «Застосувати», щоб затвердити всі позначені коментарі. Використовуємо аналогічний процес (з іншою груповою дією), щоб перемістити групу коментарів до «Спаму» або «Кошика» [4; 14, с. 70-75; 21, с. 200; 23, с. 97].

Висновки до розділу 1

У першому розділі було здійснено аналіз сучасних систем керування вмістом (СКВ), що широко застосовуються у веброзробці. Встановлено, що СКВ є ключовим інструментом для розробки як простих інформаційних ресурсів, так і складних динамічних сайтів із широкою функціональністю.

У ході порівняльного дослідження СКВ WordPress, Joomla та Drupal було створено узагальнюючу таблицю з ключовими параметрами оцінювання, що охоплювала такі аспекти, як: простота інсталяції, гнучкість налаштування,

наявність розширень, якість технічної документації, активність спільноти користувачів, безпека, продуктивність та можливості кастомізації. За результатами аналізу встановлено, що WordPress демонструє перевагу за п'ятьма критично важливими характеристиками: доступність розширень, функціональне різноманіття, якість супровідної документації, активність користувацької спільноти та зручність користування. Це стало ключовим аргументом на користь вибору WordPress як базової платформи для реалізації проєкту сайту-візитки.

Детально розглянуто особливості використання WordPress у контексті створення сайтів-візиток. Проаналізовано механізми налаштування тем, підключення шаблонів та використання плагінів для розширення стандартного функціоналу. Підкреслено важливість правильного вибору теми оформлення, оскільки вона визначає загальну структуру, макет і типові елементи взаємодії користувача з сайтом. Також визначено, що застосування плагінів дозволяє без програмування інтегрувати додаткові компоненти, такі як форми зворотного зв'язку, галереї, слайдери тощо, що є особливо актуальним для сайтів, орієнтованих на зовнішню презентацію закладу.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА САЙТУ-ВІЗИТКИ ДЛЯ АБІТУРІЄНТІВ

2.1. Формування вимог до функціоналу та дизайну сайту. Створення структури сайту та макету інтерфейсу

Щоб створити сайт на WordPress, нам потрібно врахувати три ключові моменти: спосіб введення різних видів контенту; категорії, які потрібні для організації контенту; вигляд макету та дизайну.

Введення контенту. WordPress збирає фрагменти контенту в HTML-сторінки, і одним з найважливіших завдань процесу планування є розробка корисної і простої організації цих фрагментів. Інша важлива задача – вирішити, як розбити контент на найменші логічні частини. Пізніше легко зібрати ці шматки різними способами; розбивати шматки на частини пізніше, коли ми зрозуміємо, що вони занадто великі, буде дорого. Простий приклад: краще мати кожен лист як окремий шматок, ніж мати один шматок з усіх листів. Ураховуючи зазначене, розглянемо базову карту структури сайту, подану на рисунку нижче [13, с. 9; 21, с. 28-30].

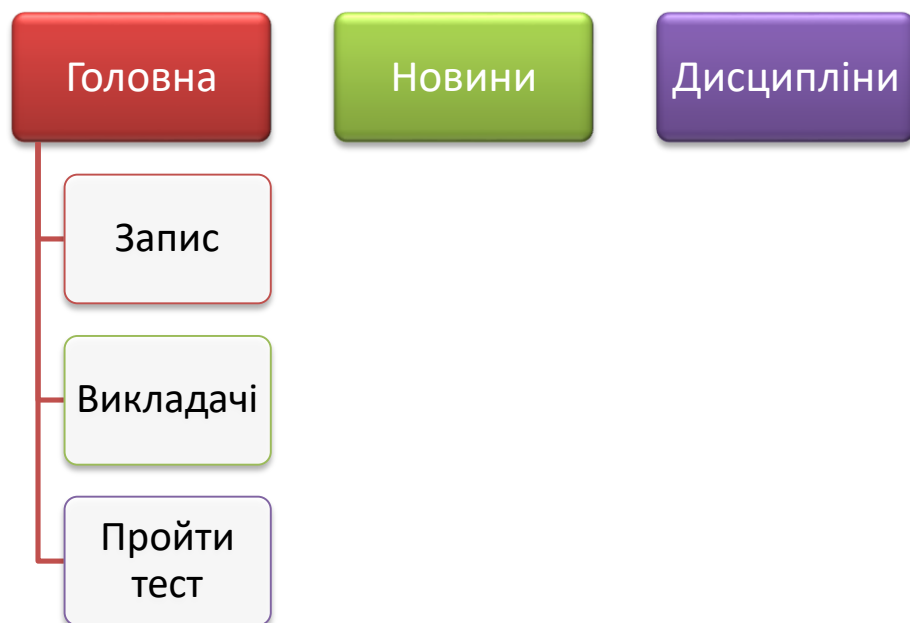


Рис. 2.1. Карта сайту

На першому рівні структури сайту представлено три основні сторінки, а також підсторінки, що відповідають трьом підрозділам сторінки «Головна». Інші елементи вебресурсу реалізовано у вигляді окремих HTML-сторінок, які об'єднані загальним навігаційним меню та гіперпосиланнями [13, с. 10; 21, с. 28-30].

Слід зазначити, що на Рис. 2.1 представлено плановану організацію структури сайту з точки зору його візуального та навігаційного представлення під час перегляду користувачем. Водночас ця структура не обов'язково збігається з ієрархією елементів у системі управління вмістом WordPress. Проте, використовуючи функціональні можливості зазначеної СКВ, зокрема створення сторінок і підсторінок, можливо реалізувати структуру, ідентичну до тієї, що відображена у плані [13, с. 10; 21, с. 28-30].

Інший спосіб подумати про цей процес – подивитися на контент і запитати, чи можна його використовувати в різні способи на всьому сайті [13, с. 10; 21, с. 28-30].

Введемо наші три сторінки «Головна», «Новини» та «Дисципліни» як окремі сторінки WordPress, а допоміжну інформацію – як окремі фрейми сторінки «Головна» [13, с. 10; 21, с. 30-31].

Повертаючись до карти сайту, ось список того, як ми збираємося вводити різні типи контенту в WordPress [13, с. 11; 21, с. 30-31]:

- «Головна» – односторінковий;
- «Новини» – односторінковий;
- «Дисципліни» – односторінковий;
- «Запис» – окремий фрейм;
- «Викладачі» – окремий фрейм;
- «Пройти тест» – окремий фрейм.

Організація категорій. Тепер настав час подумати про структуру категорій для постів і про те, як ця структура буде пов'язана з нашими основними типами фрагментів контенту: записи, викладачі, тест [13, с. 11-12; 21, с. 41-42].

Застосування фреймового підходу дає змогу не лише ефективно реалізувати головну сторінку сайту, але й просто пов'язати всі фрейми з цією сторінкою. Нехай це буде просто; нехай все буде легкодоступним – ось девіз цього процесу планування. Відтак, доцільно перейти до структури категорій, заснованої на фреймах, а не окремих підсторінках [13, с. 11-12; 21, с. 41-42].

Вигляд майбутнього сайту. Частиною процесу планування будь-якого сайту є визначення макета сторінок і того, як вони будуть виглядати. WordPress збирає і доставляє HTML-сторінки за допомогою групи файлів шаблонів, які називаються темами, тому для досягнення бажаного зовнішнього вигляду сайту достатньо обрати відповідну тему, яка забезпечує необхідний дизайн. Хоча можливе замовлення індивідуально розробленої теми, спеціально адаптованої до потреб конкретного ресурсу, у більшості випадків доцільно скористатися однією з наявних безкоштовних тем, які відповідають основним вимогам до оформлення [13, с. 12-14; 21, с. 50-51].

Найкраще місце для пошуку теми – це каталог тем WordPress.org <http://wordpress.org/extend/themes/> (починаючи з версії 2.8 WordPress, ми можемо зробити це безпосередньо з екрану адміністрування). Теми в цьому каталозі не тільки безкоштовні, але й проходять попередню перевірку користувачами, що забезпечує їхню надійність та коректне функціонування [13, с. 12-14; 21, с. 50-51].

Під час загального проєктування сайту дотримання традиційного та безпечного дизайну є не лише бажаним, а й необхідним. Користувачі звикли знаходити певні елементи інтерфейсу у передбачуваних місцях, і відхилення від цієї логіки може спричинити труднощі у навігації та зниження рівня зручності використання ресурсу. Основна мета дизайну полягає в тому, щоб сприяти зосередженню уваги відвідувачів на змісті сайту, а не на механізмах його взаємодії. Дотримання усталених конвенцій у структурі веб-сторінок підвищує зручність навігації. Індивідуальність проєкту може бути реалізована через візуальні та текстові елементи, зокрема заголовки та контент, проте

базовий макет доцільно формувати на основі перевірених шаблонів, поданих у проаналізованих нами джерелах [13, с. 12-14; 21, с. 50-51].

