

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя  
Факультет комп'ютерно-інформаційних систем та програмної інженерії

(повна назва факультету)

Кафедра програмної інженерії

(повна назва кафедри)

## КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: «Розробка шаблону сайту кафедри згідно вимог з можливістю  
адміністрування, модерації, конфігурування на WordPress»

Виконав: студент 6 курсу, групи СПМ-61  
спеціальності 121 «Інженерія програмного  
забезпечення»

(шифр і назва спеціальності)

\_\_\_\_\_ Малярський В.О.  
(підпис) (прізвище та ініціали)

Керівник \_\_\_\_\_ Дячук С.Ф.  
(підпис) (прізвище та ініціали)

Нормоконтроль \_\_\_\_\_ Бойко І.В.  
(підпис) (прізвище та ініціали)

Завідувач кафедри \_\_\_\_\_ Петрик М.Р.  
(підпис) (прізвище та ініціали)

Рецензент \_\_\_\_\_ Баран І.О.  
(підпис) (прізвище та ініціали)

Тернопіль 2020

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет Факультет комп'ютерно-інформаційних систем та програмної інженерії  
(повна назва факультету)

Кафедра Кафедра програмної інженерії  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Петрик М.Р.  
(прізвище та ініціали)

(підпис)

«    »

20\_\_ р.

## ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр  
(назва освітнього ступеня)

за спеціальністю 121 «Інженерія програмного забезпечення»  
(шифр і назва спеціальності)

студенту Малярському Віктору Олеговичу  
(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка шаблону сайту кафедри згідно вимог з можливістю адміністрування, модерації, конфігурування на WordPress»

Керівник роботи к.т.н., доцент Дячук Степан Федорович  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «\_\_» \_\_\_\_\_ 20\_\_ року № \_\_\_\_\_

2. Термін подання студентом завершеної роботи \_\_\_\_\_

3. Вихідні дані до роботи Веб-сайт кафедри програмної інженерії за посиланням «<https://kaf-pi.tntu.edu.ua/>», віддалений репозиторій на GitHub, пояснювальна записка.

4. Зміст роботи (перелік питань, які потрібно розробити)

Метою та завданням атестаційної роботи є удосконалення програмного забезпечення системи керування контентом типу WordPress в умовах обмежених інформаційних ресурсів. Адаптація системи до використання на різних девайсах та забезпечення можливості адміністрування, модерації та конфігурування її адміністраторами.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)  
Мультимедійна презентація на 19 слайдів.



## РЕФЕРАТ

Атестаційна робота на тему «Розробка шаблону сайту кафедри згідно вимог з можливістю адміністрування, модерації, конфігурування на WordPress». Малярський Віктор Олегович. Тернопільський національний технічний університет імені Івана Пулюя, Факультет комп'ютерно-інформаційних систем і програмної інженерії, Кафедра програмної інженерії, група СПм-61, Тернопіль, 2020. С. – 72, розділи – 4, рисунки – 21.

Метою та завданням атестаційної роботи є удосконалення сайту кафедри програмної інженерії, створеного на системі WordPress. В процесі виконання роботи було проведено дослідження методології CI/CD, впровадження системи контролю версій та сучасних методик розробки веб-сайтів в цілому. В результаті розробки було оновлено систему створення сторінок сайту, адаптовано існуючі сторінки до перегляду на мобільних пристроях та встановлено сайт на систему контролю версій.

Ключові слова: САЙТ, WORDPRESS, CI/CD, CMS, VCS, REVISR, ELEMENTOR

## ABSTRACT

Certification work on the topic "Development of the template of the website of the department in accordance with the requirements with the possibilities of administration, moderation, configuration on WordPress". Maliarskyi Viktor. Ternopil Ivan Pulyuy`s National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, Group SPm-61, Ternopil, 2020. Pages - 72, sections - 4, imagines - 21.

The purpose and task of the certification work is to improve the website of the Department of Software Engineering, created on the WordPress system. In the course of the work, a study of the CI / CD methodology, the introduction of a version control system and modern methods of website development in general was conducted. As a result of the development, the website creation system was updated, the existing pages were adapted for mobile browsing, and a version control system website was installed.

Keywords: SITE, WORDPRESS, CI / CD, CMS, VCS, REVISR, ELEMENTOR

## ЗМІСТ

ВСТУП.....	4
1. АНАЛІЗ ОБЛАСТІ ДОСЛІДЖЕННЯ .....	6
1.1. Сучасні методи розробки веб-сайтів.....	6
1.1.1. Розробка вручну за допомогою HTML .....	7
1.1.2. Використання програмного забезпечення для розробки сайтів .....	9
1.1.3. Використання інструментальних систем CMS .....	12
1.1.4. Використання фреймворків.....	13
1.1.5. На хмарних платформах SaaS .....	16
2. ДОСЛІДЖЕННЯ ОБРАНОЇ ТЕМАТИКИ.....	19
2.1. Система керування контентом WordPress .....	19
2.2. Методологія CI/CD .....	28
2.3. Системи контролю версій.....	37
3. РОЗРОБКА СИСТЕМИ .....	44
3.1. Сайт кафедри програмної інженерії.....	44
3.3. Впровадження системи контролю версій .....	54
3.4. Встановлення нового функціоналу для створення сторінок та адаптація їх під мобільні пристрої.....	56
4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ .....	60
4.1. Охорона праці .....	60
4.2. Безпека в надзвичайних ситуаціях .....	63
ВИСНОВОК .....	67
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	69
ДОДАТКИ .....	71
ДОДАТОК А .....	73
ДОДАТОК Б.....	78
ДОДАТОК В.....	80

## ВСТУП

На сьогоднішній день основним засобом отримати будь-яку інформацію є інтернет. Кожен користувач глобальної мережі зараз має доступ практично до всіх знань, які людство здобувало тисячоліттями. Само-собой, кожен почав використовувати інтернет для різноманітних цілей. Хтось для саморозвитку, шукаючи все що йому цікаве за допомогою пошукових систем, хтось для розваги, читаючи різні статті, новини чи книги онлайн. Проте дуже важливою перевагою появи та масового використання мережі інтернет стала можливість поширення інформації про свій бізнес, свою організацію чи спільноту. Простішими словами, інтернет став місцем розміщення реклами та методом розповісти людям про власну справу.

Ось чому стала актуальною ідея створити веб-сайт кафедри Програмної Інженерії Тернопільського національного технічного університету імені Івана Пулюя. Це дозволяє абітурієнтам, студентам та їх батькам отримувати потрібну інформацію про кафедру, її працівників, діяльність, наукові програми, тощо. Відповідно, що більше людей дізнається про кафедру, то більша вірогідність розширити її студентський склад.

Також на веб-сайті кафедри ПІ є можливість розміщувати допоміжні документи для студентів, бланки та анкети для абітурієнтів, наукові роботи та статті для викладачів і працівників кафедри. Все це робить зручнішим навчання та роботу користувачів сайту.

Метою та завданням атестаційної роботи є удосконалення програмного забезпечення системи керування контентом типу WordPress в умовах обмежених інформаційних ресурсів. Адаптація системи до використання на різних девайсах та забезпечення можливості адміністрування, модерації та конфігурування її адміністраторами.

Об'єктом дослідження є створення безпечної системи для безперервного поширення інформації про кафедру Програмної Інженерії а також обміну

новинами, статтями та науковими роботами між працівниками кафедри та студентами.

Предметом дослідження є створення шаблону сайту за допомогою системи менеджменту контенту WordPress та налаштування його таким чином, щоб була можливість безпечно та швидко керувати його наповненням в умовах обмежених інформаційних ресурсів.

Під час виконання атестаційної роботи методом аналізу та збору інформації було виявлено, що існуюча система потребує адаптації до її коректного відображення та використання на мобільних девайсах а також забезпечення керування її наповненням без втрати інформації при умовах збоїв у серверній частині.

Тому новизною кінцевої системи стало внесення у неї деяких концептів методології розробки програмних продуктів з безперервною інтеграцією та безперервним впровадженням, англійською: *continuous integration / continuous deploy (CI/CD)*. А також новим є встановлення у систему функціоналу, що адаптує її до використання на мобільних пристроях.

В результаті отримано оновлену систему менеджменту контенту на WordPress з можливістю контролю версій, створення резервних копій для збереження інформації а також можливістю відміни виконаних змін над контентом веб-сайту. Було завантажено додатковий плагін, та перероблено існуючі сторінки та записи на них таким чином, що весь внутрішній контент, зображення та віджети коректно відображаються і на широкоформатних пристроях (комп'ютери, ноутбуки, планшети) і на мобільних пристроях.

Частина результатів досліджень для атестаційної роботи було опубліковано у тезах до наукової конференції «VIII Науково-технічна конференція, Інформаційні моделі, системи та технології», що проходила 9-10 грудня 2020 року у ТНТУ [1].

# 1. АНАЛІЗ ОБЛАСТІ ДОСЛІДЖЕННЯ

## 1.1. Сучасні методи розробки веб-сайтів

Сучасні тенденції розробки Веб-сайтів почали зароджуватися ще в 2004 році. Це була ера сміливих веб-сайтів, сайтів, спрямованих на громади [2]. В порівнянні з найпершими текстовими сайтами, тодішні сторінки почали виділятися різноманітною типографією тексту та блискучими градієнтами. Кути стали округлими, краї пом'якшилися, і веб-дизайн знову злетів зі швидкістю світла.

Веб-сайти стали більш функціональними і потребували більшої кількості інтерфейсів для належної роботи. Віджети були представлені всюди, щоб допомогти інтегрувати один сайт з іншим. Особливо це використовувалось, коли з'явилися соціальної мережі, сайти з зовнішніми посиланнями та особисті блоги.

Ця епоха також ознаменувалася збільшенням доступності веб-сайтів для простих людей. Такі розробки, як WordPress, Blogger та соціальні медіа, а також зручні вказівки щодо створення веб-сайту щодня допомагали людям створювати веб-сайти, не вивчаючи HTML або CSS.

Дизайн веб-сайтів неймовірно еволюціонував за останні кілька років, і одна річ, яка дала їм особливий поштовх, був смартфон. Було запроваджено дизайн мобільного веб-сайту, що дозволило людям правильно та зручно переглядати сайти на своїх телефонах.

Багато великих веб-сайтів створювали мобільні версії спеціально для смартфонів та планшетів, які швидко ставали популярними пристроями. В самому Інтернеті великі та швидкозростаючі сайти соціальних медіа створили більше віджетів, які користувачі могли додавати до своїх блогів, а інші веб-сайти створювали віджети, призначені для переходу на сайти соціальних медіа.

У теперішньому дизайні типографіка стала ще більш різноманітною, а дизайни на основі сітки швидко стають нормою.

Сьогодні дизайн веб-сайтів - це величезний бізнес. На мою думку, дизайн став більш складний, але менш захаращений. Ранні веб-сайти були важчими для



навігації. У наш час достатньо добре розробленого веб-сайту, щоб Ваш бізнес досяг успіху.

На сьогоднішній день існує 5 основних методів веб-розробки [2]

- Розробка вручну за допомогою HTML
- Використання програмного забезпечення для розробки сайтів
- Використання інструментальних систем, таких як CMS
- Використання популярних на даний час фреймворків
- На хмарних платформах SaaS

### **1.1.1. Розробка вручну за допомогою HTML**

З появою стандарту розмітки сторінок HTML цей метод був найпоширенішим. Основною програмою для розробки став звичайний Блокнот, що є на будь-якому комп'ютері. Але цей метод має суттєві недоліки, він є дуже трудомісткий. І крім того, створити звичайний веб-сайт без CSS, JavaScript та інших мов програмування досить складно.

З'явилася нова специфікація мови розмітки HTML5, яка досі використовується для створення веб-сторінок [4]. HTML5 стала найкращою мовою для структурування та представлення контенту в Інтернеті. Вона може бути використаною для розробки інтерактивних веб-сайтів для різноманітних платформ і браузерів. Також інструменти для розробки мобільних додатків на основі HTML5 привабливі для розробників, які націлені на мобільний користувачів.

Попит на HTML5 зростає, оскільки він стає передовою в розробці eLearning, HTML5 - це поточна версія мови розмітки гіпертексту, за допомогою якої професіонали та експерти eLearning можуть публікувати та доставляти вміст на смартфони.

HTML5 допомагає вбудовувати відео та аудіо, картинки з високим розширенням, діаграми, анімацію та багато інших типів насиченого вмісту без використання плагінів чи сторонніх програм, оскільки функціонал тепер

вбудований у браузер, і Ви можете отримати доступ до програм HTML5 без завантаження їх на пристрої.

HTML5 пропонує більш описову та вдосконалену семантику, він має більш насичені медіа-елементи, Ви можете використовувати чистіший, акуратніший код, це дозволяє видалити більшість тегів div і замінити їх семантичними елементами HTML5, також Ви можете збільшити семантичне значення веб-сторінки, оскільки код дуже стандартизований за допомогою елементів HTML5.

Можна помітити більшу узгодженість у кодуванні сторінок на HTML5, на відміну від тих, що написані на HTML4. Це полегшить дизайнерам та розробникам розуміння того, як веб-сторінка структурована.

З'явилися нові функції та стандарти. Після того, як ви виявите доступні функції сучасних браузерів, ви зможете скористатися цими функціями у вашій програмі. Ви можете спростити програму за допомогою простих інтерфейсних інструментів, інструментів перетягування та перегляду, вікі та інших корисних елементів.

За допомогою HTML 5 під час побудови веб-сайту ви можете вказати файли, які браузер повинен кешувати. Тобто, коли ви оновлюєте сторінку в автономному режимі, сторінка завантажується правильно. Цей тип кешування має багато переваг, таких як перегляд в автономному режимі, набагато швидше завантаження файлів і зменшення навантаження на сервер.

Файли cookie використовуються для відстеження унікальних даних користувачів протягом багатьох років, але вони мають серйозні недоліки. Найбільший недолік полягає в тому, що всі дані ваших файлів cookie додаються до кожного заголовка HTTP-запиту. Найкраща практика - зменшити розмір файлів cookie. Ми можемо покращити це за допомогою HTML5, використовуючи сховище сеансів та локальне сховище (два різні сховища в HTML5) замість файлів cookie. Це не постійна база даних, але вона дозволяє тимчасово зберігати структуровані дані.

Також доволі корисним додатком HTML5 є геолокація. За допомогою неї ви можете дізнатись, де ви знаходитесь, і можете поділитися цією інформацією з

людьми. Існує багато способів щоб з'ясувати вашу локацію, наприклад, ваша IP-адреса, бездротове з'єднання з мережею, до якої мобільної станції під'єднаний ваш телефон або окремий модуль GPS, який обчислює широту та довготу на основі інформації, що надсилається супутниками.

Нові API геолокації HTML5 роблять розташування, виявлене за допомогою GPS або іншими методами, безпосередньо доступним для будь-якого сумісного з HTML5 додатку, що базується на веб-браузері. Користувачі HTML5 мають можливість встановити Drag&Drop API (Інтерфейс програмного додатку), тобто додаткову функцію інтерфейсу, що дозволяє обирати елементи перетягуванням їх у кошик чи меню.

Користувачам не потрібно завантажувати будь-яке програмне забезпечення для перегляду мультимедійного вмісту, адже HTML5 дозволяє розробникам легко створювати ігри та інші види додатків, зручні для мобільних пристроїв, а також існує безліч анімацій, які сумісні з HTML5.

Написанню коду HTML5 легше навчитися порівняно з іншими технологіями. Компанії можуть заощадити гроші, якщо розробляють додатки, які працюють у всіх операційних системах, а не мають власні програми. Ви можете писати та використовувати код як посилання для проектів, що базуються на Android, Windows або iOS.

HTML5 дозволяє розробляти програми, які адаптуються до різної роздільної здатності, розмірів екрану, пропорцій та вимог. Можуть бути створені розширені функції для сучасних пристроїв, таких як планшети чи смартфони.

### **1.1.2. Використання програмного забезпечення для розробки сайтів**

Програмне забезпечення для веб-дизайну надає користувачам інструменти для створення, редагування та оновлення веб-сторінок. Веб-дизайнери використовуватимуть ці програми для створення макета та вмісту веб-сторінки за допомогою редагування HTML або редактора WYSIWYG. Компанії, які працюють

без використання конструкторів веб-сайтів, вимагатимуть від своїх дизайнерів побудови та управління фреймворків структури HTML та CSS для свого веб-сайту.

Ці продукти включають функції для редагування коду та візуалізації проектів. Багато з них також мають вбудовані фреймворки та шаблони, які допомагають у створенні базової структури веб-сайту, а також бібліотеки елементів із шаблонами або шорткодами, щоб заощадити час під час проектування.

Програмне забезпечення для веб-дизайну забезпечує більш незалежну платформу з можливістю її налаштування, ніж платформи для розробки веб-сайтів, але має деякі подібні функції. Програмне забезпечення для веб-дизайну, як правило, дозволяє імпортувати код із таких інструментів, як текстові редактори, або візуальних елементів із таких інструментів дизайну, як інструменти векторної графіки.

Щоб претендувати на включення до категорії веб-дизайну, продукт повинен:

- Надавати доступ для редагування та / або експорту HTML та CSS
- Мати шаблони або веб-фреймворки, які можна налаштувати
- Надавати інструменти для візуалізації інтерфейсного дизайнерського коду
- Мати бібліотеки елементів або шорткоди
- Володіти можливостями веб-публікації

В загальному, до інструментів для розробки HTML сайтів відносяться програми, які мають у собі візуальні редактори вмісту (на основі дизайну), програми, які можуть автоматично генерувати код сторінки (WYSIWYG засоби), та програми, які дозволяють автоматизувати написання коду з можливістю його редагування.

Найпопулярнішими редакторами на основі дизайну є:

Adobe DreamWeaver - є однією з найкращих флагманських програм, що коли-небудь запускалися Adobe. Загалом, це дуже корисна програма для розробки веб-сайтів та програмного забезпечення. Вона організовує роботу з HTML або CSS та робить її набагато простішою, дозволяючи легко працювати з файлами. Це відбувається завдяки високорозвиненому загальному інтерфейсу користувача.

SharePoint Designer - це програма для веб-дизайну та дизайну додатків, яка використовується для створення та налаштування сайтів і програм SharePoint. За допомогою SharePoint Designer Ви можете створювати сторінки, багаті на дані, створювати потужні рішення з підтримкою робочого циклу та розробляти зовнішній вигляд вашого веб-сайту. SharePoint Designer забезпечує унікальний досвід створення сайтів, надаючи одну платформу, де ви можете створити сайт; налаштувати компоненти, що складають сайт; розробити логіку сайту навколо бізнес-процесу; і розгорнути сайт як пакетне рішення. Ви можете зробити все це без написання коду.