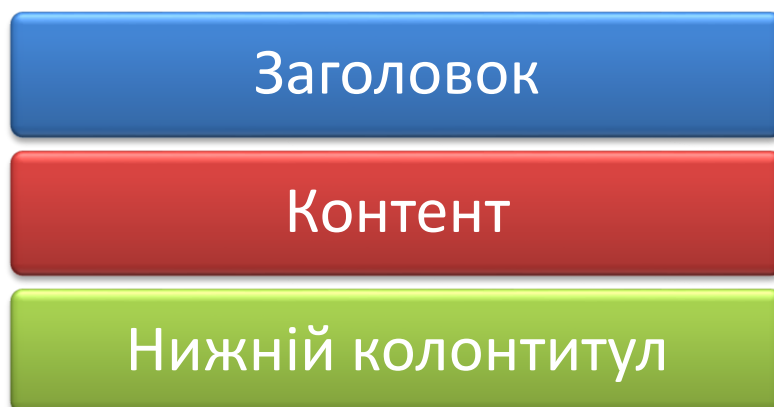


Рис. 2.2. Макет сайту

Ми починаємо з області заголовка вгорі, що включає меню всіх наших сторінок (фреймів), логотип навчального закладу та телефон приймальної комісії. Наш контент заповнить решту сторінки, а внизу буде розміщено нижній колонтитул, що включає контактні дані з посиланнями на наші сторінки в соцмережах, розклад роботи приймальної комісії та адресу навчального закладу [13, с. 12-14; 21, с. 56-57].

Розділ контенту сайту для сторінки «Головна» має містити низку функціональних елементів, спрямованих на інформування та підтримку абітурієнтів. Зокрема, передбачається наявність фрейму «Запис», який забезпечує швидку реєстрацію шляхом введення основних персональних даних (ім'я, телефон, електронна адреса) з подальшим підтвердженням через кнопку «Відправити». Крім того, важливо включити інформаційний блок, що міститиме відомості щодо спеціальності, переліку предметів для вступу, строків та вартості навчання, а також освітньо-професійної програми з можливістю її завантаження на пристрій користувача.

Доцільним є також створення фрейму «Наші викладачі», який дозволить потенційним студентам ознайомитися з викладацьким складом кафедри у форматі інтерактивного слайдера.

Окрему увагу слід приділити підрозділам «Наші переваги», «Студентська рада», що сприятимуть формуванню довіри до закладу.

Інструментом для попереднього професійного самовизначення може слугувати фрейм «Пройти тест», де реалізовано ІТ-тест на профорієнтацію.

Крім того, рекомендується впровадження розділу «Як нас знайти» з інтегрованою Google-картою для зручного визначення місця розташування освітньої установи. Для покращення комунікації з потенційними вступниками необхідно включити також розділ «FAQ/Питання», у якому подаватимуться відповіді на найбільш поширені запити.

Дизайн сайту – набагато більш індивідуальне питання, ніж макет сайту, але ми все одно повинні думати про своїх відвідувачів, і з цієї точки зору простота дизайну завжди краще. У простому дизайні легше зорієнтуватися і відвідувачі з більшою ймовірністю зосередяться на змісті. У випадку з сайтом-візиткою нам важливо створити правильний настрій (зосередженість, серйозність, спокій), водночас забезпечивши структурованість та зручність перегляду контенту без надмірного візуального навантаження [13, с. 12-14; 21, с. 56-57].

Щоб передати правильний настрій, ми використаємо світлий фон з абстрактним малюнком у вигляді геометричних фігур та контрастні темні кольори в оформленні об'єктів сторінок.

2.2. Реалізація сайту-візитки на основі СКВ WordPress

В розділі 1.3 ми вже описали загальні кроки для створення сайту-візитки засобами СКВ WordPress. Тепер опишемо дії для реалізації нашого сайту-візитки.

Спочатку встановимо потрібну нам тему.

Крок 1: Пошук та встановлення теми. Увійшовши до свого WordPress, ми потрапимо на головну сторінку інформаційної панелі. Праворуч ми побачимо деякі параметри екрана. Прокрутимо вниз і оберемо пункт «Оформлення». Коли ми оберемо «Зовнішній вигляд», з'являться деякі інші опції. Серед них оберемо пункт «Теми» [12, с. 64-65; 24, с. 75-78].

На сторінці керування темами у верхній частині розміщено кнопку «Додати нову». Натискання цієї кнопки відкриває вікно для пошуку тем. У полі пошуку необхідно ввести назву бажаної теми, наприклад, «GeneratePress», після чого система запропонує відповідний варіант для встановлення [12, с. 64-65; 24, с. 75-78].

Після того, як ми встановили тему під назвою «GeneratePress», наступним кроком буде її активація. Якщо ми не активуємо тему, вона не працюватиме на нашому сайті [12, с. 64-65; 24, с. 75-78]. Після активації ми можемо починати роботу. У деяких випадках система запропонує встановити додаткові плагіни для розширення функціональності теми [12, с. 64-65; 24, с. 75-78].

Крок 2: Встановлення плагіну. Після встановлення та активації теми «GeneratePress» наступним етапом є інсталяція відповідного плагіна, що забезпечує розширену функціональність для роботи з темою [12, с. 182-183; 24, с. 78-79]. Для встановлення плагіна необхідно у панелі адміністратора перейти до розділу «Плагіни» та обрати пункт «Додати новий». У полі пошуку слід ввести назву «GeneratePress», після чого встановити та активувати відповідний плагін.

Крок 3: Вибір дизайну. Для вибору дизайну необхідно у панелі керування перейти до розділу «Плагіни» та відкрити «Встановлені плагіни». У представленому списку слід знайти плагін «Початкові шаблони» та натиснути на опцію «Переглянути бібліотеку» [12, с. 190-191; 24, с. 79-80]. Після цього відкриється сторінка з доступними шаблонами дизайну, серед яких можна обрати відповідний варіант для подальшого використання [12, с. 190-191; 24, с. 79-80].

Крок 4: Вибір конструктора сторінок. Після переходу до бібліотеки шаблонів користувачу пропонуються варіанти редакторів сторінок, зокрема Elementor, Beaver Builder, Gutenberg і Brizy. Для подальшої роботи доцільно обрати редактор Elementor як один із найбільш зручних для візуального налаштування сторінок [12, с. 193 -195; 24, с. 80-84].

Перейдемо до дизайну хедера сайту. Хедер сайту – це верхня частина веб-сторінки. Він знаходиться у верхній частині кожної сторінки і виконує кілька дуже важливих функцій. Це більше, ніж просто місце для вашого логотипу; це частина послідовного користувацького досвіду, який поділяють усі хороші веб-сайти. Дизайн хедера сайту варіюється від одного веб-сайту до іншого [12, с. 185; 24, с. 85-86].

Як стратегічна частина сторінки, яку люди бачать у перші секунди завантаження сайту, хедер сайту діє як своєрідне запрошення. Хедер повинен містити основну інформацію про сайт, щоб користувачі могли за лічені секунди зрозуміти, що він пропонує [12, с. 185; 24, с. 85-86].

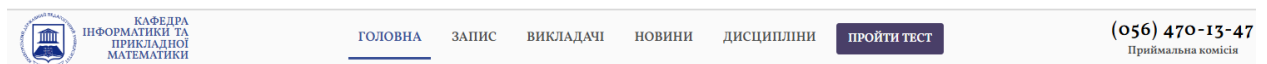


Рис. 2.3. Хедер сайту-візитки

Наступним кроком буде створення та налаштування головного меню. Для цього увійдемо до свого облікового запису WordPress, оберемо «Зовнішній вигляд», а потім «Налаштувати». Після натискання кнопки «Налаштувати» відкривається нова сторінка. На відкритій сторінці у розділі «Меню» слід обрати пункт «Навігація», де відображаються наявні на сайті пункти меню. Для додавання нових пунктів потрібно натиснути «Додати елементи», обрати потрібну сторінку (наприклад, «Головна») та підтвердити зміни, натиснувши «Опублікувати». Після цього вибраний пункт буде додано до меню сайту.

Якщо пункт меню, який ми хочемо додати, не існує як сторінка, публікація або категорія, нам доведеться повернутися до панелі керування WordPress, створити нову сторінку з потрібною нам назвою, розмістити на ній необхідний вміст, а потім клікнути кнопку «Опублікувати».

Коли ми це зробимо, це означитиме, що ми створили сторінку під назвою «Головна». Отже, ми можемо повторити процес, для інших команд меню. Це «Зовнішній вигляд» > «Налаштування» > «Меню» > «Навігація», а

потім обираємо створену нами сторінку з відповідною назвою і додаємо її до меню [12, с. 172-173; 24, с. 87-90].

Таким чином додамо всі нам потрібні команди меню до шапки сторінки (див. Рисунок 2.3).

Розглянемо створення контенту головного меню сайту за допомогою «Elementor».

Крок 1: Увійдіть до системи. Для редагування необхідно увійти до облікового запису WordPress і відкрити сайт (<http://kdpu-informatics.byethost32.com/>) у новій вкладці браузера. Після переходу на головну сторінку слід натиснути кнопку «Редагувати за допомогою Elementor» у верхній панелі інтерфейсу. Відкриється візуальний редактор, що забезпечує зміну структури та вмісту сторінки.

Редагування головної сторінки в середовищі «Elementor» здійснюється шляхом вибору відповідного текстового блоку та внесення змін до його вмісту. Це дозволяє адаптувати сторінку відповідно до індивідуальних потреб. Після завершення редагування необхідно натиснути кнопку «Оновити» у нижньому правому куті редактора для збереження змін.

Надалі для внесення правок застосовується той самий алгоритм: перехід на головну сторінку, відкриття редактора через пункт «Редагувати за допомогою Elementor», редагування вмісту та підтвердження змін натисканням «Оновити». Збережена інформація буде відображатися на сайті незалежно від того, чи авторизований користувач у WordPress [12, с. 175-177; 24, с. 90-92].

Додавання нового пункту меню з нуля. Увійдемо до панелі керування WordPress > «Вибрані сторінки» > «Додати новий» > Додамо нову назву («Новини») > «Редагувати з Elementor» > Перетягнемо модуль «Заголовок» у місце, відведене праворуч > Клікнемо на початковий шаблон із символом «А», щоб вибрати односторінковий шаблон для створення сторінки > Оберемо один із шаблонів (ми можемо обрати той самий дизайн/шаблон, що використовується на домашній сторінці та інших сторінках) > Оберемо один з

дизайнів оформлення, що відповідає обраній темі > Імпортуємо шаблон на сторінку.

Почавши редагувати тексти в темі ми також можемо видалити зображення в темі і замінити їх своїми власними, які ми можемо завантажити з комп'ютера.

Після завершення редагування сторінку слід оновити, натиснувши кнопку «Оновити» внизу редактора. Наступним етапом є додавання створеної сторінки до навігаційного меню сайту як окремого пункту. [12, с. 178-179; 24, с. 93-99].

Додавання сторінки для формування пункту меню. Оскільки ми вже створили сторінку з назвою «Новини», додати її як пункт меню дуже просто. Увійдемо до панелі керування WordPress > Клікнемо на «Зовнішній вигляд» > «Налаштувати» > «Меню» > «Навігація» (наразі встановлено: «Основне меню») > «Додати елементи» > Клікнемо на сторінку, яку ми хочемо додати («Новини») > Розташуємо їх у тому порядку, в якому вони були додані до пунктів меню, якщо нам не подобається їх розташування > «Опублікувати».

Після цього сторінка буде опублікована як пункт меню. Якщо ми відвідаємо свій веб-сайт, ми побачимо зміни. Наприклад, ми побачимо «Новини» як один з пунктів меню. Якщо ми клікнемо на нього, відкриється нова сторінка, на якій буде показано вміст сторінки [12, с. 172-174; 24, с. 101-106].

Розміщення логотипу на сайті. Логотип – це невеликий графічний знак, який ідентифікує компанію, асоціацію, організацію, бренд, команду, продукт, веб-сайт або особу.

Щоб завантажити логотип на наш сайт і зробити його видимим для відвідувачів, виконаємо наступні кроки: Увійдемо на свою адміністративну панель адміністратора WordPress > Оберемо опцію екрана «Зовнішній вигляд» > Знайдемо опцію «Налаштувати екран» і клікнемо на неї > Оберемо заголовок > Оберемо «Ідентифікація сайту» > Клікнемо на «Виберіть логотип» під заголовком «Логотип».

Після відкриття відповідної вкладки з'являється вікно для вибору логотипа. Зображення можна обрати з медіатеки WordPress, завантажити з локального пристрою або вставити з онлайн-джерел. Після вибору файлу натискаємо кнопку «Вибрати». За потреби зображення можна обрізати або залишити без змін, скориставшись опціями «Обрізати зображення» або «Пропустити обрізку». Після завершення налаштування логотипа слід натиснути кнопку «Опублікувати» та переглянути результат на сайті [12, с. 174; 24, с. 113-122].

Зміна назви та опису сайту. Нижче наведені кроки, як змінити назву та опис нашого сайту за допомогою панелі адміністратора WordPress: Увійдемо до панелі адміністратора > Оберемо «Зовнішній вигляд» > Оберемо «Налаштувати» > Оберемо «Заголовок» > Оберемо «Ідентифікатор сайту» > Внесемо необхідні зміни в розділі «Назва сайту» та «Теглайн».

Для зміни назви та опису сайту необхідно перейти до відповідного розділу налаштувань, внести нові значення у відповідні поля (наприклад, «Кафедра інформатики та прикладної математики»), після чого активувати опції «Відображати назву сайту» та «Відображати слоган». Для збереження змін слід натиснути кнопку «Опублікувати» у верхньому правому куті інтерфейсу, а потім перевірити оновлений вигляд сайту [12, с. 262-263; 24, с.124-130].

Віджети та їх використання на веб-сайтах. Віджети дозволяють користувачам WordPress додавати різні функції та можливості до своїх бічних панелей WordPress без написання коду. Віджети – це крихітні блоки певної функціональності, які ми можемо розмістити на бічних панелях нашого сайту (також відомі як області, готові для віджетів).

Віджети можуть бути використані для додавання наступних елементів на веб-сайти: хмарне сховище тегів на сайті, рядок пошуку на сайті, зображення, сторінки на сайті, панелі сайтів, відео, останні публікації, архіви, аудіо, категорії, календар на веб-сайті, посилання на соціальні мережі на веб-

сайтах, галереї, меню навігації та кастомний HTML для хедера або бічної панелі вашого сайту [12, с. 162-167; 24, с.132-133].

Додавання віджету галереї на сайт. Віджет «Галерея» надає нам простий спосіб відобразити фотогалерею або слайд-шоу в бічній панелі нашого сайту. Ці зображення можна розмістити в заголовку, бічній панелі або в нижньому колонтитулі сайту.

Кроки додавання галереї на сайт за допомогою віджету галереї: Увійдемо до адміністративної панелі > Клікнемо на «Віджети» > Оберемо «Галерея» > Оберемо область, в якій ми хочемо розмістити галерею.

Можливо, ми хочемо, щоб галерея була в верхньому, бічному або нижньому колонтитулі 1, 2, 3 або 4 – оберемо будь-яку область. Вкажемо назву галереї за бажанням. Клікнемо на кнопку «Додати зображення».

Відкриється нове спливаюче вікно з проханням відмітити якомога більше зображень, які ми хочемо розмістити в галереї. Клікнемо на кнопку «Створити нову галерею». Після того, як ми оберемо будь-яке зображення, ліворуч з'явиться інформація про нього. Там є «Назва», «Альтернативний текст», «Опис» і «Підпис».

Продовжуємо: Клікнемо «Вставити галерею» > Клікнемо «Зберегти», а потім «Готово» > Клікнемо на «Керувати за допомогою попереднього перегляду».

Додавання пунктів меню в нижній колонтитул. Для додавання пунктів меню в нижній колонтитул сайту, зробимо наступне: Увійдемо в адмінпанель нашого сайту на WordPress > Клікнемо на «Зовнішній вигляд» > Оберемо «Меню» > Клікнемо на кнопку «Створити нове меню» > Дамо назву меню. Після задання назви, наприклад, «Дизайн нижнього колонтитула», слід натиснути кнопку «Створити меню».

На наступному етапі у списку доступних сторінок необхідно обрати ті, що мають відображатися в нижній частині сайту, та додати їх до меню.

Далі: Перетягнемо пункти меню, щоб упорядкувати їх > Клікнемо на кнопку «Зберегти меню» > Відвідаємо свій сайт, щоб побачити, як з'явилися нові зміни [12, с.172-177; 24, с. 202-206].

Розміщення посилання на соціальні мережі на нашому сайті. Останнім часом важко знайти якісний сайт, на якому не було б кнопок або посилань на соціальні мережі. Розробники веб-сайтів використовують кнопки посилань на соціальні мережі для залучення трафіку на свої сайти [12, с. 238-239; 24, с. 206].

Щоб розмістити посилання на соціальні мережі на нашому веб-сайті, виконаємо наступні кроки.

Етап 1: Отримання зображень іконок соціальних мереж. Для додавання іконок соціальних мереж до сайту необхідно завантажити відповідні зображення. Їх можна знайти через пошукові системи або спеціалізовані ресурси, наприклад, <https://www.iconfinder.com/>. Завантажені зображення рекомендується зберегти у відповідну локальну папку для подальшого використання.

Етап 2: Завантаження зображень до медіа. Увійдемо до адміністративної панелі WordPress > Оберемо «Медіа» > Клікнемо «Додати новий» > Оберемо завантажені зображення іконок соціальних мереж з нашого комп'ютера і завантажимо їх до медіатеки > Клікнемо на «Віджети» > Оберемо віджет «Текст».