Розглянемо редактори на основі коду:

Macromedia HomeSite - це інструмент HTML-дизайну для професійних веб-розробників. Він дає вам переваги підвищення продуктивності користувачів, вдосконаленого управління проектами, розширеного розгортання сайту та додаткової підтримки новітніх веб-технологій. Це простий редактор веб-розробок, який містить лише код. Розширені функції кодування дозволяють миттєво створювати та модифікувати теги HTML, CFML, JSP та XHTML, тоді як покращені інструменти продуктивності дозволяють легше перевіряти, повторно використовувати, орієнтуватися та формувати код. Підлаштуйте Macromedia HomeSite відповідно до ваших потреб, розширивши його функціональність та налаштувавши інтерфейс. Особливості включають розбірливий код для поліпшення читабельності сторінки, автозавершення визначених користувачем рядків та редагування за допомогою розділеного вікна.

HotDog - це надійний, гнучкий та зручний редактор HTML. Починаючи з 1994 року, мільйони користувачів у всьому світі користувалися першокласною мережею підтримки та доступом до інших продуктів та послуг розробника. Цей редактор поставить вас на правильний шлях до створення професійного веб-сайту. Він має вбудований інструмент управління веб-сайтами, відображувач навігації, лист властивостей HTML та вбудовану утиліту FTP. Ви також можете імпортувати, створювати та редагувати таблиці стилів CSS.

AcеHTML - це чудовий редактор HTML, який пропонує сумісність із широким колом мов. Ця програма має панель налаштування схожу на набір Microsoft Office, і серед мов, які вона підтримує та може ними керувати, можна виділити Perl, XML, WML, ASP, PHP, ColdFusion, CSS та JavaScript.

### **1.1.3. Використання інструментальних систем CMS**

Система управління контентом (Content Management System) - це програмна платформа, яка дозволяє своїм користувачам створювати, редагувати, архівувати веб-сайти. А також співпрацювати, звітувати, публікувати, розповсюджувати та інформувати інших розробників. Їх графічний інтерфейс користувача (GUI) робить взаємодію з базою даних веб-сайту зручною [5].

CMS веб-сайти використовують HTML (мова розмітки гіпертексту) та CSS (каскадні таблиці стилів) для створення та дизайну своїх сторінок. Вони є двома важливими основними компонентами для створення веб-сторінок. HTML забезпечує структуру сторінки, CSS - візуальний та звуковий макет.

Система управління контентом дозволяє користувачам без будь-яких знань кодування змінювати, модифікувати та редагувати вміст веб-сайтів за допомогою інтерфейсу WYSIWYG, скорочення від "те, що ти бачиш, те і отримуєш". Дані, введені в програмне забезпечення управління контентом, зберігаються в базі даних, яка відображає веб-сторінку за допомогою шаблону. Потім CSS цієї сторінки може контролювати її вигляд.

CMS дозволяє користувачам керувати вмістом із внутрішнього інтерфейсу користувача або інформаційної панелі. Існує велика кількість програмного забезпечення для управління контентом, яке можна встановити в один клік. Це полегшує роботу нетехнічного маркетолога у використанні та навігації. Більшість програм CMS із найвищим рейтингом для менших стартапів є відкритими та безкоштовними.

Це означає, що вам не доведеться добре володіти іншими компонентами що входять до CMS, а саме JavaScript (мова програмування, яка зазвичай використовується у веб-розробці), HTML (мова гіпертекстової розмітки), CSS (каскадні таблиці стилів), PHP (Персональна домашня сторінка або процесор гіпертексту), та MySQL (система управління реляційними базами даних із відкритим кодом, заснована на структурі мови запитів SQL).

Створення веб-сайту за допомогою системи управління контентом є аналогом гри з пластиковою цеглою Legos. Ви можете вибрати, які цеглинки найкраще підходять для побудови вашого сайту. Це дозволяє писати текст та вставляти картинки та графіку безпосередньо з панелі управління.

Веб-сайти CMS побудовані з базами даних, подібними до електронних таблиць Excel, із захищеним та простим у користуванні інтерфейсом [6]. Завдяки новішим ітераціям більшість CMS управляються та постійно оновлюються в міру розвитку Інтернету.

Варіантів нових платформ веб-побудови CMS багато. Традиційним варіантом є WordPress. Wordpress - це система з відкритим кодом із багатогранними функціями, шаблонами, темами та плагінами, тому не потрібно витратити час на встановлення та створення веб-сайтів, що працюють у реальному часі. WordPress - це фактично програмне забезпечення, що зараз використовують приблизно 75 мільйонів веб-сайтів. В даний час на нього припадає понад чверть усіх веб-сайтів.

#### **1.1.4. Використання фреймворків**

Веб-фреймворк - це програмний продукт, який призначений для підтримки розробки веб-додатків, включаючи веб-сервіси, веб-ресурси та веб-API, проте він не забезпечує готових рішень, а лише надає так званий каркас для подальшої розробки сторінок [6]. Веб-фреймворки забезпечують стандартний спосіб створення та розгортання веб-додатків у Всесвітній мережі. Вони спрямовані на автоматизацію накладних витрат, пов'язаних із загальними діями, що виконуються у веб-розробці. Наприклад, багато веб-фреймворків надають бібліотеки для

доступу до баз даних, шаблонів та управління сеансами, і вони часто сприяють повторному використанню коду. Хоч вони часто націлені на розробку динамічних веб-сайтів, вони також застосовні до статичних веб-сайтів.

Веб-фреймворк підтримує створення, розробку та публікацію веб-додатків та веб-сайтів. Він може включати в себе веб-сервіси, API та інші ресурси. Веб-фреймворки - це програмні фреймворки, які пропонують стандартний і доступний спосіб створення та розробки веб-додатків.

Хоч додатки з підтримкою Інтернету і є зараз звичними явищами і ними користуються щодня більшість людей, підключених до Інтернету, але щодо розвитку методологій та інструментів вони все ще перебувають у зародковому стані. Ось чому так важливо мати веб-фреймворк, який би служив системою підтримки для розробки та розгортання таких програм.

Веб-фреймворк - це бібліотека коду, яка робить веб-розробку швидшою та простішою, надаючи основні шаблони для створення надійних, масштабованих та адаптивних веб-додатків. Веб-фреймворки існують, щоб спростити розробнику створення веб-програми. Подумайте про це як про спосіб створення ярликів, які можуть запобігти рутинному і повторюваному пошуку файлів у пам'яті комп'ютера. Наприклад, можливо, ви написали код для обробки перевірки даних для вашої веб-програми. Щоб запобігти необхідності переписувати цей код з нуля кожного разу, коли ви створюєте веб-сайт або веб-службу, веб-фреймворк робить це за вас.

Веб-фреймворк відрізняється від веб-сервера тим, що веб-сервер насправді запускає веб-додаток, тоді як веб-фреймворк більше нагадує віртуальну базу даних або бібліотеку, яка допомагає пришвидшити процес розробки та написання сайтів.

Фреймворки надають функціональність через власний код або через розширення для створення повсякденних операцій, необхідних для запуску веб-додатків. Ці операції включають:

- Маршрутизація URL-адрес
- Управління та перевірка форми вводу



- Налаштування HTML, XML, JSON та інших продуктів із механізмом шаблонування
- Конфігурація з'єднання з базою даних та безпосередня маніпуляція даними за допомогою об'єктно-реляційного мапування (ORM)
- Веб-безпека від міжсайтової підробки запитів (CSRF), SQL Injection, міжсайтових сценаріїв (XSS) та інших частих зловмисних атак
- Збереження та відтворення сесій

Існує кілька веб-фреймворків, розроблених для полегшення кодування та розробки додатків для тих, хто працює в даній галузі. Кілька найбільш популярних та успішних веб-фреймворків включають:

- Django
- Ruby on Rails
- Angular
- React
- Vue
- Laravel

Ці фреймворки допомагають розробці на таких мовах, як HTML, Ruby, PHP, JavaScript та Python, коли мова йде про створення веб-додатків та веб-сервісів. Існує безліч речей, які можна зробити за допомогою веб-фреймворку, наприклад, генерація HTML, формування шаблонів за замовчуванням та можливість доступу до бібліотеки інформації про кодування, щоб зробити код простим та автоматизованим кожного разу, коли розробник робить новий веб-додаток.

Неодмінно рекомендується використовувати фреймворки, щоб допомогти впорядкувати та пришвидшити процес розробки, а популярні фреймворки, такі як Ruby on Rails, розроблені, щоб стати потужним посилюючим фактором для проекту. Коли розробник вирішує використовувати веб-фреймворк для написання коду, слід пам'ятати декілька речей.

Починаючи з веб-фреймворку, слід врахувати:

Взаємодія з базами даних - Багато веб-фреймворків мають можливість застосовувати взаємодію з базами даних при використанні повного пакету.

Генерація HTML - фреймворки мають можливість генерувати HTML-код за допомогою шаблонів, це може полегшити розробникам всі процеси якщо правильно встановити всі налаштування та вказати всі потрібні значення.

Маршрутизація - більшість фреймворків надають можливість зіставити URL-адресу, яку запитує користувач, з кодом, який відповідає їй.

Веб-фреймворки допомагають виконувати різноманітні завдання, починаючи від створення шаблонів та доступу до бази даних, до управління сесіями та повторного використання коду. Це великий актив для створення динамічних веб-сайтів, але може бути використаний також і для розробки статичних веб-сайтів.

### **1.1.5. На хмарних платформах SaaS**

Software as a service (Програмне забезпечення як послуга) SaaS - це модель розповсюдження програмного забезпечення, при якій сторонній провайдер розміщує додатки та робить їх доступними для клієнтів через Інтернет [8]. SaaS - це одна з трьох основних категорій хмарних обчислень, разом з інфраструктурою як послугою та платформою як послугою (IaaS та PaaS).

SaaS тісно пов'язані з Application Service Provider (ASP) та моделями обчислювальних програм на вимогу. Модель розміщення та управління додатками SaaS подібна до ASP, де провайдер розміщує програмне забезпечення замовника та доставляє його затвердженим кінцевим користувачам через Інтернет.

У програмі на замовлення SaaS програмне забезпечення постачальник надає клієнтам мережевий доступ до єдиної копії програми, яку постачальник створив спеціально для розповсюдження SaaS. Вихідний код програми однаковий для всіх клієнтів, і коли імплементуються нові функції або атрибути, вони розгортаються для всіх клієнтів. Залежно від угоди про рівень обслуговування (SLA), дані клієнта для кожної моделі можуть зберігатися локально, у хмарі або і так і так.

Організації можуть інтегрувати додатки SaaS з іншим програмним забезпеченням, використовуючи інтерфейси прикладного програмування (API). Наприклад, бізнес може писати власні програмні засоби та використовувати API провайдера SaaS для інтеграції цих інструментів із пропозицією SaaS.

Існують додатки SaaS для основних бізнес-технологій, таких як електронна пошта, управління продажами, управління взаємовідносинами з клієнтами (CRM), управління фінансами, управління людськими ресурсами (HRM), виставлення рахунків та співпраця. Серед провідних постачальників SaaS - Salesforce, Oracle, SAP, Intuit та Microsoft.

#### Переваги:

SaaS усуває необхідність організацій встановлювати та запускати програми на своїх комп'ютерах або у власних центрах обробки даних. Це виключає витрати на придбання, забезпечення та обслуговування обладнання, а також ліцензування, встановлення та підтримку програмного забезпечення. Інші переваги моделі SaaS включають:

Гнучкі платежі - замість того, щоб купувати програмне забезпечення для встановлення або додаткове обладнання для його підтримки, клієнти підписуються на пропозицію SaaS. Як правило, вони платять за цю послугу щомісяця, використовуючи модель оплати на час. Перехід витрат на постійні операційні витрати дозволяє багатьом підприємствам здійснювати більш якісне та передбачуване бюджетування. Користувачі також можуть припинити підписку на SaaS у будь-який час, щоб зупинити ці постійні витрати.

Масштабоване використання - хмарні сервіси, такі як SaaS, пропонують високу вертикальну масштабованість, що надає клієнтам доступ до більшої або меншої кількості послуг або функцій на вимогу.

Автоматичні оновлення - замість того, щоб купувати нове програмне забезпечення, клієнти можуть покладатися на постачальника послуг SaaS для автоматичного виконання оновлення та управління виправленнями. Це ще більше зменшує навантаження на власний ІТ-персонал.

Доступність та постійність - оскільки програми SaaS доставляються через Інтернет, користувачі можуть отримати до них доступ із будь-якого пристрою та місця, де працює Інтернет.

#### Недоліки:

SaaS також має деякі потенційні недоліки. Підприємства повинні покладатися на сторонніх постачальників, які надаватимуть програмне забезпечення, підтримуватимуть його і запускатимуть, відстежуватимуть та повідомлятимуть точні рахунки та сприятимуть створенню безпечного середовища для даних бізнесу. Проблеми можуть виникати, коли постачальники стикаються з перебоями в роботі послуг, вносять небажані зміни в пропозиції послуг або порушують безпеку, і все це може суттєво вплинути на здатність клієнтів використовувати пропозиції SaaS. Для попередження цих проблем клієнти повинні підписувати угоду про надання послуг зі своїм постачальником SaaS та переконуватися, що вона виконується

## 2. ДОСЛІДЖЕННЯ ОБРАНОЇ ТЕМАТИКИ

### 2.1. Система керування контентом WordPress

Оскільки темою атестаційної роботи було вибрано розробку сайту кафедри саме на базі WordPress, в першу чергу потрібно дослідити всі аспекти даної системи. WordPress - це система управління контентом (CMS), за допомогою якої було розроблено понад 35% веб-сайтів у всьому світі, і ця кількість продовжує зростати.

CMS - це веб-додаток, що дозволяє власникам веб-сайтів, редакторам та авторам керувати своїми веб-сайтами та публікувати вміст без будь-яких знань програмування.

WordPress використовує PHP та MySQL, що підтримується майже всіма веб-хостами. Використання спеціалізованого планування хостингу WordPress може також забезпечити вам переваги щодо швидкості, продуктивності та надійності.

CMS часто використовується для ведення блогів, але сайт WordPress можна легко перетворити на магазин електронної комерції, портфоліо, веб-сайт газет або що-небудь інше, що потрібно користувачеві.

Однією з найкращих речей WordPress є те, що він має інтуїтивно зрозумілий та зручний інтерфейс. Якщо ви знаєте, як користуватися Microsoft Word, ви зможете легко створювати та публікувати вміст.

Найкраще те, що WordPress є відкритим і безкоштовним для всіх. Це дозволяє мільйонам людей по всьому світу створювати якісні сучасні веб-сайти.

Подорож WordPress розпочалося в 2003 році, коли двоє розробників, Метт Малленвег та Майк Літл, почали створювати нову платформу для ведення блогів поверх вже не активного програмного забезпечення b2/cafeblog [9]. Незабаром після того, як від цього проекту відмовились, вони вирішили закінчити його та продовжувати розробку на власних умовах.

Першу версію WordPress 1.0 вони випустили в січні 2004 року. Вона виглядала зовсім не так, як багатofункціональна система управління вмістом, яку

ми сьогодні знаємо як WordPress. Однак він уже мав більшість основних функцій, якими ми користуємось і сьогодні, такі як редактор WordPress, простий процес встановлення, власні та постійні посилання, система управління користувачами, можливості модерування коментарів.

З 2004 року WordPress пережив неймовірну трансформацію. На сьогодні проект WordPress розробляється, управляється та підтримується спеціальною спільнотою з відкритим кодом, що складається з тисяч членів світу. Вони працюють віддалено, багато з них є волонтерами та особисто зустрічаються лише на розподіленій конференції WordPress під назвою WordCamp. WordPress все ще швидко розвивається, і нова версія виходить кожні 2-3 місяці. Кожна версія додає на платформу нові функції та оновлення системи безпеки.

Коли ви введете слово WordPress в пошуковий рядок в Google або інших пошукових системах, ви знайдете два пов'язані веб-сайти: WordPress.com та WordPress.org. Це може заплутати, оскільки обидва сайти мають власні цілі та базу користувачів.

WordPress.org - це веб-сайт, який вам потрібен, якщо ви хочете запуснути власний сайт WordPress. Тут ви можете завантажити платформу та знайти безліч безкоштовних плагінів та тем. Щоб розмістити свій власний веб-сайт, вам також потрібно зареєструвати домен і зареєструватися на веб-хостингу.

WordPress.com - це веб-хостингова компанія, що використовує систему управління вмістом WordPress для всіх блогів, які вона розміщує [10]. Користувачі, які підписалися на послугу, отримують власні блоги з типом домену myblog.wordpress.com (однак власні домени також доступні як платна функція). Хостинг за замовчуванням безкоштовний, хоча він має деякі обмеження.

Загалом різницю між WordPress.com та WordPress.org можна описати за такими пунктами [11]:

Хостинг. WordPress.com - це комплексний пакет з безкоштовним хостингом, а щоб використовувати WordPress.org потрібно подбати про те, щоб розмістити свій сайт на сторонній хостинговій компанії, такій як Namecheap.

Вартість. WordPress.com є безкоштовним, поки вам не потрібні додаткові функції, які зазвичай входять до складу WordPress.org. А він в свою чергу включає тисячі безкоштовних аддонів, ви просто платите за реєстрацію домену на хостингу.

Складність. Навігація користувальницьким інтерфейсом є інтуїтивно зрозумілою, і використання функцій є простим в обох версіях. Технічно WordPress.com має перевагу щодо простоти використання, оскільки у ньому подбали про встановлення та розміщення.

Дизайн та верстка. Обидві платформи дозволяють оформляти ваші сторінки стандартними темами (шаблонами) зі сховища тем. Хоча скрипт WordPress.org пропонує безмежні можливості для налаштування та завантаження нових тем, а користувачі WordPress.com повинні оновлювати старі версії вручну, щоб налаштувати теми своїх сайтів.

Гнучкість. Що стосується тем та плагінів, WordPress.org надає доступ до будь-якого плагіна, безкоштовного або платного, щоб максимізувати потенціал CMS. Тоді як WordPress.com включає вбудовані функції, подібні до плагінів, вам доведеться перейти на платну версію, щоб додати плагіни з каталогу.

Електронна комерція. Монетизувати свій сайт простіше завдяки WordPress.org, що дозволяє розміщуватися самостійно. Ви не можете створити Інтернет-магазин на WordPress.com.

Одне важливе зауваження щодо безкоштовної версії WordPress.com: ваш веб-сайт завантажений з фірмовою символікою WordPress, а банери рекламують, що ви не будете заробляти гроші. Знову ж таки, оновлення вашої версії їх видалить. Самостійні інсталяції WordPress.org пропонують повну свободу брендування з самого початку. Якщо ви хочете отримувати дохід від реклами на своїх сторінках, ви можете додати їх за допомогою плагінів. Продавайте скільки завгодно оголошень, а головне, зберігайте 100% заробленого. І навпаки, продаж реклами на WordPress.org заборонений, якщо ви не перейдете на платний план.

На відміну від власної платформи WordPress.com не надає сайтам членства з різним рівнем доступу, обмежуючи вашу можливість отримувати дохід від користувачів вашого сайту.

Безпека. Користувачам WordPress.com не доводиться турбуватися про безпеку сайту, про це подбає вся підтримка та обслуговування їхнього сайту. За допомогою WordPress.org ви несете повну відповідальність за захист свого веб-сайту, включаючи оновлення, резервні копії та головне - оптимізацію.

Спільнота розробників. Що стосується ресурсів, обидві платформи WordPress мають універсальну підтримку, включаючи велику кількість розробників. Оскільки WordPress керується спільнотою, спільнота волонтерів-розробників пропонує керівництво користувачам сайту та подальший розвиток платформи WordPress.

Допоки ви можете створювати веб-сайти із безкоштовною версією WordPress, ми не можемо недооцінювати значення плагінів, щоб максимізувати потенціал вашого веб-сайту. Крім того, якщо ви хочете зберегти весь свій прибуток, вибір WordPress.com надасть вам повний контроль над вашими веб-сайтами та доходом. Ви завжди можете перейти з WordPress.com на WordPress.org або будь-яку іншу платформу.

WordPress - надзвичайно універсальний інструмент, який можна використовувати для створення декількох різних типів веб-сайтів. Можливостей, здається, безмежно, але можливо перерахували найпоширеніші способи використання.

## Блоги

WordPress розпочався як інструмент ведення блогу, і це досі залишається одним із найпоширеніших видів використання веб-платформи. Ви можете починати з малого з мінімальними витратами, наприклад, з початкового плану Hostinger. Потім, коли ваш блог стає дедалі більше доступним, ви можете перейти на більш розширені варіанти.



## Портфоліо

Художники, фотографи, музиканти, письменники та інші творчі професіонали та фрілансери часто використовують WordPress для створення своїх онлайн-портфоліо. На WordPress.org та інших сторонніх сайтах доступні кілька тем портфоліо, які включають усі функції, які вам можуть знадобитися, такі як галереї зображень, лайтбокси та сітки портфоліо.

## Сайти бізнесу

Якщо у вас невеликий бізнес, ви також знайдете WordPress як хороше рішення для вашого веб-сайту. Хоч WordPress надзвичайно популярний серед малого та середнього бізнесу, він все частіше використовується і великими корпораціями.

## Благодійні сайти

Оскільки WordPress має низьку вартість обслуговування та відносно рівну статистику навчання, не дивно, що це також улюблена платформа благодійних організацій та церков. Завдяки WordPress, сьогодні все більше некомерційних організацій є присутніми в Інтернеті, ніж будь-коли раніше.

## Інтернет-магазини

За допомогою WordPress ви також можете створити інтернет-магазин без будь-яких знань програмування. WooCommerce, найпопулярніший плагін електронної комерції для WordPress, дозволяє додавати на ваш сайт такі важливі функції, як кошики для покупок, система управління замовленнями, сторінки оформлення замовлення, повернення коштів одним кліком та інші доповнення.

## Популярні сайти, що використовують WordPress

Хоча WordPress в основному відомий як платформа для блогерів та малого бізнесу, він також використовується багатьма відомими веб-сайтами.

## The New Yorker

Вся структура популярного журналу The New Yorker працює на основі WordPress.



Рисунок 2.1 – Використання WordPress сайтом The New Yorker

## TechCrunch

TechCrunch - це веб-сайт для людей, які цікавляться передовими технологіями та новинами про стартап - можливо, не випадково вони обрали WordPress як свою платформу.

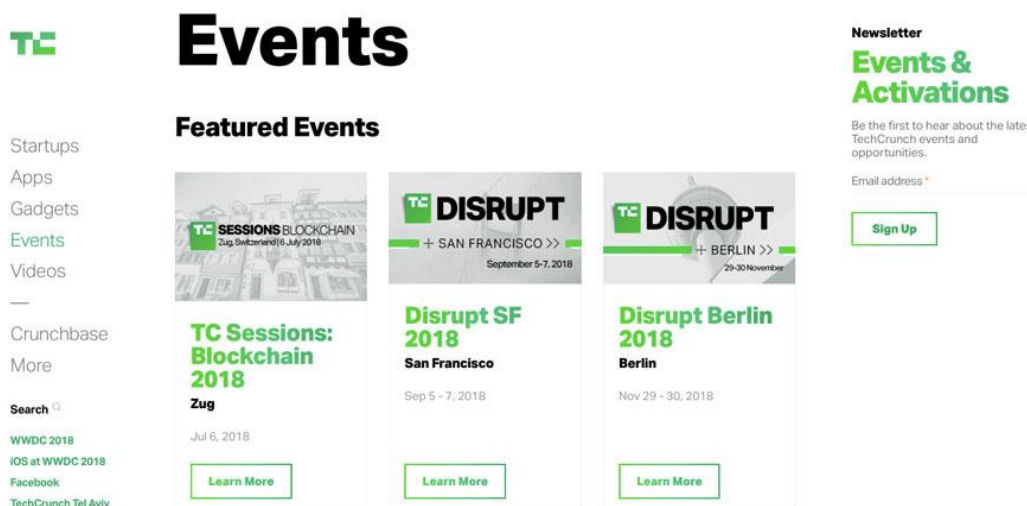


Рисунок 2.2 – Використання WordPress сайтом Tech Crunch

## Sony Music

Веб-сайт Sony Music - чудовий приклад успішних комерційних брендів, які використовують WordPress.

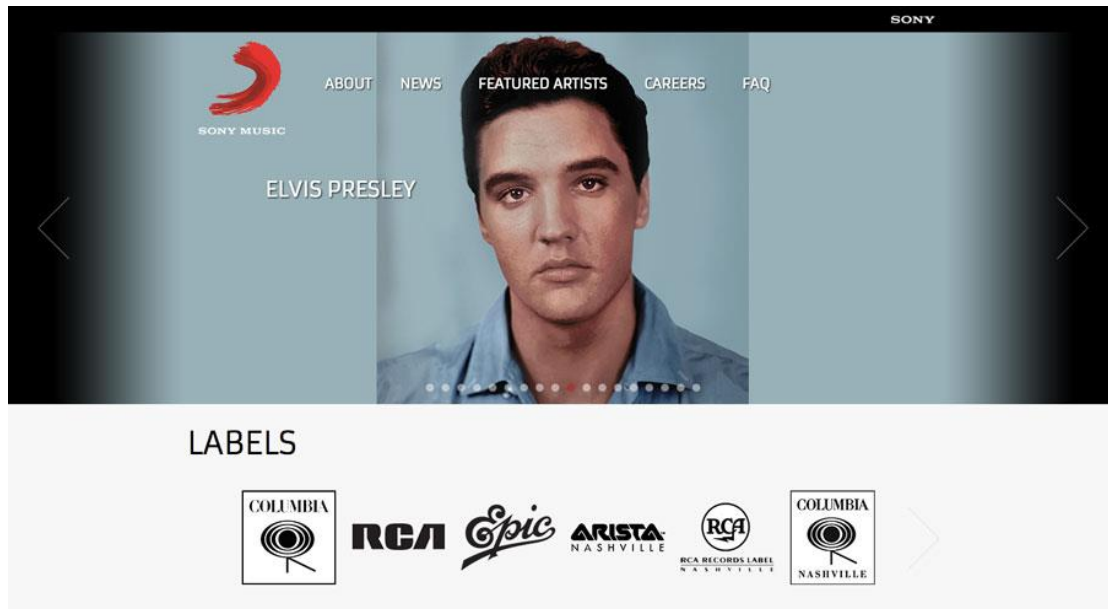


Рисунок 2.3 – Використання WordPress веб-сайтом Sony Music

## Білий Дім

Офіційний веб-сайт Білого Дому також працює на WordPress.



Рисунок 2.4 - Веб-сайт Білого дому з використанням WordPress

## Співачка Бейонсе

WordPress неймовірно популярний серед художників, фотографів, музикантів та інших креативів, навіть таких зірок, як Бейонсе. WordPress дозволяє їй забезпечити захоплюючий User Experience своїм шанувальникам.

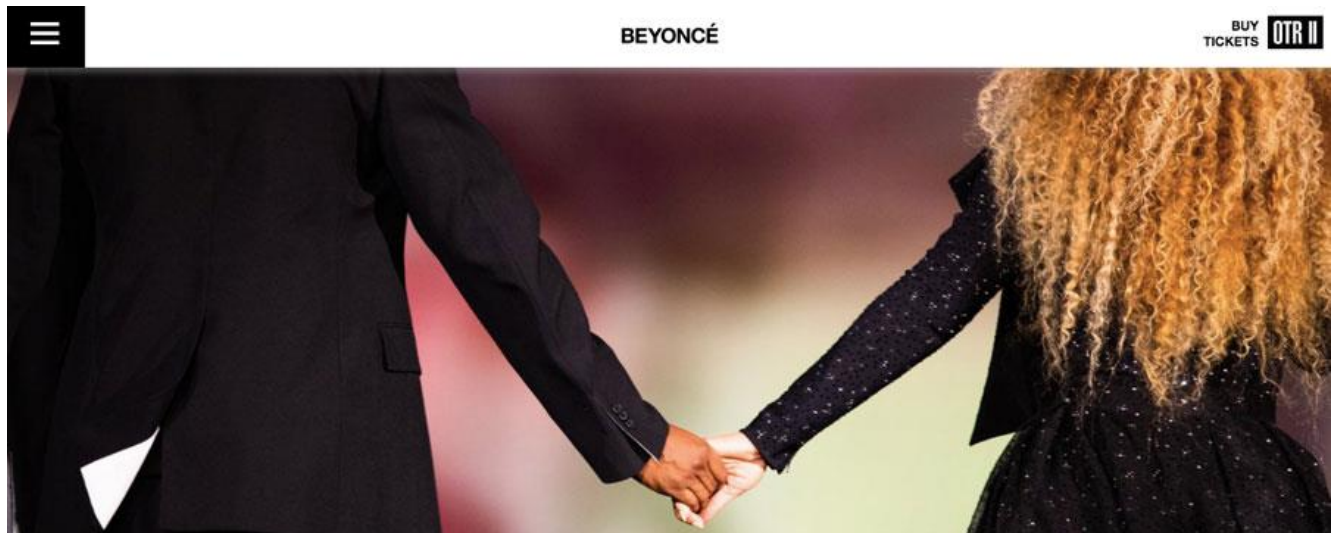


Рисунок 2.5 - Веб-сайт співачки Бейонсе з використанням WordPress

## Плюси та мінуси WordPress

WordPress має дуже багато переваг, і це одна з найпопулярніших платформ для тих, хто хоче запустити веб-сайт без необхідності знання програмування. Крім того, це хороший вибір для програмістів, оскільки це дуже адаптивний інструмент. Ось список деяких основних моментів, про які слід пам'ятати.

### Плюси

Невисока вартість - Вам потрібно заплатити лише за домен та послугу веб-хостингу. Програмне забезпечення WordPress та багато плагінів та тем безкоштовні.

Простий процес встановлення та оновлення - на відміну від багатьох інших CMS, WordPress не потребує конфігурацій, і ви можете оновити його одним натисканням миші.

Просте управління - Вам не потрібні знання програмування для виконання повсякденних завдань, таких як написання та редагування публікацій, завантаження та редагування зображень, управління користувачами, додавання меню та встановлення плагінів та тем.

Спеціальний дизайн - завдяки тисячам готових до використання тем WordPress Ви можете легко створити власний дизайн, який відповідає вашій ніші. Наприклад, є конкретні теми для ресторанів, медичних працівників, малого бізнесу, продовольчих блогерів, тощо.

Спеціальна функціональність - Ви можете використовувати плагіни для розширення функцій WordPress за замовчуванням. В основному, ви можете знайти плагін для кожного конкретного завдання - від оптимізації пошукової системи до закріплення подій.

Корисна спільнота - WordPress має всесвітню спільноту та надзвичайно корисний форум підтримки. Якщо ви чогось не знаєте або маєте якісь проблеми, можете швидко знайти допомогу.

Система з відкритим кодом - Вам ніколи не доведеться платити за програмне забезпечення WordPress, що є перевагою над альтернативними системами управління контентом.

## Мінуси

Проблеми з безпекою - оскільки WordPress забезпечує більше 35% Інтернету, він також часто націлений на хакерів. Однак, якщо ви встановите плагін безпеки, ви зможете зменшити ризик.

Вміст сторонніх розробників - оскільки більшість плагінів та тем створюються сторонніми розробниками, вони іноді глючать. Перш ніж встановлювати новий плагін або тему, завжди читайте опис та відгуки. Якщо ви все ще не впевнені, попросіть підтримки у спільноти.

Час завантаження сторінки - якщо у вас забагато плагінів, ваш сайт може стати повільним. Встановлення плагіна кешування, як правило, може вирішити цю проблему.

## 2.2. Методологія CI/CD

Зараз людство на межі прориву нових технологій. Зростаюча кількість компаній постачає і впроваджує своє програмне забезпечення за лічені хвилини. Часто команди намагаються передати програмне забезпечення в руки замовника через відсутність послідовності та надмірну кількість ручних налаштувань. Безперервна інтеграція (CI) та безперервна доставка (CD) розгортають програмне забезпечення до робочого середовища зі швидкістю, безпекою та надійністю [12].

Безперервна інтеграція (CI) - це практика розробки, яка вимагає від розробників інтегрувати код у спільне сховище кілька разів на день. Потім кожна інтеграція перевіряється автоматизованою збіркою, що дозволяє командам виявляти проблеми на ранній стадії. Регулярно інтегруючи, ви можете швидко виявляти помилки та легше знаходити їх.

За допомогою CI розробник практикує інтеграцію коду безперервно оновлюючи продукт паралельно з рештою команди. Інтеграція відбувається після операції “git push”, як правило, для основної гілки системи контролю версій. Потім на виділеному сервері автоматизований процес створює додаток і запускає набір тестів, щоб підтвердити, що найновіший код інтегрується з тим, що в даний час вже знаходиться у головній гілці.

Якщо ви робите CI і з якихось причин інтеграція не вдається, це означає, що зламана збірка стає найвищим пріоритетом для виправлення, перш ніж продовжувати додавати більше функцій. Важливою є якість системи, а не лише швидкість її розробки. CI працює у три простих етапи: впровадження, тестування та виправлення. Але, незважаючи на цю простоту, CI може стати складною справою, якщо лише деякі члени команди практикують це. Отже, ця методологія також вимагає змін у культурі та підтримки з боку керівництва.

Далі розглянемо три етапи робочого процесу CI.

Інтегруйте всі оволодіння щодня

З усіх трьох етапів безперервної інтеграції, впровадженням змін у основну гілку проекту технічно оволодіти найпростіше, але в культурному відношенні

найскладніше. І це правда тому, що щоденна інтеграція коду не означає, що розробник впроваджуватиме код у гілку змін конкретного функціоналу. Практика CI полягає у введенні коду в головну гілку, оскільки саме ця гілка буде використовуватися для випуску програмного забезпечення. Етап "push to master" також відомий як розробка на основі магістралі (trunk-based development).

Коли ви практикуєте CI, це не означає, що ви більше не будете використовувати додаткові гілки, що були призначені для інтеграції нового функціоналу. Єдина відмінність полягає в тому, що, оскільки ви посилюєте зворотний зв'язок при постійній інтеграції коду, гілки стають тимчасовими. Наприклад, гілка може жити лише день, а потім вона інтегрується в головну гілку.

Якщо нові функції чи виправлення помилок не були виконані до кінця, ви можете інтегрувати неповні зміни, використовуючи прапори функцій. Прапори функцій - це умова «if», яка визначає, запускати новий код чи ні. Якщо зміна ще не завершена, прапорець вимкнено за замовчуванням. Таким чином, коли ви інтегруєте код, решта команди має шанс переглянути його. Той самий спосіб застосовується, якщо новий код має помилки.

#### Покладайтесь на автоматизовані тести

Для перевірки кожного разу, коли розробник інтегрує новий код, CI покладається на автоматизований та надійний набір тестів. Якщо вам потрібно скомпілювати код, першим тестом є те, чи код компілюється чи ні. Тоді ви можете включити стільки тестів, скільки вважаєте за потрібне.

Виникає питання скільки тестів слід включити. Щоб це визначити, пам'ятайте, що метою CI є надання якнайшвидшого зворотного зв'язку. Якщо розробнику доведеться почекати годину, щоб отримати відгук - це не спрацює. Розробник вважатиме за краще чекати, а не інтегрувати код, таким чином впровадження може трапитися через день. Крім того, якщо побудова триває занадто довго, команда вимагатиме узгодження того, хто може інтегрувати. Ось чому команда, можливо, захоче повернутися до звичайного процесу розробки і використовувати довгі гілки впровадження нових функцій. Зберігайте петлю CI



якомога коротшою. Можливо, вам доведеться звернути більше уваги на модульне тестування (Unit Testing) та злегка на інтеграційне тестування (Integration Testing).

Отже, набір тестів повинен мати можливість надати інформацію, що щось порушено; тести повинні бути надійними. Коли ви виявляєте помилку у розробці, створіть тест і включіть його до циклу CI. Ви можете легко впевнитися, що це надійно: якщо прапорець для виправлення помилки вимкнено і збірка зламалася, тоді як якщо прапорець для виправлення помилки увімкнено, збірка працює.

Звертайте увагу на виправлення зламаної збірки в першу чергу

Коли збірка зламана, її виправлення має бути пріоритетом для команди. І важливість його виправлення має бути спільною звичкою у культурі.

Фіксація конструкції як загально прийнятий принцип, походить від системи компанії Toyota, яку виробник автомобілів використовував для виробництва автомобілів. Джон Вілліс написав пост на цю тему, і в ньому він описує процес Toyota так:

Toyota впровадила шнур Andon як фізичну мотузку, яка слідувала за конвеєром і яку можна було потягнути, щоб зупинити виробничу лінію в будь-який час. Більше того, це не був запит дозволу зупинити лінію. Тяга фактично зупиняла лінію.

У програмному забезпеченні еквівалент шнура Toyota полягає в тому, що кожен раз, коли порушується збірка, ніхто інший не може продовжувати інтегрувати зміни в коді.