Коли віджет відкриється, уважно подивимося і оберемо опцію віджета «Текст».

Додавання текстового віджета відбувається наступним чином. Ми можемо додати цей текстовий віджет, клікнувши на кнопку «Додати» > Даємо віджету назву (ми можемо дати віджету назву «Зв'яжіться з нами») > Оберемо позицію в нижньому колонтитулі, де ми хочемо, щоб віджет з'явився.

Етап 3: Створення віджета нижнього колонтитула з посиланнями на соціальні мережі. Перейдемо до адміністративної панелі WordPress > Оберемо «Зовнішній вигляд» > Оскільки ми хочемо отримати посилання на соціальні

мережі в нижньому колонтитулі веб-сайту, нам слід вибрати «Колонтитул 1» > Натиснемо «Додати медіа» > Оберемо зображення соціальних мереж з медіатеки по черзі > Змінімо кілька деталей про кожне зображення.

У підзаголовку «Параметри відображення вкладень» необхідно встановити вирівнювання по лівому краю, у полі «Посилання на» обрати «Власна URL-адреса» та вставити відповідне посилання на соціальну мережу.

У полі «Розмір» доцільно встановити найменше доступне значення, оскільки іконки розміщуватимуться в нижньому колонтитулі сайту. Після цього натискаємо «Вставити в публікацію». Аналогічні дії виконуються для інших іконок із відповідними посиланнями.

Після завершення додавання всіх зображень необхідно зберегти віджет і перевірити результати на сайті. Натискання на будь-яку з іконок автоматично перенаправляє користувача на відповідну сторінку у соціальній мережі (рис. 2.4).

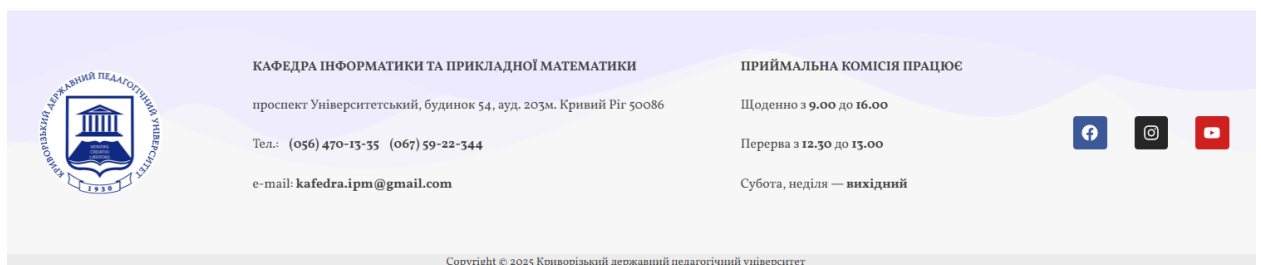


Рис. 2.4. Нижній колонтитул з посиланнями на соцмережі

2.3. Реалізація власних плагінів для розширення функціональності сайту-візитки

Перейдемо до реалізації власних плагінів для розширення функціоналу сайту-візитки.

Отже, перший необхідний для нас плагін – це «Календар новин KDPU» для WordPress. Плагін створює календар, який відображає дати публікацій новин на сайті WordPress, виділяючи ці дати особливим кольором та додаючи на них посилання.

Загальна структура та призначення плагіна. Плагін «Календар новин KDPU» створює віджет календаря, який можна розмістити на бічній панелі сайту WordPress або вставити безпосередньо в контент за допомогою спеціального коду (так званого «шорткоду» [news_calendar]). Основне призначення плагіна – відображати на календарі дати публікацій на сайті, щоб відвідувачі могли легко бачити, коли були опубліковані новини.

«	Травень 2025						»
Пн	Вт	Ср	Чт	Пт	Сб	Нд	
			1	2	3	4	
5	6	7	8	9	10	11	
12	13	14	15	16	17	18	
19	20	21	22	23	24	25	
26	27	28	29	30	31		

Рис. 2.5. Календар новин KDPU для WordPress

Основні поняття PHP для розуміння коду. Перш ніж перейти до детального опису, розглянемо кілька основних понять PHP:

1. Змінні – це контейнери для зберігання даних. У PHP перед назвою змінної ставиться знак долара \$. Наприклад: `$month = 5;` створює змінну з назвою «month» і присвоює їй значення 5.

2. Функції – це блоки коду, які виконують певну задачу. Функції мають назву і можуть приймати аргументи (вхідні дані). Наприклад: `function add_numbers($a, $b) { return $a + $b; }` – створює функцію з назвою «add_numbers», яка приймає два аргументи і повертає їх суму.

3. Класи – це шаблони для створення об’єктів. Класи містять властивості (змінні) та методи (функції). Наприклад: `class Calendar { public $month; public function display() { ... } }` – створює клас з назвою «Calendar» з властивістю «month» і методом «display».

4. Умовні оператори – дозволяють виконувати різні дії залежно від умов. Наприклад: `if ($day == 1) { echo "Перший день місяця"; }` – перевіряє, чи дорівнює змінна «day» значенню 1, і якщо так, то виводить текст.

5. Цикли – дозволяють повторювати дії кілька разів. Наприклад: `for ($i = 1; $i <= 10; $i++) { echo $i; }` - виводить числа від 1 до 10 [18; 20 с. 63-90].

Розбір коду плагіна.

1. Заголовок плагіна.

*/**

Plugin Name: Calendar News(KDPU)

Description: Виводить міні-календар з позначеними датами публікацій новин (оголошень).

Version: 2.1

Author: Гура Станіслав (покращено)

**/*

Це коментар, який містить метадані плагіна. WordPress використовує цю інформацію для відображення у списку плагінів в адміністративній панелі. Тут вказано назву плагіна, його опис, версію та автора.

2. Реєстрація шорткоду та віджету.

`add_shortcode('news_calendar', 'render_news_calendar');`

`add_action('widgets_init', 'register_custom_calendar_widget');`

- `add_shortcode` – це функція WordPress, яка реєструє новий шорткод (спеціальний код, який можна вставити в контент). У цьому випадку, реєструється шорткод [news_calendar], який викликає функцію `render_news_calendar` (описана далі в коді).

- `add_action` – це функція WordPress, яка додає дію (хук) до певної події. У цьому випадку, коли WordPress ініціалізує віджети (подія `widgets_init`), буде викликана функція `register_custom_calendar_widget`.

3. Функція реєстрації віджету.

`function register_custom_calendar_widget() {`

`unregister_widget('WP_Widget_Calendar');`

```

    register_widget('Custom_Calendar_Widget');
}

```

Ця функція:

1. Видаляє стандартний віджет календаря WordPress (`unregister_widget('WP_Widget_Calendar')`)
2. Реєструє новий віджет календаря `Custom_Calendar_Widget` (клас, описаний далі в коді)

4. Клас віджету календаря.

```

class Custom_Calendar_Widget extends WP_Widget {
    // Код класу
}

```

Клас `Custom_Calendar_Widget` розширює (наслідує) базовий клас `WP_Widget`, який надає WordPress. Це означає, що наш клас отримує всі функціональні можливості базового класу і може їх розширювати.

4.1. Конструктор класу.

```

public function __construct() {
    parent::__construct(
        'custom_calendar',
        'Календар новин',
        array('description' => 'Календар з відміченими датами публікацій')
    );
}

```

Конструктор класу – це спеціальна функція, яка викликається автоматично при створенні нового об'єкта класу. У цьому випадку, конструктор: викликає конструктор батьківського класу (`parent::__construct`), передаючи йому: унікальний ідентифікатор віджету (`'custom_calendar'`), назву віджету для відображення в адміністративній панелі (`'Календар новин'`), опис віджету (`'Календар з відміченими датами публікацій'`).

4.2. Відображення віджету.

```

public function widget($args, $instance) {

```

```

$title = apply_filters('widget_title', $instance['title']);
echo $args['before_widget'];
if (!empty($title)) {
    echo $args['before_title'] . $title . $args['after_title'];
}
echo render_news_calendar();
echo $args['after_widget'];
}

```

Ця функція відповідає за відображення віджету на сайті. Вона:

1. Отримує заголовок віджету з налаштувань (\$instance['title']) і застосовує до нього фільтри WordPress для правильного відображення.
2. Виводить HTML-код, який WordPress додає перед кожним віджетом (\$args['before_widget']).
3. Якщо заголовок не порожній, виводить його з відповідним оформленням.
4. Викликає функцію `render_news_calendar()`, яка генерує календар (описана далі в коді).
5. Виводить HTML-код, який WordPress додає після кожного віджету (\$args['after_widget']).