Що є досить цікавим та неочікуваним, у Toyota шнур Andon можуть тягнути багато разів на день. Усі в команді дізнаються про виникнення проблеми, та що відбуваються. Крім того, той, хто перериває збірку, спробує вирішити проблему, і якщо потрібна допомога, тоді решта команди буде там. У програмному забезпеченні, якщо збірку не вдається виправити протягом декількох хвилин, команда повинна вирішити, видаляти написаний код чи вимкнути прапорець функції. Наявність "зеленої" будови вже не проблема однієї людини. Це повинно бути пріоритетом, як у Toyota.



Ідея виправлення зламаної збірки полягає в тому, що збірка завжди буде виробляти робочий код, який можна випустити.

### Інструменти для CI

CI - це переважно культурний зсув, але деякі інструменти можуть допомогти вам швидко впровадити ці зміни. За словами Джеймса Шора, ви навіть можете почати з долара на день. Все що потрібно для початку застосування цієї методології – це стіл та дзвоник на ньому. Джеймс Шор таким чином чітко висловлює важливу думку: відсутність інструментів не є виправданням. Ви можете впровадити CI без них.

Тим не менше, інструменти можуть допомогти. Ось список найпоширеніших інструментів, які ви можете почати використовувати сьогодні.

Jenkins - безкоштовний інструмент з відкритим кодом на основі Java, який надає вашому проекту велику гнучкість.

Azure Pipelines - продукт Microsoft, безкоштовний для до п'яти користувачів та проектів з відкритим кодом.

Cloud Build - керована послуга від Google Cloud Platform.

Travis CI - популярний інструмент для проектів із відкритим кодом GitHub, який пропонує рішення, яке розміщується на хостингу людиною або розміщується самостійно.

GitLab CI - безкоштовний інструмент від GitLab, який також може інтегруватися з іншими інструментами через API.

CircleCI - інструмент, який популярний для проектів GitHub і має рішення для розміщення на хостинг розробником та автоматично. Почати можна безкоштовно.

CodeShip - рішення, яке розміщується на хостинг автоматично. Ви можете розпочати з безкоштовної версії, але це платний інструмент.

Інструментів CI є більше, але у даному списку наведено найбільш вживані на різноманітних сучасних проектах. Важко визначити найкращий інструмент, але на даний момент можна сказати, що це Azure Pipelines. З його допомогою ви можете

інтегрувати багато інструментів. Крім того, він простий у використанні, і ви можете визначити код у вигляді послідовностей.

### Безперервна доставка CD

Безперервна доставка - це можливість отримувати зміни всіх типів - включаючи нові функції, зміни конфігурацій, виправлення помилок та експерименти - у виробництво або в руки користувачів, безпечно та швидко на стійкій основі. Ми досягаємо всього цього, гарантуючи, що наш код завжди знаходиться у стані розгортання, навіть перед особами тисяч розробників, які щодня вносять зміни.

Ви постійно проводите зміни всіх типів у середовищі для споживача. Ви можете доставити зміни конфігурації, зміни інфраструктури, чи будь-які інші. Зазвичай CI, як відомо, є практикою розробника, а CD практикою оператора. Місія CI полягає у наданні артефакту в певний момент часу програми, який задовольняє очікування клієнтів - іншими словами, має хорошу якість.

Потім місія CI полягає в переміщенні цих артефактів у всі різні середовища життєвого циклу організації. Найважливіше для CD - це те, що він завжди буде використовувати один і той же артефакт у всіх середовищах. Отже, побудова в CI відбувається лише один раз, а не для кожного середовища. Вироблений артефакт працюватиме із плейсхолдерами або змінними середовища з підходом разової побудови продукту.

Іншим критичним фактором є те, що для більш плавного розгортання кожне середовище, що відрізняється від виробничого, повинно бути подібним. Розробка, тестування та впровадження повинні бути виробничим середовищем. Однорідні середовища можливо буде важко досягти у великих організаціях, але ідея полягає у використанні однакових інструментів, процесів та конфігурацій у всіх середовищах. У CD виробництво не є особливим середовищем; це лише черговий етап у процесі розробки.

У якийсь момент розгортання в будь-якому середовищі стане нудним нудним. І це основна мета цієї методології.

## Принципи CD

Нижче наведено життєво важливі принципи постійної доставки. Коли ви застосуєте їх на практиці, вони допоможуть вам безпечно та швидко доставити програмне забезпечення стійким способом, відповідно до визначення CD, яке описано вище.

### Систематизовані невеликі розгортання

Зазвичай кожного разу, коли відбувається розгортання, стабільність програми знаходиться під загрозою. Тому ми прагнемо віддаляти розгортання один від одного. Але проблема такого підходу полягає в тому, що в підсумку ми накопичуємо багато змін. Швидше за все, одна з цих змін може мати проблеми, що змушує нас відкотити інші зміни, які працювали.

До чого CD сприяє у поєднанні з CI, що передбачає якість продукту - це те, що ми повинні працювати невеликими кроками.

Коли вам потрібно внести суттєві зміни в програму, використовуйте прапори функцій, як це робив Facebook, перш ніж випустити Messenger. Або застосуйте принцип декомпозиції і розділіть складні зміни на невеликі та прості зміни. Якщо ви робите розгортання частіше і працюєте невеликими партіями, ризик розгортання буде меншим.

### Автоматизація з людським втручанням

У компанії Toyota, це називають Jidoka, або «автоматизація з людським дотиком» - це філософія, що використовується для автоматизації. Вся повторна ручна праця, яку люди чи інженери виконують постійно під час кожного розгортання, повинна бути автоматизованою. Люди, які дотримуються завченими рецептами на пам'ять або читають інструкцію з експлуатації, зазнають невдачі, коли вони знаходяться під тиском відповідальності. І навпаки, машини - ідеальні кандидати для повторюваних завдань. Автоматизація є найважливішим принципом у CD, оскільки вона сприяє підвищенню стійкості процесу.

Після того, як ви запровадите автоматизацію, вам все одно потрібні будуть думки та рішення людей для вдосконалення процесу. Ви можете досягти автоматизації за допомогою таких практик, як інфраструктура як код, конвеєр як код або управління конфігурацією (infrastructure-as-code, pipeline-as-code, configuration management).

CD допоможе створити розгортання в один клік, які можуть бути активовані на вимогу в будь-який час.

Завжди вдосконалюйте систему

Якщо ви не порівнюєте власні результати з чужими, ви не можете вдосконалюватись. Якщо ви знаєте, що можливостей для вдосконалення багато, почніть із простих речей. У якийсь момент, як вже було описано раніше, розгортання стануть нудними з низьким ризиком.

Якщо Ви зараз збираєте програму на машині розробника, почніть із використання виділеного сервера для CI та побудуйте додаток на іншому сервері. Якщо Ви вручну копіюєте та вставляєте артефакти програми, почніть з автоматизації процесу копіювання / вставки. Не має значення, додали ви невеликі вдосконалення чи задоволені тим, наскільки вдосконалений процес. Завжди є що вдосконалити. CD не має дати завершення.

Модель спільної відповідальності

CD робить працю команди легшою, оскільки зараз конвеєр розгортання - це не просто операційна проблема. Системні інженери будуть шукати шляхи, щоб допомогти розробникам якісно створювати програмне забезпечення. І вони зроблять це не просто тренінгом, а й надаючи всі необхідні інструменти, необхідні розробнику, щоб краще зрозуміти проблеми. Команда не буде оптимізувати процес лише для них, але вона додасть вдосконалення, які допоможуть організації надати більшу цінність для клієнтів. Це замість традиційного підходу винагороджувати розробників за те, як швидко вони доставляють, або винагороджувати системних

інженерів за те, наскільки надійна система. Якщо це те, що ви робите, ці дві команди матимуть різні цілі, і це не буде працювати.

Кожен у команді відповідає за постійне створення більш безпечного, швидкого та детермінованого конвеєра доставки.

### Інструменти для CD

Деякі інструменти для CD - це також інструменти для CI. Ось чому тут може зустрітися кілька інструментів зі списку інструментів для CI. Але є також кілька нових.

Jenkins - також може бути використаний для CD із плагінами Ansible або Terraform.

Azure Pipelines - містить розділ визначення впровадження, який ви можете інтегрувати зі стадією побудови з CI.

Spinnaker - набирає популярність, і це інструмент, за допомогою якого Netflix робить релізи на CD.

GitLab CI - дозволяє налаштувати процеси розгортання та релізу за допомогою GitLab.

GoCD - пропозиція ThoughtWorks, яка застосовує принципи, описані вище.

Знову ж таки, цей список не в змозі визначити, скільки насправді є інструментів. Ви можете використовувати будь-який інструмент, який найкраще вам підходить.

### Типовий робочий процес CI / CD

Нарешті, можна зібрати всі концепції, які описані вище, із традиційним, але простим робочим процесом для процесу CI / CD із використанням стека Microsoft. Кожна частина може бути взаємозамінна з іншими інструментами, платформами та мовами.

Важливим є робочий процес, який зображено на рисунку 6. Рисунок показує всі кроки описаної методології.



Рисунок 2.6 – Методологія CI / CD

Підсумовуючи, ось як можна це описати:

- 1) Інженер кодує зміни програми за допомогою Visual Studio.
- 2) Коли код готовий до інтеграції, його надсилають до сховища Git в Azure Repos.
- 3) CI автоматично запускає виконання тестових випадків, які підтверджують, що код доступний для впровадження.
- 4) У Azure Pipelines конвеєр випуску запускається автоматично для розгортання артефактів, вироблених на етапі CI.
- 5) Артефакт потрапляє у веб-програму Azure - скажімо, до середовища розробки.
- 6) Application Insights збирає інформацію з сайту для надання зворотного зв'язку з командою.
- 7) Команда використовує інформацію, доступну після випуску, щоб дізнатись про стан та вплив останньої версії.
- 8) Будь-яка нова функція чи виправлення помилок додається та визначається пріоритетом у відставанні.

- 9) Коли відбувається впровадження в інше середовище, таке як тестування або виробництво, кроки 1–4 не потрібно буде запускати знову.

CI / CD допомагає командам бути більш продуктивними при доставці програмного забезпечення з якісною вбудованою системою. Але шлях до розгортання в один клік, який можна створити на вимогу, не є простим. Це головним чином тому, що хоч і існують потужні інструменти, які допоможуть вам досягти CI / CD ефективніше, ця методологія вимагає культурних змін: мислення, яке кожна людина в команді повинна розвивати в собі.

Зрештою, впровадження CI / CD не має дати завершення, як проект, де коли ви вже знаходитесь на певному етапі про це можна забути. Він повинен постійно вдосконалюватися. Платформи, такі як Retrace, що забезпечують централізоване ведення журналу та відстеження помилок, допоможуть вам постійно вдосконалюватися на основі відгуків, які ви отримуєте після кожного випуску у виробництво.

### **2.3. Системи контролю версій**

Що таке "контроль версій", і чому він є важливим. Контроль версій - це система, яка фіксує зміни у файлі чи наборі файлів з часом, щоб ви могли відновлювати певні версії пізніше. На практиці контроль версій можна використовувати практично з будь-якими файлами на комп'ютері.

Якщо ви графічний або веб-дизайнер і хочете зберегти кожен зображення чи макета, система управління версіями (VCS) - дуже розумна річ для використання. Вона дозволяє повернути вибрані файли назад у попередній стан, повернути весь проект назад у попередній стан, порівняти зміни з часом, побачити, хто останній змінив щось, що могло спричинити проблему, хто та коли вніс проблему та багато іншого. Використання VCS також, як правило, означає, що

якщо ви зіпсуєте щось або втратите файли, ви зможете легко відновити їх. Крім того, ви отримуєте все це за дуже невеликі накладні витрати.

Контроль версій, також відомий як Редакційний контроль, або контроль джерела - це процес управління колекцією програмного коду та змін, який надає вам безліч можливостей, таких як:

- Підтримка кількох версій коду
- Можливість повернутися до будь-якої попередньої версії
- Розробники можуть працювати паралельно
- Аудит простежуваності з чітким уявленням про те, хто, коли, де і які зміни зробив
- Синхронізація коду
- Копіювання / об'єднання / скасування змін
- Простежування різниці між версіями
- Забезпечує повне резервне копіювання, не займаючи багато місця
- Перегляд історії змін
- Доступний як для малих, так і для великих проектів
- Можливість ділитися та працювати над кодом по всьому світу

Простішими словами, система контролю версій дозволяє керувати та відстежувати весь свій вихідний код, а також надає докази того, що всі ваші зміни зберігаються у сховищі.

Існує декілька видів систем контролю версій [13]

Локальні системи контролю версій

Багато людей у якості контролю версій використовують примітивне копіювання файлів, чий стан вони прагнуть зберегти, у іншу папку на тому ж комп'ютері. Цей підхід дуже поширений, бо він дуже простий у застосування, але попри це він спричиняє багато проблем та помилок. Наприклад дуже просто забути, в якій папці ви знаходитесь, і ненароком перезаписати в невірний файл або скопіювати файли, які ви не хотіли копіювати.



Для вирішення цієї проблеми колись давно було розроблено локальні VCS, які мали просту базу даних, що зберігала всі зміни у файлах і давала можливість редагувати їх.

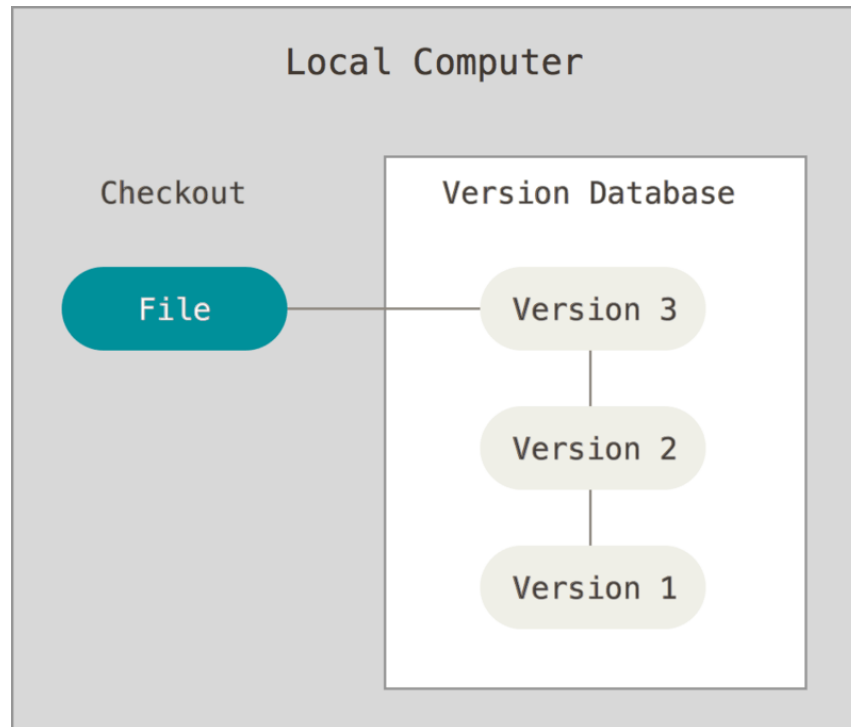


Рисунок 2.7 - Діаграма Локального контролю версій

Одною з найпоширеніших інструментів VCS була система RCS, яка і зараз використовується багатьма машинами. RCS зберігає записи виправлень (тобто різницю між файлами) у спеціальному форматі в пам'яті комп'ютера; потім можливо заново відтворити, як виглядав будь-який файл у будь-який момент часу, вказавши всі налаштування.

### Централізовані системи контролю версій

Інша велика проблема, з якою стикаються люди, полягає в тому, що системи розробляються багатьма розробниками одночасно з різних машин. Для вирішення цього питання були розроблені централізовані системи контролю версій (CVCS). Ці системи (такі як CVS, Subversion і Perforce) мають єдиний сервер, який містить

усі версії файлів, а також системи, які перевіряють актуальність цих файлів. Протягом довгого періоду часу це було стандартом контролю версій.

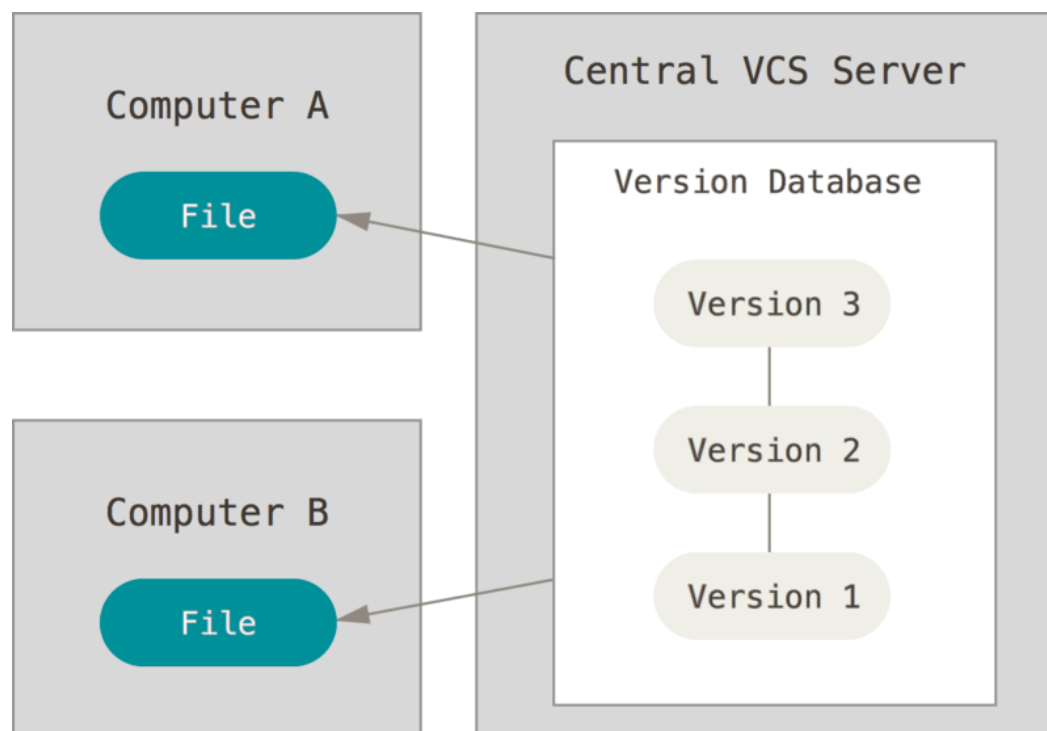


Рисунок 2.8 - Діаграма централізованого управління версіями

Такий метод має ряд переваг, особливо, якщо порівняннювати з локальними VCS. Наприклад, всі учасники проекту більше чи менше знають, чим займаються їх колеги кожного робочого дня. Адміністратори мають чіткий контроль над тим, хто що може робити, і набагато простіше керувати CVCS, ніж мати справу з локальними базами даних кожного клієнта.

Однак у цієї системи також є серйозні мінуси. Найбільш очевидним є момент коли відмовляє центральний сервер. Якщо в не працює годину, то протягом цієї години ніхто не може взагалі діставати з нього збережені файли або зберігати нові версії файлів, над якими вони працюють. Якщо жорсткий диск, на якому знаходиться центральна база даних, пошкоджується, а належні резервні копії не зберігаються, ви втрачаєте абсолютно все - всю історію проекту, за винятком останніх версій, які працівники мають на своїх локальних машинах. Локальні

системи VCS страждають від цієї ж проблеми - коли у вас є вся історія проекту в одному місці, ви ризикуєте втратити все.

### Розподілені системи контролю версій

З метою виправити таку проблему з'явилися розподілені системи контролю версій (DVCS). У DVCS (таких як Git, Mercurial, Bazaar або Darcs) клієнти не просто перевіряють останні версії файлів, вони повністю розгортають сховище, включаючи всю його історію. Таким чином, якщо який-небудь сервер впаде, а всі системи співпрацювали через цей сервер, будь-яке з клієнтських сховищ може бути скопійовано назад на сервер для його відновлення. Кожен клон - це по-суті повне резервне копіювання всіх даних.

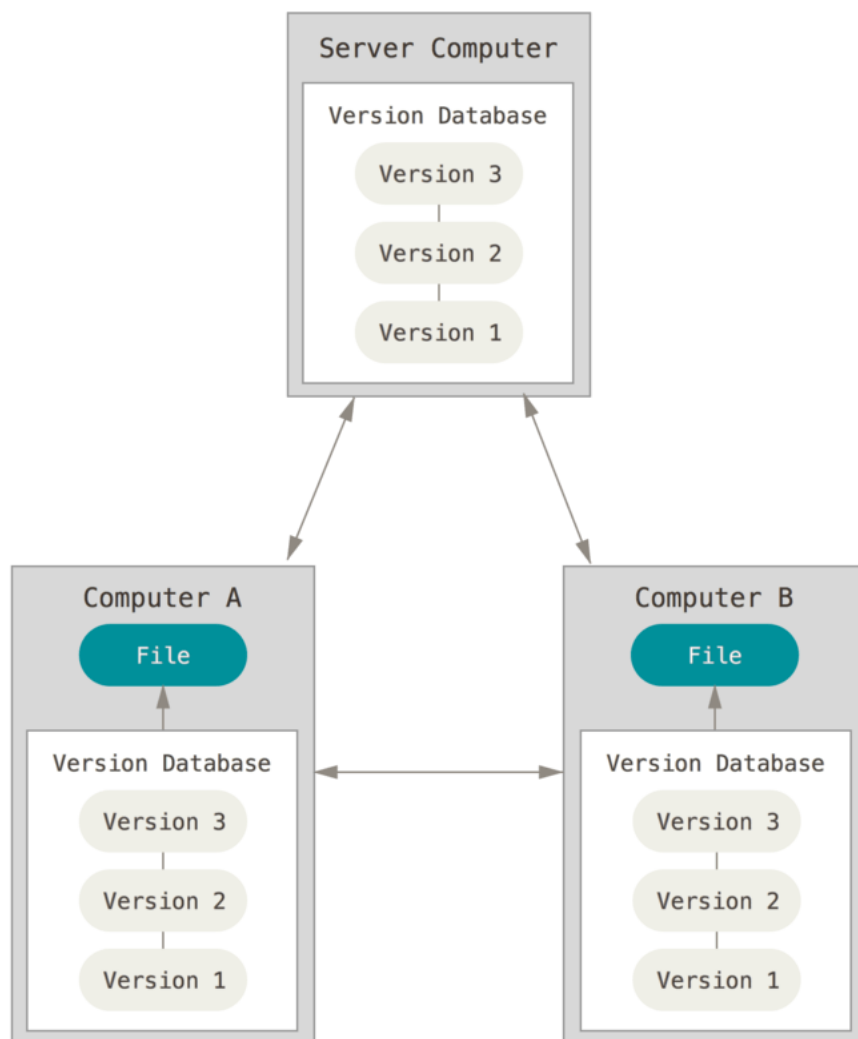


Рисунок 2.9 - Діаграма розподіленого контролю версій

Крім того, багато з цих систем досить добре справляються з наявністю декількох віддалених сховищ, з якими вони можуть працювати, тому ви можете співпрацювати з різними групами працівників по-різному одночасно в рамках одного проекту. Це дозволяє налаштувати кілька типів робочих процесів, які неможливі в централізованих системах, наприклад ієрархічні моделі.

Кращі практики систем контролю версій [14]:

Переконайтеся, що ви поклали весь свій код, документи, файли конфігурації тощо під систему контролю версій.

Часто оновлюйте свою робочу копію, щоб врахувати чужі зміни у вашій локальній робочій копії та отримувати останні.

Перш ніж виконувати коміт, всебічно протестуйте код, щоб переконатися, що він справді завершений і не має побічних ефектів.

Часто фіксуйте зміни, але не виконуйте половину виконаної роботи, яка може порушити будь-яку функціональність.

Пишіть описове повідомлення коміту, щоб будь-який інший учасник міг легко зрозуміти, що змусило вас здійснити цю зміну і чим воно відрізняється від попереднього коміту.

Завжди пам'ятайте, що інструменти базуються на рядках і переконайтеся, що не змінюєте формат файлу під час оновлення коду.

Використовуйте загальну структуру проекту та конвенцію про найменування, щоб уникнути плутанини для інших

Налаштовуйте сповіщень про зміни для розсилки електронного листа щоразу, коли робиться будь-яка комісія.

Завжди регулярно робіть резервні копії вашого сховища або плануйте щовечора створювати резервні копії всіх сховищ.

Переваги системи контролю версій

Система контролю версій працює як база даних всього вашого коду та вносить зміни, а не дублює файли, що допомагає економити багато місця на диску

Вона зберігає всю історію всіх файлів, що забезпечує повну простежуваність і чутність змін щодо того, які зміни були внесені в який файл, коли, чому та ким.

Вона надає можливість повернутися до останньої версії або будь-якого попереднього етапу відповідно до вимог.

Вона запобігає ризику втратити функціонуючий код або зламати тестові сценарії, перезаписуючи файли, оскільки ви завжди можете вийняти останній робочий код у будь-який момент часу.

Допомагає визначити різницю в будь-якому наборі файлів, порівняти версії та об'єднати зміни відповідно до вимог.

Вона надає вам можливість мати цілком незалежні версії коду, якщо ви віддаєте перевагу іншому коду розробки, після того, як будете готові, ви можете об'єднати файли, щоб створити остаточну робочу версію.

Вона надає можливість працювати як розподілена команда з повною співпрацею по всьому світу та економить час та додаткові зусилля для всіх. Не потрібно чекати, поки інші закінчать роботу.

## 3. РОЗРОБКА СИСТЕМИ

### 3.1. Сайт кафедри програмної інженерії

Згідно вимог, веб-сайт кафедри розроблено на системі менеджменту контенту (CMS) WordPress. Оскільки основною вимогою до сайту була можливість адміністрування, модерації та конфігурування, цей сервіс ідеально підходить для його розробки. Він надає можливість створювати веб-сайти використовуючи візуальний редактор та не втручаючись у вихідний код. Для швидкого формування структури сайту WordPress пропонує безліч готових шаблонів, які можна редагувати під свої потреби. Для того щоб додати до власного сайту особливий функціонал можна використовувати плагіни, створені іншими користувачами.

Отже, для основи структури сайту кафедри програмної інженерії було обрано шаблон системи WordPress під назвою Twenty Seventeen.

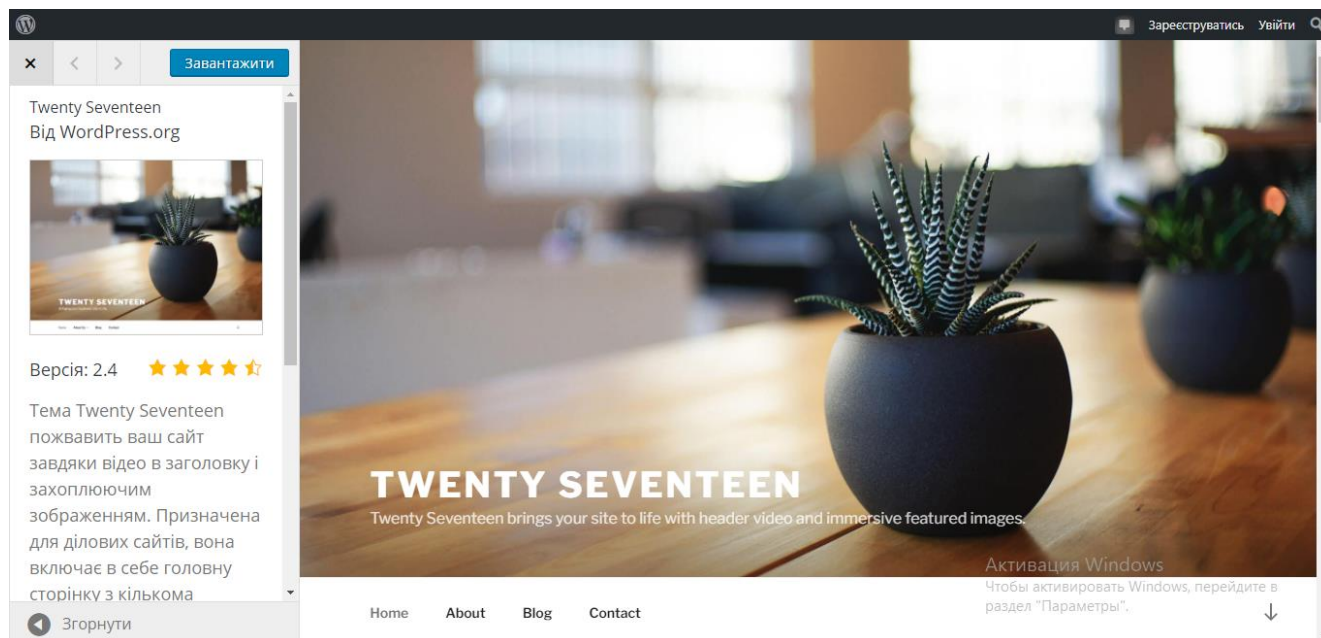


Рисунок 3.1 – Шаблон Twenty Seventeen

Структуру навігації (Меню) сайту розроблено таким чином, щоб у неї входили всі пункти, вказані у вимогах до сайту. Для зручності навігації, кнопки меню розроблено так, що при натисканні на них при потребі відкривається ще одне

Drop Down меню. Це допомагає зробити навігацію зручнішою, більш розширеною, а саме меню компактнішим та візуально структурованішим.

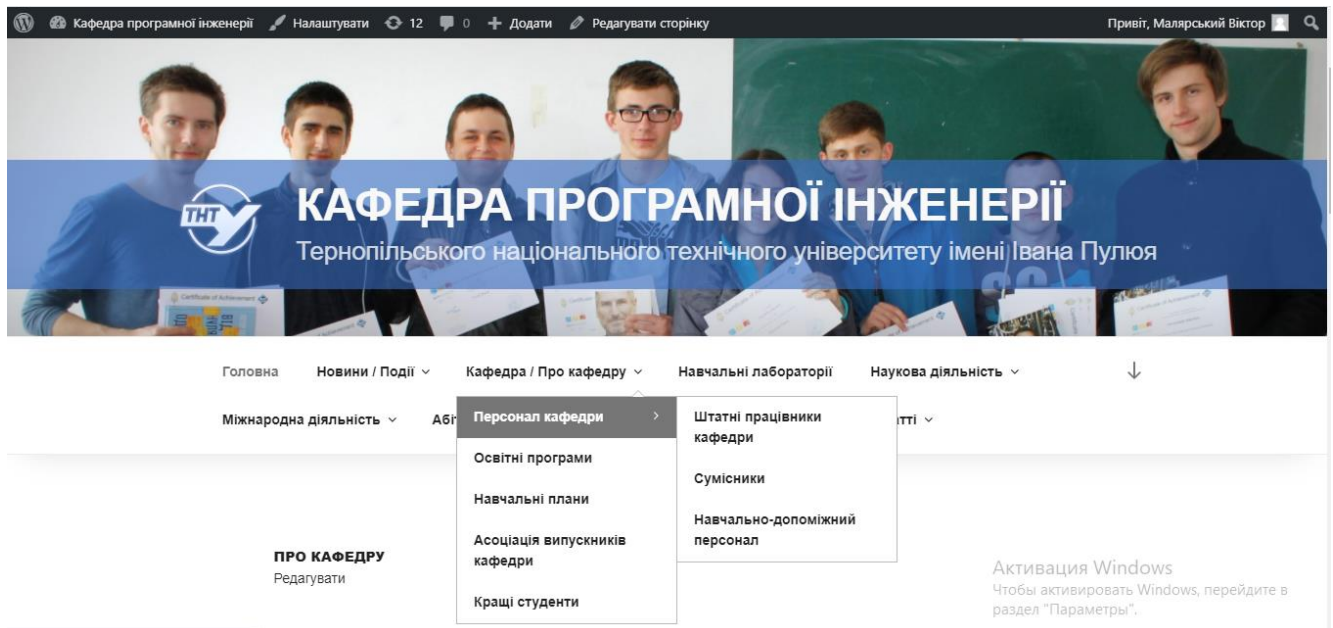


Рисунок 3.2 – Навігація сайту

Пункти та підпункти меню сайту кафедри:

Головна сторінка (Про кафедру)

Новини/Події

- Новини кафедри
- Новини університету

Кафедра/Про кафедру

- Персонал кафедри (Штатні працівники кафедри, Сумісники, Навчально-допоміжний персонал)
- Освітні програми
- Навчальні плани
- Асоціація випускників кафедри
- Кращі студенти

Навчальні лабораторії

Наукова діяльність

- Наукова робота
- Наукові проекти
- Наукове співробітництво
- Наукові здобутки
- Аспірантура

#### Міжнародна діяльність

- Міжнародне співробітництво
- Міжнародні проекти, гранти, нагороди
- Подвійні дипломи
- Стажування (дисертації) за кордоном

#### Абітурієнту

- Бланки
- Цікаві статті

#### Студенту

- Графіки навчального процесу
- Навчальні дисципліни кафедри
- Вибіркові/альтернативні дисципліни
- Навчально-методична література кафедри
- Практики
- Курсове та дипломне проектування

#### Контакти

#### Логін/Реєстрація

- Профіль

#### Статті

- Безпека програм та даних
- Розробка прикладного програмного забезпечення для Бази Даних
- Artificial Intelligence and Data Mining

Для встановлення додаткового функціоналу сайту кафедри, у систему було завантажено ряд плагінів:



## Content Views

Плагін, що дозволяє створювати красивий вигляд веб-сайту, допомагаючи структурувати контент або статті у вигляді красивих сіток чи списків. Має зручний інтерфейс, зрозумілий функціонал та не вимагає поглиблених знань програмування для створення та адміністрування статей.

## CyberSyn

CyberSEO Lite - це спрощена версія CyberSEO Pro - найдосконалішого, але простого у використанні плагіна для автоматичного ведення блогу та курирування вмісту для WordPress. CyberSEO Lite - це потужний агрегатор RSS, призначений для імпорту RSS-каналів та автоматичного створення статей WordPress для вашого веб-сайту. Плагін оснащений унікальними та дуже потужними програмними засобами, які ви можете знайти лише у преміальних рішеннях.

На даний момент CyberSEO Lite - це єдиний безкоштовний плагін для агрегації RSS із відкритим кодом, який здатний витягувати повнотекстові статті зі скорочених RSS-каналів, перекладати статті з / на понад 100 мов за допомогою служб Google Translate і Yandex Translate, обертати його зі сторонніми вертушками вмісту \* та іншими. Всі ці функції на 100% безкоштовні та доступні відразу після активації плагіна.

У нашому випадку цей плагін було використано для того щоб синхронно відображати новини університету, що автоматично завантажуються з сайту [tntu.edu.ua](http://tntu.edu.ua) .

## Duplicator

Простий плагін для створення повних копій та бекапу сайтів на WordPress, включаючи копіювання баз даних. За допомогою цього плагіну можна робити резервне копіювання сайту для того, щоб безпечно робити зміни на сторінках та не перейматися про втрату якихось файлів чи даних. Також цей плагін використовують для перенесення сайту на WordPress з одного хостингу чи серверу на інший. Він дозволяє автоматизувати цей процес.

Мінусом цього плагіну є те, що резервне копіювання потрібно кожного разу проводити самостійно.

### Embed Any Document

Цей WordPress плагін дозволяє легко завантажувати та вставляти документи на веб-сайт WordPress без будь-яких додаткових плагінів браузера, таких як Flash або Acrobat reader. Плагін дозволяє вибрати між засобами перегляду документів Google та Microsoft Office Online для відображення ваших документів.

### Feedzy RSS Feeds Lite

Агрегатор RSS FEEDZY допомагає вам об'єднати необмежену кількість RSS-каналів та автоматично публікувати їх на вашому сайті протягом декількох хвилин.

За допомогою FEEDZY RSS Aggregator ви можете сидіти склавши руки, дозволяти своєму веб-сайту автоблогувати і завжди залишатись свіжим з новим вмістом, що натомість допомагає покращити маркетинг вмісту, підвищити рівень SEO, збільшити залучення аудиторії та, можливо, заробити більше грошей.

Цей плагін також було використано для автоматичного підтягування новин університету з офіційного сайту ТНТУ.

### Nav Menu Roles

Цей плагін дозволяє налаштовувати вигляд головного меню сайту в залежності від рівня доступу користувача (zareєстрований/не zareєстрований користувач, адміністратор, супер-адміністратор)

### Restrict User Access

Дозволяє створювати різні додаткові рівні доступу користувачів та налаштовувати видимість інформації та доступ до різного функціоналу для них. Це дає можливість посилити безпеку сайту.

## Tabs Responsive

Tabs Responsive заснований на фреймворку Bootstrap, тому цей дизайн плагіна чуйний та зручний для використання на будь-яких мобільних пристроях. За допомогою цього плагіна ви можете відображати стиль вертикальних та горизонтальних вкладок. Додайте кілька вкладок на декілька сторінок і розміщуйте публікації. Для вмісту доступна необмежена колірна схема та стиль шрифту.

Ви можете додати зображення, відео, музику або інший шорт-код у поле опису вкладки, сторінки за допомогою плагіна colorbox. Там також доступна опція віджетів, просто додайте адаптивний віджет "Вкладки" у свою область віджетів.