4.3. Форма налаштувань віджету.

```

public function form($instance) {
    $title = isset($instance['title']) ? $instance['title'] : 'Календар';
    ?>
    <p>
        <label for="<?php echo $this->get_field_id('title');
?>">Заголовок:</label>
        <input class="widefat" id="<?php echo $this->get_field_id('title'); ?>"
            name="<?php echo $this->get_field_name('title'); ?>" type="text"
            value="<?php echo esc_attr($title); ?>"
        </p>

```

```
<?php
}
```

Ця функція відображає форму налаштувань віджету в адміністративній панелі WordPress. Вона:

1. Отримує поточний заголовок віджету або використовує значення за замовчуванням ('Календар').
2. Виводить поле введення для заголовку віджету.
3. `$this->get_field_id('title')` і `$this->get_field_name('title')` – це функції базового класу, які генерують унікальні ідентифікатори та імена полів форми.
4. `esc_attr($title)` – це функція WordPress, яка очищує значення від потенційно небезпечних символів.

4.4. Збереження налаштувань віджету.

```
public function update($new_instance, $old_instance) {
    $instance = array();
    $instance['title'] = (!empty($new_instance['title'])) ?
strip_tags($new_instance['title']) : '';
    return $instance;
}
```

Ця функція викликається при збереженні налаштувань віджету і відповідає за обробку введених даних. Вона:

1. Створює новий масив для зберігання оброблених налаштувань.
2. Очищує заголовок від HTML-тегів за допомогою функції `strip_tags()` і зберігає його.

3. Повертає масив з обробленими налаштуваннями.

5. Функція створення календаря.

```
function render_news_calendar() {
    // Код функції
}
```

Ця функція є основною частиною плагіна і відповідає за створення HTML-коду календаря. Розглянемо її детально:

5.1. Підключення Google Fonts.

```
$output = "<link href='https://fonts.googleapis.com/css2?family=Vollkorn&
display=swap' rel='stylesheet'>";
```

Тут створюється змінна \$output, в яку записується HTML-код для підключення шрифту Vollkorn з Google Fonts. Ця змінна буде поступово заповнюватися HTML-кодом календаря, і в кінці функції буде повернена як результат.

5.2. Визначення поточного місяця і року.

```
$month          =          isset($_GET['calendar_month'])          ?
intval($_GET['calendar_month']) : date('n');
$year = isset($_GET['calendar_year']) ? intval($_GET['calendar_year']) :
date('Y');
```

Тут визначається, який місяць і рік відображати в календарі:

1. \$_GET – це глобальний масив в PHP, який містить параметри з URL.
2. isset(\$_GET['calendar_month']) перевіряє, чи є в URL параметр з назвою 'calendar_month'.
3. Якщо такий параметр є, використовуємо його значення, перетворюючи його на ціле число за допомогою функції intval().
4. Якщо такий параметр відсутній, використовуємо поточний місяць, отриманий за допомогою функції date('n').
5. Аналогічно для року, але використовуємо date('Y') для отримання поточного року.

5.3. Масив назв місяців.

```
$months_ua = [
    1 => 'Січень', 2 => 'Лютий', 3 => 'Березень', 4 => 'Квітень',
    5 => 'Травень', 6 => 'Червень', 7 => 'Липень', 8 => 'Серпень',
    9 => 'Вересень', 10 => 'Жовтень', 11 => 'Листопад', 12 => 'Грудень'
```

```
];
```

Це масив з назвами місяців українською мовою. Кожному номеру місяця (від 1 до 12) відповідає його назва.

5.4. Отримання дат публікацій.

```
global $wpdb;
```

```
$results = $wpdb->get_col("
    SELECT DISTINCT DATE(post_date) as post_date
    FROM $wpdb->posts
    WHERE post_type = 'post'
    AND post_status = 'publish'
    AND MONTH(post_date) = $month
    AND YEAR(post_date) = $year
");
```

Розглянутий вище код отримує дати публікацій з бази даних:

1. `global $wpdb` – отримує доступ до глобального об'єкта бази даних WordPress.
2. Виконується SQL-запит, який обирає унікальні дати (`DISTINCT DATE(post_date)`) з таблиці публікацій (`$wpdb->posts`), де: Тип публікації – `'post'` (`post_type = 'post'`), Статус публікації = `'publish'` (опубліковано) (`post_status = 'publish'`), Місяць дати публікації відповідає вибраному місяцю (`MONTH(post_date) = $month`), Рік дати публікації відповідає вибраному року (`YEAR(post_date) = $year`).
3. Результат запиту зберігається в змінній `$results` як масив значень.

5.5. Перетворення дат у формат «день місяця».

```
$post_dates = array_map(function($date) {
    return date('j', strtotime($date));
}, $results);
```

Тут використовується функція `array_map()`, яка застосовує вказану функцію до кожного елементу масиву:

1. Для кожної дати з масиву `$results` викликається функція `strtotime()`, яка перетворює текстову дату в UNIX-timestamp.

2. Потім функція `date('j', timestamp)` перетворює цей timestamp у день місяця (число від 1 до 31).

3. Результат зберігається в масиві `$post_dates` як лише числа дня місяця (без місяця і року).

5.6. Визначення кількості днів у місяці та дня тижня першого дня місяця.

```
$days_in_month = cal_days_in_month(CAL_GREGORIAN, $month,
$year);
```

```
$first_day_of_month = date('w', strtotime("$year-$month-01"));
```

```
$first_day_of_month = ($first_day_of_month + 6) % 7;
```

Розглянутий вище код:

1. Визначає кількість днів у вибраному місяці за допомогою функції `cal_days_in_month()`

2. Визначає день тижня для першого дня місяця:

`strtotime("$year-$month-01")` перетворює дату першого дня місяця в UNIX-timestamp

`date('w', timestamp)` повертає день тижня (0 – неділя, 6 – субота)

3. Перетворює стандартний формат дня тижня (0 – неділя) у формат, де понеділок – перший день тижня (0 – понеділок, 6 – неділя):

$(\$first_day_of_month + 6) \% 7$ – це математична формула для такого перетворення.

5.7. Створення посилань для навігації по місяцях.

```
$prev_month = $month - 1;
```

```
$prev_year = $year;
```

```
if ($prev_month < 1) {
```

```
    $prev_month = 12;
```

```
    $prev_year--;
```

```
}
```

```

$next_month = $month + 1;
$next_year = $year;
if ($next_month > 12) {
    $next_month = 1;
    $next_year++;
}
$current_url = remove_query_arg(array('calendar_month',
'calendar_year'));
$prev_url = add_query_arg(array('calendar_month' => $prev_month,
'calendar_year' => $prev_year), $current_url);
$next_url = add_query_arg(array('calendar_month' => $next_month,
'calendar_year' => $next_year), $current_url);

```

Розглянутий вище код створює посилання для навігації по місяцях:

1. Розраховує попередній місяць і рік: якщо попередній місяць менше 1 (тобто поточний місяць – січень), встановлює попередній місяць як грудень і зменшує рік на 1.

2. Розраховує наступний місяць і рік: якщо наступний місяць більше 12 (тобто поточний місяць – грудень), встановлює наступний місяць як січень і збільшує рік на 1.

3. Створює URL-адреси для посилань на попередній і наступний місяці: `remove_query_arg()` видаляє з поточного URL параметри календаря, `add_query_arg()` додає нові параметри до URL-адреси.

5.8. Формування HTML-коду календаря.

```
$output .= '<div style="font-family: Vollkorn, serif; width: 100%; max-width: 300px;">';
```

```
$output .= '<table style="border-collapse: collapse; width: 100%; text-align: center;">';
```

У наведеному прикладі коду показано, як починається формування HTML-коду календаря:

1. Додається div-контейнер з шрифтом Vollkorn і обмеженням ширини до 300 пікселів.

2. Додається таблиця з центрованим текстом.

5.9. Заголовок календаря з навігацією.

```
$output .= '<tr>';
```

```
$output .= '<th style="background: rgb(180, 180, 180); padding: 5px;">';
```

```
$output .= '<a href="' . $prev_url . '" style="color: #333; text-decoration: none;">&laquo;</a>';
```

```
$output .= '</th>';
```

```
$output .= '<th colspan="5" style="font-size: 20px; background: rgb(180, 180, 180); padding: 10px 0;">';
```

```
$output .= $months_ua[$month] . ' . $year;
```

```
$output .= '</th>';
```

```
$output .= '<th style="background: rgb(180, 180, 180); padding: 5px;">';
```

```
$output .= '<a href="' . $next_url . '" style="color: #333; text-decoration: none;">&raquo;</a>';
```

```
$output .= '</th>';
```

```
$output .= '</tr>';
```

Наведений фрагмент коду реалізує заголовок календаря:

1. Перший стовпець містить посилання на попередній місяць (« – це HTML-код для символу «).

2. Середні п'ять стовпців об'єднуються (colspan="5") і містять назву місяця і рік.

3. Останній стовпець містить посилання на наступний місяць (» – це HTML-код для символу »).

4. Усі стовпці мають сірий фон (rgb(180, 180, 180)).

5.10. Дні тижня (заголовки).

```
$output .= '<tr>';
```

```
$output
.=
'<th>Пн</th><th>Вт</th><th>Ср</th><th>Чт</th><th>Пт</th><th>Сб</th><th>
>Нд</th>';
```

```
$output .= '</tr>';
```

Тут додаються заголовки для днів тижня українською мовою.

5.11. Заповнення календаря днями місяця.

```
$output .= '<tr>';
```

```
$day = 1;
```

```
$i = $first_day_of_month;
```

```
for ($j = 0; $j < $first_day_of_month; $j++) {
```

```
    $output .= '<td></td>';
```

```
}
```

```
while ($day <= $days_in_month) {
```

```
    if (($i % 7) == 0 && $day != 1) {
```

```
        $output .= '</tr><tr>';
```

```
    }
```

```
    if (in_array($day, $post_dates)) {
```

```
        $link = get_day_link($year, $month, $day);
```

```
        $output .= "<td style='background: #493F68; padding: 5px; font-size:
18px'>";
```

```
        $output .= "<a href='$link' style='color: #FFFFFF; font-weight: bold;
text-decoration: none;'>$day</a>";
```

```
        $output .= "</td>";
```

```
    } else {
```

```
        $output .= "<td style='padding: 5px;'>$day</td>";
```

```
    }
```

```
    $day++;
```

```
    $i++;
```

```
}
```

```
while (($i % 7) != 0) {
```

```

$output .= '<td></td>';
$i++;
}
$output .= '</tr></table></div>';

```

Розглянутий вище код демонструє заповнення календаря днями місяця:

1. Починається новий рядок таблиці (<tr>).
2. Ініціалізуються змінні: \$day = 1 – початковий день місяця, \$i = \$first_day_of_month – поточна позиція в тижні (від 0 до 6).
3. Додаються порожні клітинки на початку місяця, якщо перший день місяця не понеділок: цикл виконується від 0 до \$first_day_of_month – 1, для кожної ітерації додається порожня клітинка (<td></td>).
4. Заповнюються дні місяця (від 1 до кількості днів у місяці): цикл виконується, поки \$day <= \$days_in_month, якщо досягнуто кінця тижня (((\$i % 7) == 0) і це не перший день місяця (\$day != 1), закінчується поточний рядок і починається новий (</tr><tr>). Перевіряється, чи є публікації в цей день (in_array(\$day, \$post_dates)): якщо є, клітинка має фіолетовий фон (#493F68) і містить посилання на архів публікацій за цей день; якщо немає, клітинка просто містить номер дня. Збільшуються лічильники \$day і \$i.
5. Додаються порожні клітинки в кінці місяця, щоб завершити тиждень: цикл виконується, поки (\$i % 7) != 0, для кожної ітерації додається порожня клітинка (<td></td>).
6. Закриваються всі відкриті HTML-теги (</tr></table></div>).

5.12. Повернення HTML-коду календаря.

```
return $output;
```

Функція повертає сформований HTML-код календаря, який буде виведений на сторінці.

Загальна логіка роботи плагіна

1. Плагін реєструє шорткод [news_calendar] і віджет «Календар новин».

2. Коли користувач переглядає сторінку з шорткодом або віджетом, викликається функція `render_news_calendar()`.
3. Функція визначає поточний місяць і рік або використовує параметри з URL.
4. Функція отримує з бази даних дати публікацій новин для вибраного місяця і року.
5. Функція формує HTML-код календаря, виділяючи дати з публікаціями та додаючи посилання на них.
6. Функція повертає HTML-код, який вставляється на сторінку.

Логіка роботи плагіна з датою.

1. Поточна дата визначається за допомогою функції `date()` з PHP: `date('n')` повертає номер поточного місяця (від 1 до 12), `date('Y')` повертає поточний рік (наприклад, 2025).
2. Користувач може змінити місяць і рік за допомогою навігаційних стрілок у календарі, які додають до URL параметри `calendar_month` і `calendar_year`.
3. Плагін перевіряє наявність цих параметрів у URL і використовує їх для вибору місяця і року для відображення.
4. Дати публікацій визначаються за допомогою SQL-запиту до бази даних, який обирає дати публікацій для вибраного місяця і року.

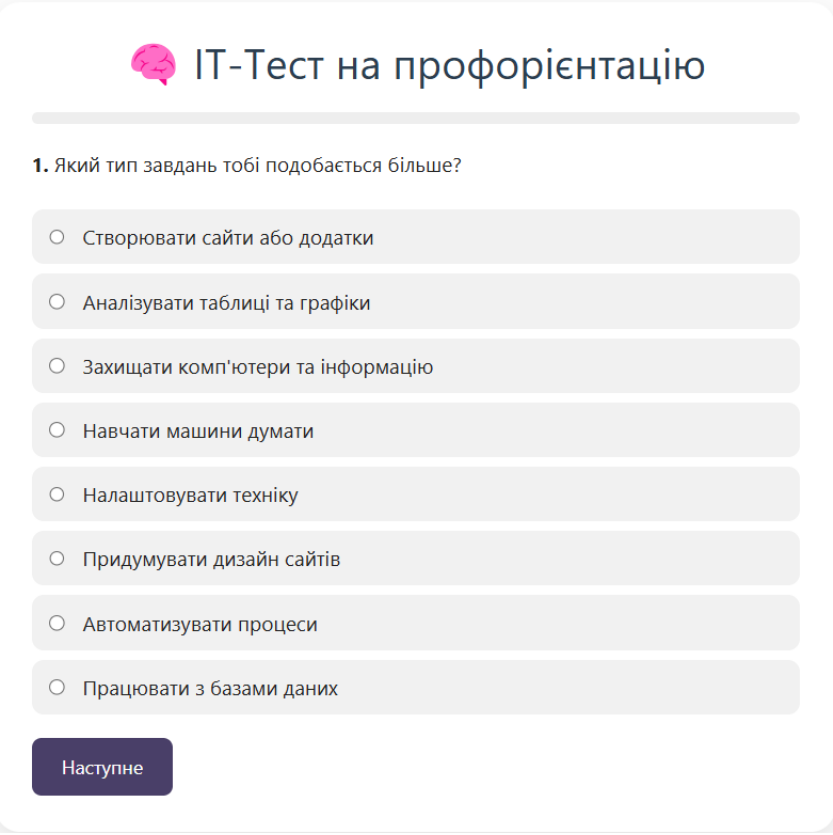
Як плагін визначає, які дати мають публікації.

1. Плагін виконує SQL-запит до бази даних WordPress, який обирає унікальні дати публікацій новин для вибраного місяця і року.
2. Отримані дати перетворюються у формат «день місяця» (число від 1 до 31) і зберігаються в масиві `$post_dates`.
3. При формуванні календаря для кожного дня місяця перевіряється, чи є він у масиві `$post_dates`: якщо день є в масиві, він відображається з фіолетовим фоном і посиланням на архів публікацій за цей день; якщо дня немає в масиві, він відображається як звичайний день без виділення.

Таким чином, плагін «Календар новин (KDPU)» для WordPress створює календар, який показує дати публікацій новин на сайті. Плагін можна використовувати як віджет на бічній панелі або вставляти безпосередньо в контент за допомогою шорткоду [news_calendar]. Календар виділяє дати з публікаціями фіолетовим кольором і додає посилання на архів публікацій за цей день. Користувач може переміщатися між місяцями за допомогою навігаційних стрілок.

Цей плагін є корисним інструментом для відображення активності публікацій на сайті та надання відвідувачам зручного способу знаходження новин за датами.

Наступний плагін – «ІТ-тест для початківців» – призначений для створення віджета, що забезпечує перевірку базових знань у сфері інформаційних технологій з метою профорієнтації користувачів-початківців.



 **ІТ-Тест на профорієнтацію**

1. Який тип завдань тобі подобається більше?

- ☐ Створювати сайти або додатки
- ☐ Аналізувати таблиці та графіки
- ☐ Захищати комп'ютери та інформацію
- ☐ Навчати машини думати
- ☐ Налаштовувати техніку
- ☐ Придумувати дизайн сайтів
- ☐ Автоматизувати процеси
- ☐ Працювати з базами даних

Наступне

Рис. 2.6. ІТ-тест для початківців

Загальна структура коду

Код складається з трьох технологій:

1. PHP – мова серверної частини (бекенд);
2. HTML/CSS – розмітка та стилі (фронтенд);
3. JavaScript – інтерактивність (фронтенд).

Частина 1: Заголовок плагіна та безпека.

```
/**
```

```
 * Plugin Name: IT-Тест на профорієнтацію
```

```
 * Description: IT-тест із 10 послідовних питань, що допомагає визначити
найкращий напрям для новачків, з прогрес-баром.
```

```
 * Version: 1.5
```

```
 * Author: Гура Станіслав
```

```
 */
```

```
if (!defined('ABSPATH')) exit;
```

Перший блок коду є коментарем із метаданими, що ідентифікується системою WordPress як службова інформація про плагін.

Другий рядок виконує функцію захисного механізму: він перевіряє, чи визначено константу ABSPATH, яка наявна лише в середовищі WordPress. Це дозволяє запобігти несанкціонованому прямому виконанню файлу поза межами CMS WordPress, що підвищує рівень безпеки.