## User Registration

Плагін реєстрації користувачів надає вам простий спосіб створити зовнішню реєстраційну форму користувача та форму входу. Поля перетягування роблять впорядкування та створення форм надзвичайно простими. Плагін легкий, розширюваний і може використовуватися для створення будь-якого типу реєстраційної форми.

Окрім реєстраційної форми, плагін також підтримує налаштування сторінки облікового запису зовнішнього профілю для редагування профілю, зміни пароля, виходу та іншого. Користувачі можуть відвідати сторінку свого облікового запису після реєстрації та переглянути деталі, які вони заповнили, і можуть внести зміни, якщо це необхідно.

Реєстраційні форми на 100% адаптуються до мобільних пристроїв та оптимізовані для показу на будь-яких пристроях.

Під час отримання технічного завдання, було проведено опитування адміністраторів веб-сайту кафедри для того щоб отримати інформацію, що їх влаштовує та що не влаштовує у існуючій системі, які функції потрібно покращити, які забрати чи додати. Після опитування було проведено детальний аналіз зауважень та разом з адміністраторами було вирішено:

- Для того щоб забезпечити стабільну роботу сайту в умовах обмежених інформаційних ресурсів, організувати та підключити сайт до системи контролю версій. Це дозволить повністю контролювати зміни на сторінках, робити відкати сайту до попередніх версій, відмінити неінтегровані в репозиторій зміни, а також робити автоматичні резервні копії сайту та бази даних.
- Додати функціонал, що полегшить створення та редагування сторінок та статей на сайті. Вся конфігурація буде відбуватися у візуальному редакторі.
- Зробити кожен сторінку сайту адаптивною до мобільних пристроїв. Забезпечити коректне відображення сторінок на пристроях з будь-яким розширенням.

### 3.2. Копіювання сайту на локальний сервер

Для того щоб безпечно робити всі зміни на сайті WordPress, та і взагалі будь-якому сайті, його варто повністю скопіювати на локальний сервер та створити тестове середовище для того щоб робити якісь глобальні зміни у системі і при цьому не переривати її роботу та не змінювати її на реальному робочому середовищі, або ж у нашому випадку домені.

На даний час робочий домен сайту кафедри програмної інженерії знаходиться на внутрішньому FTP сервері університету. Для того щоб зробити його копію потрібно зайти на нього використавши дані для входу, а саме доменне ім'я, ім'я користувача, пароль та порт приєднання серверу.

Для того щоб виконати вхід та копіювання файлів з серверу було використано систему керування файлами FileZilla FTP Client. Ця система дозволяє виконувати будь-які дії з файлами як на локальному комп'ютері, так і на серверах з FTP доступом.

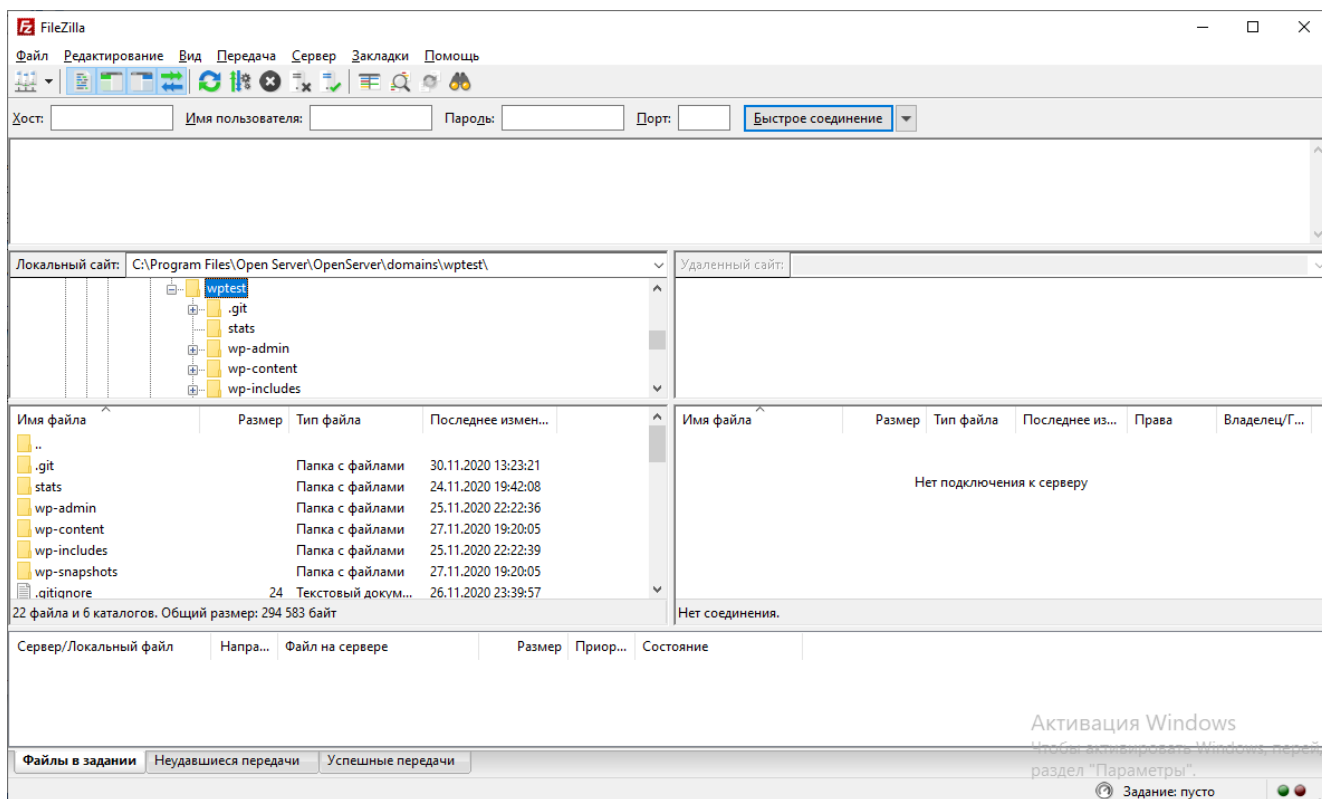


Рисунок 3.3 – Интерфейс системы FileZilla FTP Client

Після того як було зроблено копію сайту з серверу, залишилось зробити резервну копію бази даних. База даних також знаходиться на хостингу веб-сайту. Потрібно зайти у порт MySQL, запуститься система керування базами даних phpMyAdmin, туди також потрібно ввести логін та пароль, і тоді ми зможемо знайти серед інших таблиць потрібну нам базу даних сайту кафедри та експортувати її.

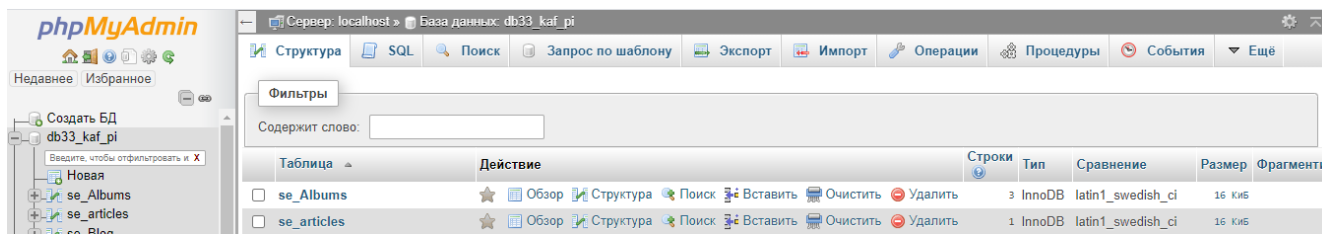


Рисунок 3.4 – Експортування бази даних

В результаті ми отримуємо два архіви, один з файлами WordPress, інший з базою даних сайту.

Далі потрібно встановити скопійований сайт на локальний сервер. У якості локального серверу було використано OpenServer. Це популярна утиліта, яка легко встановлюється та налаштовується для використання як тестового середовища для розробки веб-сайтів. У склад даної програми входять такі модулі: HTTP, PHP, MySQL/MariaDB, PostgreSQL, MongoDB, Memcached, Redis, DNS.

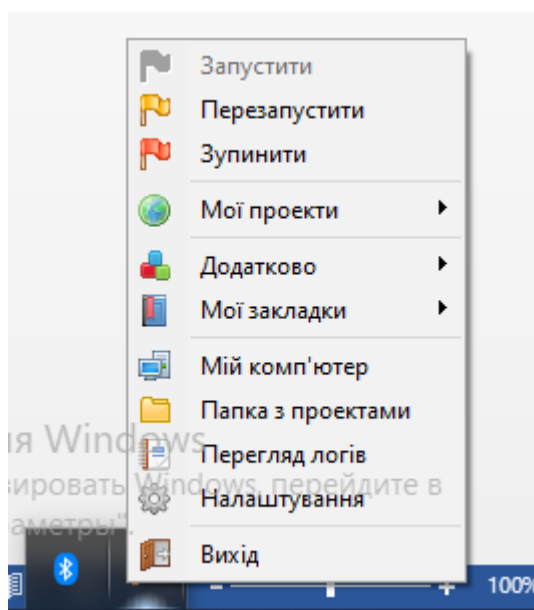


Рисунок 3.5 – Панель управління OpenServer

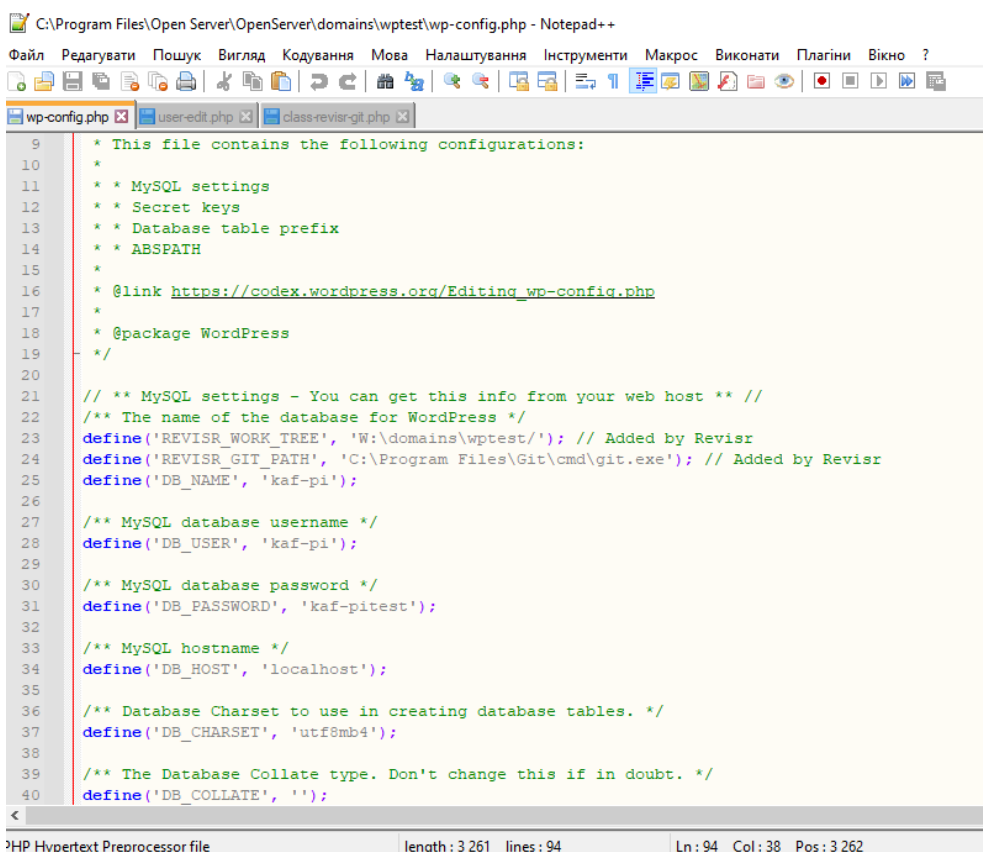
Створимо в кореневій папці серверу папку для веб-сайту та розархівуємо скопійовані файли WordPress у неї. Далі потрібно зайти у систему phpMyAdmin вже локального сервера, там створити нову базу даних, та імпортувати в неї архів скопійованої бази даних сайту кафедри.

Для того, щоб сайт на локальному сервері працював коректно потрібно замінити у базі даних всі збережені URL посилання робочого домену (<https://kaf-pi.tntu.edu.ua/>) на посилання локального серверу. Щоб зробити це швидко та автоматизовано, потрібно в консолі phpMyAdmin локального серверу ввести такі SQL команди:

```
UPDATE wp_options SET option_value = replace(option_value, 'http://kaf-
pi.tntu.edu.ua', 'http://wptest') WHERE option_name = 'home' OR option_name =
'siteurl';
UPDATE wp_posts SET post_content = replace(post_content, 'http://kaf-
pi.tntu.edu.ua', 'http://wptest');
UPDATE wp_postmeta SET meta_value = replace(meta_value, 'http://kaf-
pi.tntu.edu.ua', 'http://wptest');
```

Де “http://wptest” – це шлях до кореневої папки, що була створена на локальному сервері.

Далі потрібно змінити вміст файлу wp-config.php, а саме замінити збережені ім'я бази даних, логін та пароль до неї на ці ж показники тільки для створеної бази даних на локальному сервері



```
C:\Program Files\Open Server\OpenServer\domains\wptest\wp-config.php - Notepad++
Файл Редагувати Пошук Вигляд Кодування Мова Налаштування Інструменти Макрос Виконати Плагіни Вікно ?
wp-config.php user-edit.php class-revisr-git.php
9      * This file contains the following configurations:
10     *
11     * * MySQL settings
12     * * Secret keys
13     * * Database table prefix
14     * * ABSPATH
15     *
16     * @link https://codex.wordpress.org/Editing_wp-config.php
17     *
18     * @package WordPress
19     */
20
21     // ** MySQL settings - You can get this info from your web host ** //
22     /** The name of the database for WordPress */
23     define('REVISR_WORK_TREE', 'W:\domains\wptest/'); // Added by Revisr
24     define('REVISR_GIT_PATH', 'C:\Program Files\Git\cmd\git.exe'); // Added by Revisr
25     define('DB_NAME', 'kaf-pi');
26
27     /** MySQL database username */
28     define('DB_USER', 'kaf-pi');
29
30     /** MySQL database password */
31     define('DB_PASSWORD', 'kaf-pitest');
32
33     /** MySQL hostname */
34     define('DB_HOST', 'localhost');
35
36     /** Database Charset to use in creating database tables. */
37     define('DB_CHARSET', 'utf8mb4');
38
39     /** The Database Collate type. Don't change this if in doubt. */
40     define('DB_COLLATE', '');
```

Рисунок 3.6 – Оновлення файлу wp-config.php

Останнім кроком буде вимкнення для зручності автооновлення сторінок вордпрес. Для цього в кінці цього ж файлу wp-config.php потрібно прописати такий рядок:

```
define('WP_AUTO_UPDATE_CORE', false);
```

Тепер залишилось лише зайти на наш локальний сервер через веб-браузер. Для цього у URL рядку браузера прописуємо назву папки, яку ми створили на сервері.

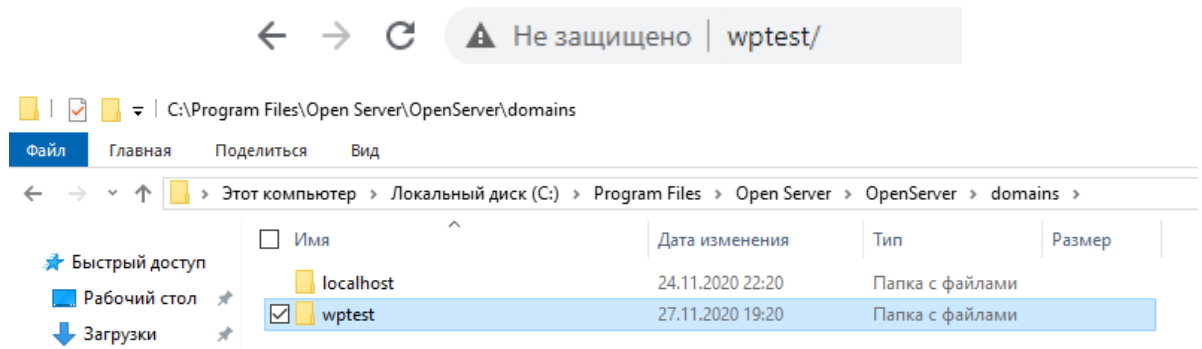


Рисунок 3.7 – Вхід на локальний сервер

Отже, ми успішно скопіювали сайт кафедри на свій локальний сервер та можемо сміливо проводити на ньому будь-які зміни.

### 3.3. Впровадження системи контролю версій

Зараз найпопулярнішою системою контролю версій є Git. Проте щоб користуватися нею потрібно детально вивчити всю її документацію та навчитися прописувати її команди у консолі GitBash. Це досить затратно в плані часу на навчання команди. До того ж щоб правильно налаштувати роботу хостингу з цією системою та безперервну передачу файлів між робочим середовищем, віддаленим репозиторієм на GitHub чи Bitbucket, та тестовим середовищем, потрібно докласти неабияких зусиль та часу. Також, оскільки WordPress має власну систему перезаписування файлів, її також потрібно адаптувати до Git вручну.

Існує просте рішення цих всіх проблем. WordPress має спеціальний вже готовий плагін для налаштування контролю версій та роботою з віддаленим репозиторієм на GitHub. Цей плагін має назву Revisr. Він дозволяє керувати веб-сайтом WordPress за допомогою сховища Git.

За допомогою Revisr ви можете:



- Відстежувати зміни до всієї інсталяції WordPress або лише тих частин, які вам більше подобаються
- Здійснювати коміти на інформаційній панелі WordPress
- Створювати резервні копії або відновлювати весь веб-сайт за лічені секунди
- Налаштовувати щоденні або щотижневі автоматичні резервні копії
- За бажанням імпортувати або експортувати зміни з віддаленого сховища, наприклад, Bitbucket або Github
- Перевіряти зміни перед розгортанням їх на іншому сервері
- Повертати файли веб-сайту та / або базу даних на попередню версію
- Швидко видаляти всі небажані зміни



Рисунок 3.8 – Плагін Revisr

Все що потрібно – це просто завантажити готовий плагін у нашу систему WordPress та створити репозиторій на GitHub. В панелі налаштувань Revisr вказуємо шлях до репозиторію, який можна скопіювати у GitHub, і отримуємо готову до роботи систему контролю версій. Як можна побачити на рисунку 3.9, за допомогою панелі управління в правому кутку можна зберігати зміни, відмінити їх, створювати резервну копію бази даних, інтегрувати та синхронізувати робочі файли з файлами у репозиторії.