Частина 2: Створення шорткоду

```
function it_test_shortcode()
```

```
{
```

```
    ob_start();
```

```
    // HTML, CSS і JavaScript код тут
```

```
    return ob_get_clean();
```

```
}
```

```
add_shortcode('it_test', 'it_test_shortcode');
```

Пояснення:

Функція – це блок коду, який виконує певну задачу [20, с. 95].

it_test_shortcode() – назва функції, яка буде викликатися, коли WordPress зустрине шорткод.

Шорткод – це спеціальний код у квадратних дужках (наприклад, [it_test]), який можна вставити в статтю або сторінку WordPress.

ob_start() запускає «буферизацію виводу» – весь HTML, який виводиться далі, не показується одразу, а зберігається в пам'яті.

ob_get_clean() завершує буферизацію, повертає накопичений HTML і очищає буфер.

add_shortcode('it_test', 'it_test_shortcode') реєструє шорткод у WordPress, пов'язуючи текст [it_test] з функцією it_test_shortcode().

Частина 3: HTML структура тесту

```
<div id="it-test">
  <h2> IT-Тест на профорієнтацію</h2>
  <div id="progress-bar">
    <div id="progress-bar-fill"></div>
  </div>
  <form id="test-form"></form>
  <button id="next-button">Наступне</button>
  <div id="result"></div>
</div>
```

Пояснення:

<div> – контейнер, що групує елементи.

id="it-test" – унікальний ідентифікатор для головного контейнера.

<h2> – заголовок другого рівня.

Перший вкладений <div> створює прогрес-бар із двох елементів.

<form> буде містити питання (вони додаються пізніше через JavaScript).

<button> – кнопка для переходу до наступного питання.

Останній <div> призначений для відображення результатів тесту [7, с. 20-23].

Частина 4: CSS стилі.

```
<style>
```

```
#it-test {
    max-width: 700px;
    margin: 40px auto;
    font-family: "Segoe UI", sans-serif;
    background: #fff;
    padding: 30px;
    border-radius: 20px;
    box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);
}

/* інші стилі... */
</style>
```

Пояснення:

CSS визначає візуальний вигляд елементів.

Селектор `#it-test` націлений на елемент з `id="it-test"`.

`max-width: 700px` обмежує ширину контейнера.

`margin: 40px auto` створює відступи 40 пікселів зверху і знизу та автоматично центрує контейнер.

`font-family` встановлює шрифт.

`background: #fff` встановлює білий фон.

`padding: 30px` створює внутрішні відступи.

`border-radius: 20px` округлює кути контейнера.

`box-shadow` створює тінь для ефекту об'єму.

Інші важливі стилі:

`.question` і `.active` керують видимістю питань.

`.answers label` стилізує варіанти відповідей.

`#progress-bar` і `#progress-bar-fill` формують індикатор прогресу.

`#result` і `.description` оформлюють блок

результатів [7, с. 53 – 58; 20, с. 425 - 450].

Частина 5: JavaScript – Управління даними.

```
<script>
```

```
const directions = {
  dev: 0,
  data: 0,
  cyber: 0,
  ai: 0,
  admin: 0,
  design: 0,
  devops: 0,
  db: 0
};
// інші дані...
```

</script>

Пояснення:

JavaScript додає інтерактивність до веб-сторінки.

const створює константу (змінну, яку не можна перепризначити).

directions – об’єкт, який зберігає лічильники балів для кожного напрямку

ІТ.

Кожен ключ (dev, data тощо) представляє ІТ-напрямок, а значення (спочатку всі 0) – кількість балів [20, с. 113; 23, с. 120-121].

Частина 6: JavaScript – Рекомендації.

```
const recommendations = {
```

```
  dev: "📖 Вам підходить напрям <strong>Розробка</strong>!<div
class='description'>Розробники створюють сайти, додатки та програми, які ми
використовуємо щодня. Ми будемо писати код на різних мовах програмування
(JavaScript, Python, Java тощо), вирішувати логічні завдання та втілювати ідеї
в реальні проекти. Це творча робота з конкретними результатами, які можна
одразу побачити.</div>",
```

```
  // інші рекомендації...
```

```
};
```

Пояснення:

recommendations – об'єкт, який містить HTML-код для кожного можливого результату тесту.

Кожен ключ (dev, data тощо) відповідає напрямку з об'єкта directions [11, с. 120-121; 20, с. 322-337].

Значення – це HTML-код із емоїї, назвою напрямку та описом.

 робить текст жирним.

<div class='description'> створює контейнер для детального опису [7, с. 20-23].

Частина 7: JavaScript – Питання тесту

```
const questions = [{
  question: "Який тип завдань тобі подобається більше?",
  options: [{
    text: "Створювати сайти або додатки",
    value: "dev"
  },
  // інші варіанти...
]
},
// інші питання...
];
```

Пояснення:

questions – масив об'єктів, де кожен об'єкт представляє одне питання.

Кожне питання має властивість question (текст питання) і options (варіанти відповідей).

Options – масив об'єктів, де кожен об'єкт має:

text – текст варіанту відповіді;

value – код IT-напрямку, який пов'язаний з цією відповіддю.

Частина 8: JavaScript – Створення форми тесту.

```
const form = document.getElementById("test-form");
```

```

const resultDiv = document.getElementById("result");
const nextBtn = document.getElementById("next-button");
const progressBar = document.getElementById("progress-bar-fill");
let currentQuestion = 0;
questions.forEach((q, index) => {
  const qDiv = document.createElement("div");
  qDiv.className = "question" + (index === 0 ? " active" : "");
  qDiv.innerHTML = `<p><strong>${index + 1}</strong>
  ${q.question}</p>`;
  const answerBlock = document.createElement("div");
  answerBlock.className = "answers";
  q.options.forEach((opt, i) => {
    const id = `q${index}_${i}`;
    answerBlock.innerHTML += `
      <label>
        <input type="radio" name="q${index}" value="${opt.value}"
required>
        ${opt.text}
      </label>`;
  });
  qDiv.appendChild(answerBlock);
  form.appendChild(qDiv);
});
nextBtn.style.display = "block";
updateProgress();

```

Пояснення:

Перші рядки отримують посилання на HTML-елементи через їхні ID.

`let currentQuestion = 0` створює змінну, яка відстежує номер поточного питання.

`questions.forEach()` проходить по кожному питанню у масиві.

`document.createElement("div")` створює новий HTML-елемент `<div>`.

`qDiv.className = "question" + (index === 0 ? " active" : "")` додає клас `question` і, якщо це перше питання, також клас `active`.

`qDiv.innerHTML` встановлює HTML-вміст елемента.

Другий цикл `forEach` створює варіанти відповідей для кожного питання.

`qDiv.appendChild(answerBlock)` додає блок відповідей до блоку питання.

`form.appendChild(qDiv)` додає блок питання до форми.

`nextBtn.style.display = "block"` робить кнопку "Наступне" видимою.

`updateProgress()` оновлює індикатор прогресу.

Частина 9: JavaScript – Обробка переходів між питаннями

```
nextBtn.addEventListener("click", () => {
    const selected =
document.querySelector(`input[name='q${currentQuestion}']:checked`);
    if (!selected) return alert("Оберіть відповідь");
    directions[selected.value]++;
    document.querySelectorAll(".question")[currentQuestion].classList.remove(
"active");
    currentQuestion++;
    if (currentQuestion < questions.length) {

document.querySelectorAll(".question")[currentQuestion].classList.add("active");
        updateProgress();
    } else {
        nextBtn.style.display = "none";
        const max = Math.max(...Object.values(directions));
        const top = Object.keys(directions).find(key => directions[key] ===
max);
        resultDiv.innerHTML = recommendations[top] || "Не вдалося
визначити напрям.";
        resultDiv.style.display = "block";
    }
});
```

```

        progressBar.style.width = "100%";
    }
});

```

Пояснення:

`nextBtn.addEventListener("click", () => { ... })` встановлює функцію, яка виконується при натисканні кнопки "Наступне".

`document.querySelector()` знаходить вибраний варіант відповіді.

`if (!selected) return alert("Оберіть відповідь")` перевіряє, чи вибрано відповідь, і показує попередження, якщо ні.

`directions[selected.value]++` збільшує лічильник для вибраного ІТ-напрямку.

Наступні рядки приховують поточне питання і відображають наступне.

Умова `if (currentQuestion < questions.length)` перевіряє, чи є ще питання.

Якщо питань більше немає, обчислюється результат: `Math.max(...Object.values(directions))` знаходить найбільшу кількість балів.

`Object.keys(directions).find(key => directions[key] === max)` знаходить напрямок з найбільшою кількістю балів.

`resultDiv.innerHTML = recommendations[top]` відображає рекомендацію для цього напрямку.