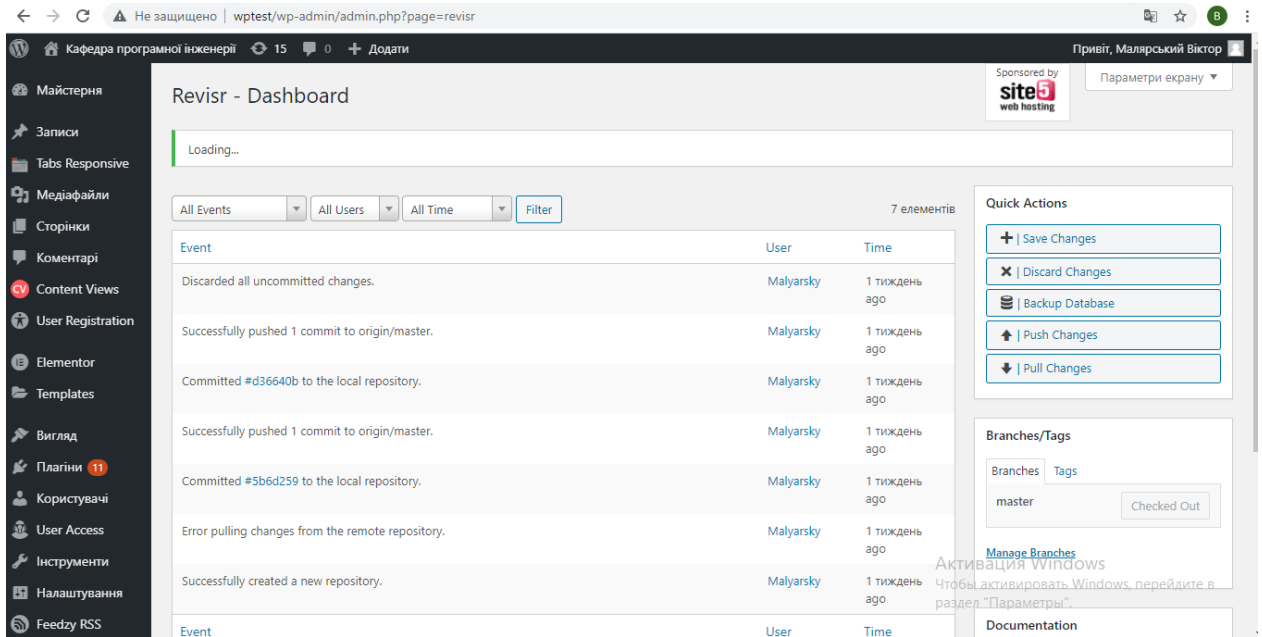


Рисунок 3.9 – Інтерфейс плагіну Revisr

Встановлення цього плагіну вирішує проблему потреби у системі контролю версій для сайту кафедри. Проте є невеликі недоліки. Revisr не відслідковує зміни у налаштуваннях шаблону сайту. Всі коміти потрібно імплементувати на репозиторій вручну. Зміни зберігаються лише тоді, коли адміністратор натискає на кнопку «Save Changes». Також цей плагін потребує підтримки сервером або хостингом системи Git, її потрібно налаштувати, інакше плагін просто не запуститься.

### 3.4. Встановлення нового функціоналу для створення сторінок та адаптація їх під мобільні пристрої

Як вже було неодноразово зазначено, WordPress – це найкраще рішення для тих, хто не має глибоких знань у програмуванні та веб-дизайні. Оскільки ця система має у собі вже готові рішення для більшості завдань та потреб користувачів.

Головною перевагою WordPress є наявність великої кількості плагінів для розробки сторінок різними методами, в тому числі і за допомогою візуальних редакторів. На мою думку, найдосконалішим та найзручнішим з них є Elementor.

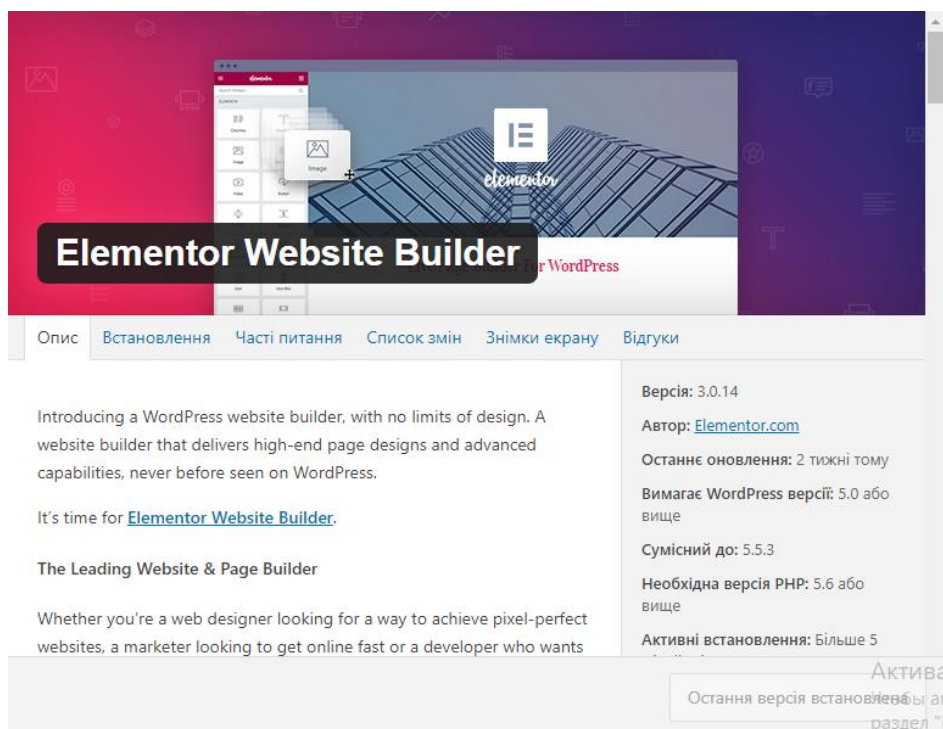


Рисунок 3.10 - Плагін Elementor

Elementor – це ціла система всередині WordPress. Він дозволяє як створювати сторінки з нуля, так і використовувати готові шаблони, та просто редагувати їх у власних потребах. В арсеналі цього плагіну є дуже багато видів віджетів та налаштувань, які можна комбінувати як завгодно.

Elementor надає можливість вистроювати сторінку за допомогою блоків, які є аналогами тегу <div> тільки за допомогою візуального редактора. Їх можна додавати, заповнювати та пересувати будь-яким чином. А зручна панель управління дозволяє конфігурувати всі елементи як заманеться користувачеві. Розміри, відступи, кольори, спливаючі вікна чи ефекти при наведенні мишкою на елемент – все це, і не тільки можливо налаштувати за допомогою цього плагіна.

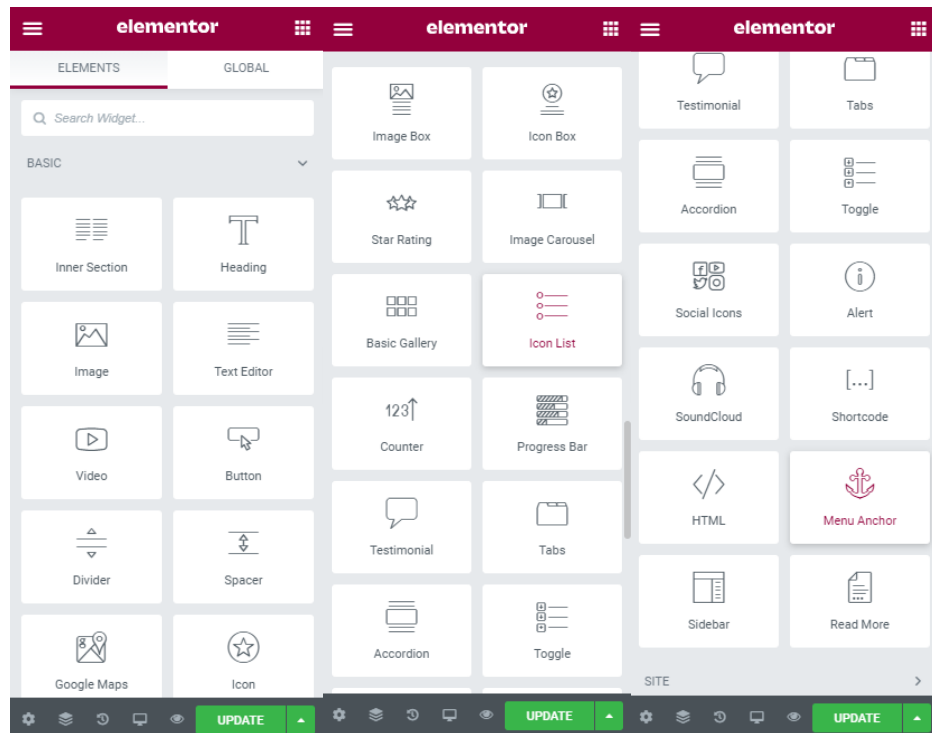


Рисунок 3.11 – Елементи панелі керування Elementor

Також великою перевагою даного плагіна є те, що у ньому є функція налаштування вигляду сайту окремо для кожного розширення девайсу: десктоп, планшет та смартфон. При чому налаштування розмірів чи певних функцій елементів для будь-якого розширення можуть бути різними. Це робить створені сторінки абсолютно адаптивними під всі девайси.

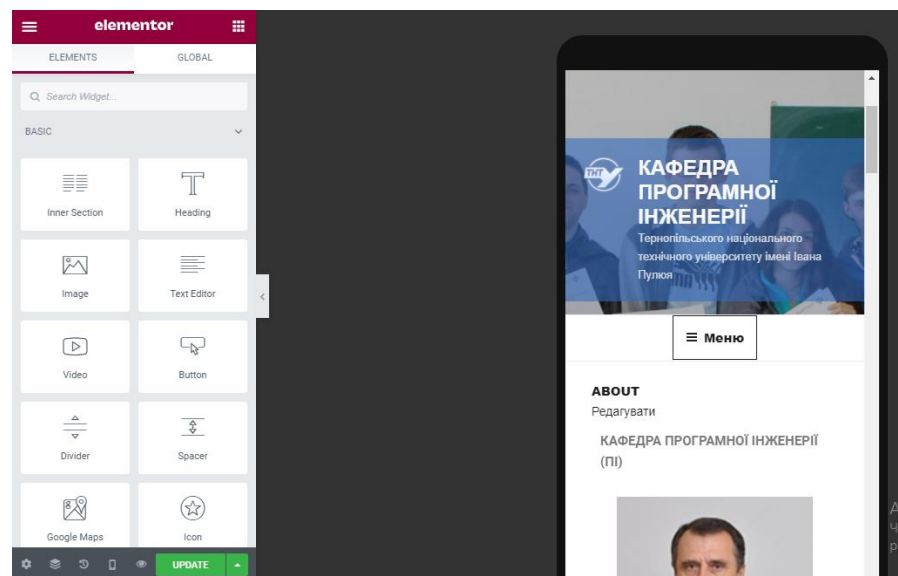


Рисунок 3.12 – Адаптація сторінки під мобільний пристрій

За допомогою плагіну Elementor було перероблено всі основні сторінки сайту кафедри програмної інженерії, проведено їх адаптацію під мобільні пристрої, а також цей плагін надасть змогу адміністраторам надалі прикладати менше зусиль до створення нових сторінок.

Таким чином за допомогою одного плагіну було покрито виконання декількох вимог до оновлення вебсайту кафедри програмної інженерії.

## 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

### 4.1. Охорона праці

Як і для всіх інших типів виробництва, для розробки програмного забезпечення також існують правила та норми охорони праці. Процес роботи з персональними комп'ютерами та ноутбуками корегуються нормами ДСанПіН 3.3.2-007-98 "Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин", НПАОП 0.00-1.28-10. Про затвердження правил охорони праці під час експлуатації електронно-обчислювальних машин [15], а умови праці та облаштування робочих місць корегуються розділами ДБН В.2.5-28:2016 та ГОСТ 12.1.005-88.

При роботі з комп'ютерами під час розробки веб-сайтів на WordPress, чи іншими методами, потрібне добре організоване робоче місце [16]. На робочому столі не повинно бути зайвих речей, за ним потрібно регулярно доглядати, витирати пил та протирати робочі поверхні, клавіатуру, мишку та екран спеціальними засобами. При роботі в офісі, кількість та відстань між робочими місцями розраховується в залежності від розміру приміщення. Площа одного робочого місця має становити не менше 6 квадратних метрів, а об'єм – не менше 20 метрів кубічних.

Крісла біля робочих місць повинні бути зручними та запобігати підтримці оптимальної робочої пози працівника, тобто такої, щоб працівник виконував свою роботу з мінімальним напруженням тіла та уникав перевтоми. Стіл повинен бути такої висоти, щоб було зручно користуватися клавіатурою протягом довгого часу. Екран повинен бути розміщений на рівні очей не ближче ніж 600 мм від обличчя працівника.

Приміщення має бути добре освітленим. В залежності від умов використовується природне, штучне та змішане освітлення. Природне освітлення повинно контролюватися з використанням жалюзі. Стіни фарбуються у світлі тони

для доброго світло-відбиття. Екрани моніторів розміщуються так, щоб природне світло з вікон падало на робочу поверхню збоку.

Робочі приміщення потрібно оснащувати системами опалення, штучним кондиціонуванням повітря та вентиляцією притяжного-витяжного типу. Працівники мають перебувати за комфортної температури, вологості та мати можливість провітрювати робочі кімнати.

Всі заземлені конструкції, що знаходяться в приміщенні, такі як водопровідні та газопровідні труби, опалювальні батареї та відкриті кабелі, повинні бути захищені захисними щитками або сітками з діелектричного матеріалу, щоб запобігти їх контактуванню з працівниками.

У випадку надмірного шуму чи вібрації комп'ютерного обладнання, робочі кімнати потрібно обладнати противібраційними килимками для забезпечення комфортних умов праці. Оскільки постійний шум та вібрація можуть спричинити швидке виснаження працівників, головні болі, а також це є сильний відволікаючий фактор, що заважає концентрації працівників.

Особливо важливим є дотримання правил протипожежної безпеки при організації роботи з великою кількістю комп'ютерного обладнання у великих та малих приміщеннях. Усі лінії електромережі повинні бути ізольовані та захищені від виникнення короткого замикання, а також від перепадів напруги. Ці фактори як мінімум можуть спричинити збої у системі чи передачі даних, або ж можуть призвести до надмірного нагрівання та навіть загорання комп'ютерного обладнання. Усі приміщення, окрім тих, у яких знаходяться робочі сервери, повинні бути оснащені системами протипожежної безпеки, а саме детекторами диму та вогню, системами автоматичної сигналізації та вогнегасниками. Вся робоча техніка повинна підключатися до електромережі лише за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. При цьому, крім фазового та нульового контактів провідників, у штепсельних з'єднаннях та електророзетках, мають бути підключені спеціальні нульові захисні провідники. Конструкція має бути такою, щоб приєднання нульового захисного

провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним.

Рівні впливу магнітних полів та електромагнітного випромінювання не повинні перевищувати норми. Адже саме вони дуже негативно впливають на центральну нервову систему людини та при передозуванні викликають головний біль, запаморочення та нудоту.

Кожного дня перед початком роботи всі розробники програмного забезпечення повинні:

- 1) Прибрати та підготувати робоче місце до робочого дня. Очистити екран та клавіатуру від накопиченого пилу.
- 2) Оглянути комп'ютерне обладнання та перевірити чи воно справне, чи не виявлено ознак короткого замикання, яке могло статися вночі.
- 3) Перевірити правильність освітлення робочого місця, запобігти наявності зустрічного світла та бликів на екрані робочого комп'ютера.
- 4) Перевірити правильність та безпечність підключення обладнання до електромережі.
- 5) При виявленні будь-яких неполадок, обов'язково повідомити про це системного адміністратора, або іншу відповідальну за це людину.

Під час робочого дня, всім розробникам програмного забезпечення рекомендується робити перерви та відходити від робочого місця на декілька хвилин для того щоб дати відпочити очам та нервовій системі. Раз в одну-дві години робити вправи на розминку очей, спини, плечей та рук. Проводити регулярні провітрювання приміщення декілька разів на день.

Правила дотримання техніки безпеки та інструкції з охорони праці потрібно проводити кожному новому працівнику компанії у його перший робочий день. За недотримання цих правил передбачаються штрафи та адміністративні стягнення.

Отже, складання та дотримання інструкцій з охорони праці є дуже важливим для організацій безпечної та комфортної роботи працівників з комп'ютерним обладнанням.



## 4.2. Безпека в надзвичайних ситуаціях

Під час роботи з комп'ютерами можуть виникати різні надзвичайні ситуації, такі як пожежа, ураження електричним струмом, коротке замикання, тощо. Працівникам компанії потрібно знати як вести себе у подібних ситуаціях та як запобігти негативних наслідків.

Пожежа може виникнути навіть у зовсім неочікуваний момент. Спричинити її може перенапруження системи, нагрівання та займання електричних приладів, таких як комп'ютер чи принтер, загорання розеток через коротке замикання, необережність працівників, тощо.

При виникненні пожежі основним є не допустити паніку серед працівників. Кожен повинен знати як вести себе у цій ситуації і діяти злагоджено. У кожному кабінеті повинен бути вогнегасник. Інструктаж з поведінки з вогнегасником потрібно проводити всім новим працівникам. При невеликому загоранні можна спробувати самому потушити полум'я вогнегасником, цупкою тканиною, або при наявності в офісі пожежного щитка з необхідним приладдям, можна використати пісок. Якщо загорівся електричний прилад чи розетка, ні в якому разі не можна тушити їх водою. Це може призвести до ураження електричним струмом, оскільки вода – чудовий провідник.

Вогнегасники є двох типів [17]. Першим є вуглекислотний вогнегасник типу ОУ-3 (ВВК-2). Цей пристрій забороняється використовувати для гасіння речей, які горять навіть без допуску повітря. Щоб розпочати гасіння потрібно зірвати пломбу і витягнути чеку з вогнегасника. Направити його так, щоб речовина покривала всю площу загорання. Тоді потрібно повернути спеціальний маховик або натиснути на важіль.

Викид речовини гасіння досягає 3 метрів, всередині вогнегасника вміщується до 4 літрів речовини з великим вмістом вуглекислого газу. Важливо дотримуватися температурного режиму при зберіганні, щоб поблизу не було обладнання опалення приміщень чи інших елементів, що мають здатність

нагріватися. Водночас місце зберігання вогнегасника повинно бути добре помітним та легкодоступним.

Другим видом є порошковий вогнегасник типу ОП-5. Важливо знати та пам'ятати як ним користуватися. Спочатку потрібно направити насадку на вогонь і витягнути захисну пломбу. Тоді натиснути на важіль і приступити до гасіння. Потрібно обов'язково врахувати температуру палаючих речей. Через на сьогодні існують хладонові речовини, що не псують поверхні дорогих матеріалів. Цей вид вогнегасників можна використовувати для гасіння електричних приладів, саме тому їх частіше встановлюють у приміщеннях де працюють з такою апаратурою.

Якщо ж полум'я поширюється швидко, потрібно негайно евакуювати всіх працівників та повідомити у службу надзвичайних ситуацій за номером 101. Евакуація повинна проходити чітко за схемою зазначеною на плані евакуації будівлі. Такий план повинен висіти при вході у кожен сектор офісу чи будинку. Під час евакуації потрібно захопити з собою необхідні речі, такі які не будуть заважати проходженню евакуації інших працівників, та спокійно вийти з будівлі не штовхаючи чи підбурюючи інших працівників.

У випадку короткого замикання, чи виходу з ладу будь-якого електроприладу є ризик ураження струмом [18]. При частій роботі з такими приладами також потрібно орієнтуватися як уникнути цього, чи що робити при ураженні.

Небезпечним є змінний струм при напрузі більше 50 В та постійний струм при напрузі більше 100 В. Ураження таким струмом є небезпечним для життя. Ризик ураження більший, якщо у людини мокра шкіра. Особливо небезпечним є проходження струму від руки до руки або від руки через спину в ноги.

Проходження струму через тіло спричиняє: опіки та електричні знаки в місці ураження, опіки або обвуглення шкіри, пошкодження нервів, м'язів (рабдоміоліз), зупинка серця, спазм коронарних артерій, переломи кісток та вивихи суглобів за рахунок скорочення м'язів, гостра дихальна недостатність через скорочення м'язів та пригнічення дихального центру, гостра ниркова недостатність. При напрузі більше 1000 Воль ризик тяжких опіків особливо великий.

У випадку ураження струмом людини потрібно в першу чергу вимкнути живлення приладу, якщо це можна зробити безпечно. Відтягнути потерпілого в безпечне місце, якщо впевнені, що немає ризику ураження для себе. Оцінити стан потерпілого. Викликати швидку допомогу за номером 103. При втраті свідомості розпочати реанімаційні заходи.

Заходи щодо усунення небезпеки ураження електричним струмом зводяться до правильного розміщення устаткування та електричних кабелів.

Рівні електромагнітного випромінювання та магнітних полів повинні відповідати вимогам ГОСТ 12.1.006. Електромагнітні поля біля комп'ютера (особливо низькочастотні) негативно впливають на людину і в першу чергу на її центральну нервову систему, викликаючи головний біль, запаморочення, нудоту. Причому нервова система реагує навіть на короткі за тривалістю впливи слабких полів: змінюється гормональний стан організму.

Нерухома напружена поза працівника призводить до втоми і виникнення болю в хребті, шиї, плечових суглобах. Інтенсивна робота з клавіатурою викликає болючі відчуття в ліктьових суглобах, передпліччях, зап'ястях і пальцях рук.

При тривалій та інтенсивній роботі за комп'ютером з'являється синдром комп'ютерного стресу (СКС), який проявляється:

- 1) головною біллю;
- 2) запаленням очей;
- 3) алергією;
- 4) дратівливістю;
- 5) млявістю і депресією;
- 6) погіршенням зосередженості і працездатності.

Причинами різноманітних симптомів СКС є 5 основних чинників:

- 1) неправильна робота очей і поза тіла;
- 2) носіння невідповідних окулярів або контактних лінз;
- 3) неправильна організація робочого місця;
- 4) розподілення фізичних, розумових, візуальних навантажень;
- 5) низький рівень візуальної підготовленості для роботи з комп'ютером.

При напруженій роботі за комп'ютером будь-який користувач щогодини повинен робити перерву на 15 хвилин через кожен годину і треба займатися іншою справою. Декілька разів на годину бажано виконувати серію легких вправ для розслаблення.

Режим праці та відпочинку при роботі з персональною електронно-обчислювальною машиною (ПЕОМ) залежить від категорії трудової діяльності. Всі роботи з ПЕОМ ділять на три категорії. Перша - епізодичне зчитування і робота з інформацією не більше 2-х годин за 8-годинну робочу зміну. Друга - зчитування інформації або творча робота не більше 4-х годин за восьми годинну зміну. Третя - зчитування інформації або творча робота тривалістю більше 4-х годин за зміну.

Якщо у приміщенні експлуатується більше одного комп'ютера, то треба врахувати, що на користувача одного комп'ютера можуть впливати випромінювання від інших, в першу чергу бокових, а також і задньої стінки сусіднього дисплея. Тому необхідний захист спеціальними фільтрами і щоб користувач розміщався від бічних і задніх стінок інших дисплеїв на відстані не ближче одного метра.

## ВИСНОВОК

Під час виконання атестаційної роботи було проведено дослідження сучасних методів створення веб-сайтів, виявлено основні інструменти та засоби для їх побудови і підтримки, описано плюси та мінуси кожної методології. Акцент дослідження було зроблено на розробку сайтів за допомогою систем менеджменту контенту (CMS), а особливо за допомогою WordPress.

Було детально вивчено концепцію безперервної інтеграції та безперервної доставки (CI/CD) програмного продукту. Розглянуто основні принципи цієї концепції, методи їх впровадження у роботу команди, плюси, мінуси, та основні інструменти для впровадження.

Також важливою частиною дослідження є використання систем контролю версій програмних продуктів. Впродовж виконання атестаційної роботи було проведено ознайомлення з видами цих систем, важливістю їх впровадження у робочі проекти. Було отримано практичні вміння застосовувати такі системи при роботі з веб-сайтами.

Метою та завданням атестаційної роботи було вдосконалення програмного забезпечення керування контентом сайту кафедри програмної інженерії типу WordPress в умовах обмежених інформаційних ресурсів.

Було проведено глибоке дослідження системи WordPress. Вияснено нюанси при роботі з нею на хостингу та на локальному сервері. Вивчено основи створення сайтів на WordPress з нуля а також адаптації і редагування існуючих. Проведено аналіз різноманітних тем (шаблонів) та плагінів даної системи. А також застосовано здобуті знання на практиці.

В результаті виконання атестаційної роботи було вдосконалено сайт кафедри програмної інженерії ТНТУ. Застосовано основні практики CI/CD. Встановлено даний веб-сайт на систему контролю версій, використовуючи програму GitHub та вбудований плагін Revisr. Адаптовано існуючу систему до перегляду на мобільних пристроях, а також полегшено процес створення сторінок сайту використовуючи плагін Elementor.

Внесені зміни дозволили зробити систему безпечнішою для редагування, адміністрування та модерації. Система контролю версій дозволить не перейматися про втрату даних та файлів, оскільки їх завжди можна буде відновити за рахунок резервних копій.

Спрощена система створення та редагування сторінок дозволить зекономити час та зусилля на створення зручних для перегляду та красивих статей. А отже, адміністратори отримають більше можливості на обдумування та саме оформлення контенту сайту.

Загалом отримана система відповідає поставленим вимогам та є зручною для використання в майбутньому. А за рахунок адаптивності, вона є прийнятною до впровадження нових змін та вдосконалень.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Дячук С.Ф., Малярський В.О., Кінах Я.І. Проектування програмних web-систем на основі використання засобів керування контентом // С.Ф. Дячук,, В.О. Малярський, Я.І. Кінах // Матеріали VIII науково-технічної конференції «Інформаційні моделі, системи та технології» 9-10 грудня 2020 р. - Тернопіль: ТНТУ, 2020.-162с.
2. The History of Website Design – Режим доступу: <https://historycooperative.org/web-design-through-the-years/>
3. Методи розробки Web-сайтів – Режим доступу: <https://sites.google.com/site/tz5103voinovakateryna/metodi-rozrobki-web-sajtiv>
4. HTML5 features , uses , advantages and disadvantages – Режим доступу: <https://www.online-sciences.com/computer/html5-features-uses-advantages-and-disadvantages/>
5. What is a Content Management System (CMS) | The Complete Guide – Режим доступу: <https://www.zesty.io/mindshare/marketing-technology/what-is-a-content-management-system-cms-the-complete-guide/>
6. Що таке CMS сайту – Режим доступу: <https://hostiq.ua/wiki/ukr/cms-ukr/>
7. What is a Web Framework – Режим доступу: <https://www.goodfirms.co/glossary/web-framework/>
8. Software as a Service (SaaS) – Режим доступу: <https://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service>
9. Переваги та недоліки CMS WordPress - Режим доступу: <https://esfirum.com/uk/blog/pros-and-cons-cms-wordpress/>
10. What is WordPress? An Overview of the World’s Most Popular CMS – Режим доступу: <https://www.hostinger.com/tutorials/what-is-wordpress>
11. WordPress vs Other Content Management Systems – Режим доступу: <https://www.namecheap.com/wordpress/wordpress-vs-other-content-management-systems/>
12. What Is CICD? What’s Important and How to Get It Right – Режим доступу: <https://stackify.com/what-is-cicd-whats-important-and-how-to-get-it-right/>

13. Getting Started - About Version Control – Режим доступу: <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>
14. Version Control Systems – All You Need To Know – Режим доступу: <https://scmquest.com/version-control-systems/>
15. НПАОП 0.00-1.28-10. Про затвердження правил охорони праці під час експлуатації електронно-обчислювальних машин – Режим доступу: [https://dnaop.com/html/31562/doc-%D0%9D%D0%9F%D0%90%D0%9E%D0%9F\\_0.00-1.28-10](https://dnaop.com/html/31562/doc-%D0%9D%D0%9F%D0%90%D0%9E%D0%9F_0.00-1.28-10)
16. Основні правила дотримання охорони праці при роботі на персональних ЕОМ – Режим доступу: <https://www.victorija.ua/dovidnik/osnovni-pravylyla-dotrymannya-ohorony-pratsi-pry-roboti-na-personalnih-eom.html>
17. Правила користування вогнегасником або як користуватись вогнегасником – Режим доступу: <https://nais.com.ua/ua/articles/kak-polzovatsya-ognetushitelem/>
18. Ураження електричним струмом – Режим доступу: <https://empendium.com/ua/chapter/B27.III.23.20>.



## ДОДАТКИ

## ДОДАТОК А

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ІВАНА ПУЛЮЯ  
КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

### ТЕХНІЧНЕ ЗАВДАННЯ

на розробку кваліфікаційної роботи

«Розробка шаблону сайту кафедри згідно вимог з можливістю адміністрування,  
модерації, конфігурування на WordPress»

Розробники:

виконавець ст. гр. СПм-61

Малярський Віктор Олегович

---

(підпис)

керівник роботи

Дячук Степан Федорович

---

(підпис)

Тернопіль 2020

## ЗМІСТ

1. ПІДСТАВИ ДО РОЗРОБКИ	3
2 ПРИЗНАЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	3
3 ВИМОГИ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ	3
3.1 Функціональні вимоги.....	3
3.2 Технічні вимоги .....	4
3.3 Програмні вимоги.....	4
4. ЕТАПИ РОЗРОБКИ	4
5. СУПРОВІДНА ДОКУМЕНТАЦІЯ	5
6. ПОРЯДОК ЗДАЧІ РОБОТИ	5
7. ВІДМІТКИ ПРО ВИКОНАННЯ ЕТАПІВ ТА ЗМІНИ В ПРОЕКТІ	6

## 1 ПІДСТАВИ ДО РОЗРОБКИ

Розробка проводиться у відповідності до графіку навчального плану підготовки магістрів за спеціальністю 121 «Інженерія програмного забезпечення».

Тема проекту: «Розробка шаблону сайту кафедри згідно вимог з можливістю адміністрування, модерації, конфігурування на WordPress».

Термін виконання: до « \_\_\_ » \_\_\_\_\_ 2020р.

## 2 ПРИЗНАЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Інформаційна система призначена для поширення інформації про кафедру програмної інженерії, оголошень та документів.

Інформаційна система буде корисною для викладачів та студентів кафедри та має інформаційне та освітнє призначення.

Інформаційна система дозволить зберігати та поширювати новини кафедри, важливу інформацію для викладачів, студентів та абітурієнтів.

## 3 ВИМОГИ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 3.1 Функціональні вимоги

Система повинна передбачати три ролі:

- Користувач
- Модератор
- Адміністратор

Для користувачів система надає доступ до:

- Публічних сторінок сайту
- Скачування публічних документів

Для модераторів система надає доступ до:

- Редагування записів на сторінках
- Створення нових записів
- Видалення записів

Для адміністраторів система надає доступ до:

- Майстерні системи WordPress, та керування виглядом сторінок

- Керування віддаленим репозиторієм
- Керуванням плагінами та темами системи

### 3.2 Технічні вимоги

Вимоги до клієнтської частини: ОС Windows, Mac, Linux, iOS, Android. Веб браузер Chrome, FireFox, Opera, Internet Explorer, Safari. Не перевантажений кеш браузера.

Для адміністратора: не менше 2 Гб внутрішньої пам'яті, локальний веб-сервер з налаштованим середовищем MySQL та система Git.

### 3.3 Програмні вимоги

Використання FTP клієнта: FileZilla.

Локальний сервер: Open Server

Розробка корневих файлів: PHP Notepad++.

## 4. ЕТАПИ РОЗРОБКИ

Розробка інформаційної системи проводиться в наступному порядку:

- аналіз системи WordPress;
- налаштування тестового середовища на локальному сервері;
- випробування технології контролю версій з використанням Git;
- налаштування плагіну Revisr та зв'язку з GitHub;
- впровадження нової системи розробки сторінок;
- тестування нових плагінів;
- оформлення супровідної документації;
- задача роботи.

Результати виконання кожного етапу проекту погоджуються з керівником роботи.

## 5. СУПРОВІДНА ДОКУМЕНТАЦІЯ

Для інформаційної системи повинні бути розроблені наступні документи:

- пояснювальна записка до роботи;
- презентація роботи;
- авторська довідка;
- диск з кодом інформаційної системи.

Пояснювальна записка до кваліфікаційної роботи оформляється згідно діючих вимог до нормоконтролю проектів.

## 6. ПОРЯДОК ЗДАЧІ РОБОТИ

Розроблена інформаційна системи повинна відповідати вимогами, що складаються з перерахованих у п.3.1 цього документу характеристик.

Для задачі проекту необхідно підготувати весь перелік документів зазначений у п.5 цього документу.

Приймання проекту проводиться спеціально створеною комісією в термін зазначені в п.1 цього документу.

## 7. ВІДМІТКИ ПРО ВИКОНАННЯ ЕТАПІВ ТА ЗМІНИ В ПРОЕКТІ

Назва етапу	Відмітка*
Аналіз системи WordPress	
Налаштування середовища розробки	
Розробка системи	
Впровадження та тестування системи	
Супровідна документація	

\* відмітки про виконання етапу ставляться керівником проекту

ДОДАТОК Б

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ІВАНА ПУЛЮЯ

МАТЕРІАЛИ

VIII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ  
МОДЕЛІ, СИСТЕМИ ТА  
ТЕХНОЛОГІЇ»**



9–10 грудня 2020 року

ТЕРНОПІЛЬ 2020



УДК 004.056.55

**С.Ф. Дячук, канд. техн. наук, доц., В.О. Малярський, магістрант, Я.І. Кінах, канд. техн. наук, доц.**

(Тернопільський національний технічний університет ім Івана Пулюя, Україна)

## **ПРОЄКТУВАННЯ ПРОГРАМНИХ WEB-СИСТЕМ НА ОСНОВІ ВИКОРИСТАННЯ ЗАСОБІВ КЕРУВАННЯ КОНТЕНТОМ**

UDC 004.056.55

**S.F. Dyachuk, Ph.D, Assoc. Prof., V.O. Malyarsky, undergraduate, I.I. Kinakh, Ph.D, Assoc. Prof.**

## **DESIGN OF SOFTWARE WEB-SYSTEMS USING CONTENT MANAGEMENT TOOLS**

Враховуючи сучасні тенденції, можна чітко зрозуміти, що проєктування та дизайн Веб- сайтів залежить від потреб та вподобань споживачів та клієнтів для яких вони створюються. Головною метою створення сторінки компанії є надання інформації про неї та про її послуги. Саме тому на сучасних сайтах практично немає нічого зайвого і все виконано у стилі мінімалізму.

Теперішнім користувачам не до вподоби, коли тяжко знайти потрібну їм інформацію серед великої кількості додаткових пропозицій або ж складного функціоналу сторінки. А це спричинить невдоволення та втрату коштів замовника. Тому сучасні сторінки повинні відповідати критерію User Friendly, тобто простими, швидкими, та зрозумілими у використанні. Для створення таких сайтів зараз існує низка засобів та технологій [1].

Класичне проєктування за допомогою HTML, CSS та JavaScript постійно розвивається та вдосконалюється за рахунок нових версій мови HTML з новими тегами, що забезпечують швидшу розробку розмітки, а також нових фреймворків JavaScript, таких як Angular.js, React.js, Vue.js, тощо. Їх використання забезпечує розширення можливостей додання зручного функціоналу, пришвидшення роботи сайту та його взаємодії з базами даних чи різними додатковими API. Також є програми, які допомагають генерувати код для створення сайтів. Серед таких програм є такі що одразу візуалізують написаний код - Adobe DreamWeaver, Microsoft FrontPage і такі що після генерації дозволяють редагувати код та автоматизувати його написання: Adobe HomeSites, HotDog, AceHTML [2].

Існують різні типи веб-фреймворків: Request-based, Component-based, Hybrid, Meta та RIA- based. Найпопулярнішими є Request-based, що безпосередньо обробляють вхідні запити та Component-based, які вбудовують обробку запитів всередині стандартних компонентів і самостійно стежать за станом. Більшість фреймворків працюють за схемою Модель-Вигляд- Контролер (MVC). Така структура допомагає розбити систему на самостійні блоки та спрощує контроль над ними.

Отже, сучасні засоби проєктування веб-сайтів дозволяють створювати раціональні за дизайном та функціональністю сторінки, що задовольняє вимоги більшості користувачів. Тому розвиток методології та удосконалення інструментів проєктування Web-систем на основі використання засобів програмного керування контентом є гостро актуальною задачею.

### **I Література.**

1. Юдін О.К. Кодування в інформаційно-комунікаційних мережах: – Монографія. - К.:НАУ, . 2007.-308с.
2. Методи розробки Web-сайтів – Режимдоступу: [https://sites.google.com/site/tz5103\\_voinovakateryna/metodi-rozrobki-web-sajtiv](https://sites.google.com/site/tz5103_voinovakateryna/metodi-rozrobki-web-sajtiv).

**ДОДАТОК В**