Частина 10: JavaScript – Оновлення прогрес-бару

```

function updateProgress() {
    const percent = Math.round((currentQuestion) / questions.length * 100);
    progressBar.style.width = percent + "%";
}

```

Пояснення:

`updateProgress()` – функція, яка оновлює індикатор прогресу.

`Math.round((currentQuestion) / questions.length * 100)` обчислює відсоток пройдених питань.

`progressBar.style.width = percent + "%"` встановлює ширину індикатора прогресу відповідно до цього відсотка.

Підсумок роботи коду.

1. Коли шорткод [it_test] додається на WordPress-сторінку, викликається функція it_test_shortcode().
2. Ця функція повертає HTML, CSS і JavaScript для тесту.
3. JavaScript створює всі питання і варіанти відповідей на основі даних.
4. Коли користувач відповідає на питання і натискає «Наступне»: бал для вибраного ІТ-напрямку збільшується, поточне питання ховається, наступне відображається, прогрес-бар оновлюється.
5. Після останнього питання: обчислюється напрямок з найбільшою кількістю балів, відображається відповідна рекомендація.

Результат роботи віджета для користувача.

1. Користувач бачить форму з першим питанням і варіантами відповідей.
2. Після вибору відповіді і натискання «Наступне», з'являється наступне питання.
3. Прогрес-бар показує, скільки питань вже пройдено.
4. Після останнього питання користувач бачить результат – рекомендований ІТ-напрямок з описом.

Цей плагін є чудовим прикладом того, як поєднання PHP, HTML, CSS і JavaScript може створити інтерактивний інструмент на WordPress-сайті.

Висновки до розділу 2

У другому розділі було здійснено комплексне проєктування та реалізацію сайту-візитки на основі системи керування вмістом WordPress. На початковому етапі сформульовано вимоги до функціонального наповнення та дизайну майбутнього ресурсу. Розроблено макет сайту, який відображає логічну структуру взаємопов'язаних сторінок і блоків інтерфейсу.

Особливу увагу приділено деталізації вимог до змістового наповнення кожної сторінки, а також опису функціонального призначення окремих

елементів. У процесі реалізації сайту було надано поетапний опис створення структури ресурсу із використанням наявних плагінів, вбудованих у застосовану тему WordPress.

Крім того, розроблено кілька власних плагінів, описано принципи їх створення, подано фрагменти програмного коду, реалізованого засобами PHP, HTML/CSS та JavaScript, а також пояснено порядок їх інтеграції з контентом сайту.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було проведено аналіз сучасних платформ для створення веб-сайтів. Нами було вивчено всі класифікації СКВ. Ми визначили їх ключові особливості, переваги та недоліки. Це дало можливість зупинити свій вибір на СКВ-конструкторах, оскільки вони виявились найуніверсальнішим засобом для створення сайтів.

Також, нами було здійснено порівняльний аналіз системи керування вмістом WordPress з іншими сучасними системами – Joomla та Drupal. Для цього ми створили порівняльну таблицю з найважливішими характеристиками для побудови майбутнього сайту. Завдяки чому було виявлено, що СКВ WordPress є найефективнішою за п'ятьма ключовими характеристиками – доступністю розширень, діапазоном функціональності, якістю документації, активністю користувацької спільноти та простотою використання. Отже ми впевнились в доцільності її використання у створенні нашого сайту-візитки.

Ми вивчили особливості використання СКВ WordPress для створення сайту-візитки. Розглянули специфіку її налаштувань, вибір шаблонів та плагінів для отримання потрібного результату при побудові сайту-візитки.

Нами було визначено вимоги до функціональності та дизайну сайту-візитки для абітурієнтів, розроблено макет та мапу, що суттєво спростило побудову нашого сайту.

Для реалізації сайту-візитки на основі СКВ WordPress відповідно до сформульованих вимог ми крок за кроком розписали алгоритми додавання й налаштування сторінок та плагінів за допомогою засобу керування C-Panel, додавання потрібних елементів на сторінку за допомогою конструктора сторінок Elementor. Також було досліджено використання такого віджету, як «Галерея» з повним його налаштуванням. Окрему увагу ми приділили створенню нижнього колонтитулу з додаванням посилань на наші сторінки в соціальних мережах. Та, головне, ми створили два власних плагіна – «Календар новин KDPU» і «ІТ-тест для початківців».

Перший плагін «Календар новин KDPU» створює календар, який відображає дати публікацій новин на сайті WordPress, виділяючи ці дати особливим кольором та додаючи на них посилання. Завдяки йому відвідувачі можуть легко бачити, коли були опубліковані новини та переходити безпосередньо до потрібної новини. В результаті відкривається сторінка з вибраною новиною.

Другий – «ІТ-тест для початківців» створює віджет для перевірки знань початківців на профорієнтацію. Цей плагін створює інтерактивний інструмент на сайті для перевірки знань абітурієнтів й допомагає їм обрати напрям навчання.

Отже, всі поставлені нами завдання дослідження були виконані в повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ганна Чала. Порівнюємо WordPress, Joomla та Drupal: яка CMS краще. URL: <https://hostiq.ua/blog/ukr/wordpress-joomla-drupal-comparison/> (Дата звернення: 16.05.2025).
2. Максим Дімура. CMS WordPress для чого підходить. URL: <https://www.site2b.ua/ua/web-blog-ua/CKB-wordpress-dlya-chogo-pidxodit.html/> (Дата звернення: 11.05.2025).
3. Національний університет «Львівська політехніка». Системи керування контентом. URL: <https://www.victoria.lviv.ua/library/students/wd/work10.html/> (Дата звернення: 01.03.2025).
4. Олена Бочарнікова. Як зробити самому сайт-візитку на WordPress: покрокова інструкція. URL: <https://marketing-2b.com/kak-sdelat-samomu-sajt-vizitku-na-wordpress-poshagovaya-instrukciya/> (Дата звернення: 11.05.2025).
5. Титенко С. В. Інформація курсу по матеріалам: Web-орієнтовані інформаційні системи. Методичні вказівки до вивчення дисципліни для студентів напрямку підготовки 6.050103 «Програмна інженерія». К.: НТУУ «КПІ» 2015. С. 51. URL: <http://www.znannya.org/?view=CKB-tytenko-1/> (Дата звернення: 15.05.2025).
6. Шинкарук О. М., Яшина О. М., Онишко О. Г. Управління якістю програмних веб-систем засобами розробки. *Вісник Хмельницького національного університету. Технічні науки*. 2020. № 6 (2020). С. 39-44. URL: <https://elar.khmnu.edu.ua/items/cc018e4f-6c1a-4f9f-b09d-df3309fb4c25//> (Дата звернення: 01.03.2025).
7. Ashley Menhennett, Pablo Farias Navarro. A Guide to HTML5 and CSS3. ZENVA, 2014.
8. Dan Rahmel. Beginning Joomla: From Novice to Professional. Apress, 2007.
9. Deane Barker. Web Content Management. O'Reilly Media, Inc., 2016.

10. Drupal User Guide. Concept: Drupal as a Content Management System.
URL: https://www.drupal.org/docs/user_guide/en/understanding-drupal.html/ –
(Дата звернення: 11.05.2025).
11. Eric Elliott. Programming JavaScript Applications. Published by O'Reilly Media, Inc., 2014.
12. George Plumley. Teach Yourself VISUALLY™ WordPress® 3rd Edition. John Wiley & Sons, Inc., 2015.
13. George Plumley. WordPress 24-Hour Trainer, Third Edition. Wiley Publishing, Inc., Indianapolis, Indiana, 2015.
14. Getting Started with WordPress. SolidWP, 2025.
15. Jennifer Hodgdon. Programmer's Guide to Drupal. O'Reilly Media, Inc., 2016.
16. Joomla! Documentation. Портал: Вчимо більше.
URL: https://docs.joomla.org/Portal:Learn_More/uk/ (Дата звернення: 11.05.2025).
17. Jose-Manuel Martinez-Caro, Antonio-Jose Aledo-Hernandez, Antonio Guillen-Perez, Ramon Sanchez-Iborra and Maria-Dolores Cano. A Comparative Study of Web Content Management Systems. MDPI, 2018.
18. Josh Lockhart. PHP. The right way. URL: <https://iflista.github.io/php-the-right-way/> (Дата звернення: 18.05.2025).
19. Lisa Sims. Building Your Online Store With WordPress and WooCommerce: Learn to Leverage the Critical Role E-commerce Plays in Today's Competitive Marketplace, 2018.
20. Robin Nixon. Learning PHP, MySQL & JavaScript. O'Reilly Media, Inc., 2021.
21. Sean McManus. Web Design in Easy Steps. 7th Edition, 2023.
22. Teaganne Finn. Amanda Downie. What is a content management system (CMS).
URL: <https://www.ibm.com/think/topics/content-management-system/> (Дата звернення: 11.05.2025).
23. Web Design for Beginners Seventh Edition. Imagine Publishing Ltd, 2016.
24. William S. Page. How to Build and Design a Website using WordPress: A Step-by-Step Guide with Screenshots, 2020.