Міністерство освіти і науки України
Тернопільський національний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Центр перепідготовки та післядипломної освіти
(повна назва факультету)
Кафедра програмної інженерії
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістр
(назва освітнього ступеня)

на тему: **Розробка структури та наповнення сайту підрозділу університету з використанням методів прямого редагування**

Виконав: студент (~~ка~~) 2 курсу, групи СПд-2
спеціальності (напряму підготовки)
121 «Інженерія програмного забезпечення»
(шифр і назва спеціальності (напряму підготовки))

		Гамулець Б.С.
	(підпис)	(прізвище та ініціали)
Керівник		Бойко І.В.
	(підпис)	(прізвище та ініціали)
Нормоконтроль		
	(підпис)	(прізвище та ініціали)
Рецензент		
	(підпис)	(прізвище та ініціали)

м. Тернопіль – 2020

АНОТАЦІЯ

Розробка структури та наповнення сайту підрозділу університету з використанням методів прямого редагування // Дипломна робота // Гамулець Богдан Степанович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно – інформаційних систем та програмної інженерії, кафедра програмної інженерії, група СПд-2, Тернопіль, 2020 // с. – 78, рис. – 43, табл. - 5, аркушів А1 - 11, бібліогр. – 17.

Ключові слова: ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, WORDPRESS, СИСТЕМА КЕРУВАННЯ КОНТЕНТОМ, PYTHON, ВЕБСАЙТ, JAVASCRIPT, HTML, CSS

Кваліфікаційну роботу магістра присвячено дослідженню методів та засобів для розробки шаблону сайту кафедри на WordPress згідно вимог з можливістю адміністрування. У даній дипломній роботі магістра розроблена та проведена аналітика наукових досліджень, технічних специфікацій та спеціалізованих статей, у результаті чого була доведена новизна та актуальність науково-дослідної роботи. Обґрунтовано доцільність та практичність використання WordPress для розробки шаблону сайту кафедри, який включатиме у власний функціонал можливість адміністрування. У кваліфікаційній роботі магістра досліджено програмні засоби для розробки та планування архітектури веб-сайту, методом узагальнення проаналізованої інформації для реалізації поставленої цілі було використано наступні технології: Python, XPM-RPC, HTML/CSS, WordPress та XAMPP. Сплановано та створено шаблон сайту кафедри, методом експерименту та підсумування доведено наукову новизну та актуальність дослідження, зокрема, покращення адміністрування та можливість конфігурації контенту для наукових співробітників, викладачів та студентів.

ANNOTATION

Development of the department site according to the requirements with possible administration. // Thesis // Hamulets Bohdan Stepanovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, group SPd-2 // Ternopil, 2020 // p. – 78, pic. – 43, tab. - 5, sheets A1 - 11, bibliography – 17.

Keywords: SOFTWARE, WORDPRESS, CONTENT MANAGEMENT SYSTEM, PYTHON, WEBSITE, JAVASCRIPT, HTML, CSS

The Master’s qualification work is devoted to the research of the methods and means for development of a template of a site of a scientific department on a WordPress according to the requirements with a possibility of administration, moderation, configuration. In this master's thesis the analysis of scientific researches, technical specifications and specialized articles is developed and carried out, as a result of which novelty and urgency of research work was proved. The expediency and practicality of using WordPress to develop a template for the department's website, which will include in its own functionality the ability to administer, moderate and configuration all in one. In this qualification work of master’s degree were investigated software tools and features for development and planning of a website architecture, by a method of generalization of the analyzed information for realization of the set purpose the following technologies were used: Python, XPM-RPC, HTML/CSS, WordPress and XAMPP. The template of the site of the department is planned and created, the scientific novelty and relevance of the research is proved by the method of experiment and summing up.

Зміст

ВСТУП

Актуальність теми.

Вебсайт у загальному – це відмінний інструмент для ведення будь якої діяльності або бізнесу, збору інформації про потенційних клієнтів, залучення нових клієнтів та управління різними бізнес-процесами.

Вебсайт кафедри необхідний для можливості залучати нових студентів, інформувати про зміни в роботі та актуальну інформацію існуючих, робити публікації новинних розсилок, наукових статей, здійснювати різні маркетингові та рекламні компанії для залучення більшої кількості потенційних абітурієнтів, проводити для наукових співробітників та членів команди колективу викладачів та професорів різноманітні семінари, з публікаціями про різноманітні методи та підходи до організації процесу навчання. Сайт кафедри це також і зручна соціальна корпоративна мережа для співробітників, яка дає можливість без обмеження спілкуватись, обмінюватися різною інформацією, розміщувати новини компанії і всілякі навчальні матеріали.

Створити сайт для кафедри зараз можуть різні компанії, але оскільки це трудомісткий та ресурсно затратний процес, він займає чимало часу та коштів на реалізацію. Винаймати сторонніх розробників не являється оптимальним рішенням оскільки знання про специфічні потреби і специфіку функціонування веб сайту доведеться у будь якому випадку. Зважаючи на те що у сучасному світі завдання з розробки веб сайту можна реалізувати самостійно, шляхом ресурсів технології WordPress, це завдання краще виконати самостійно, одночасно маючи абсолютно повний доступ та вплив як на процес і функціонал, так і на ресурси за допомого яких цей вебсайт

буде розміщений у мережі інтернет, зокрема: хостинг, серверна операційна система, клауд платформа для розміщення веб сайту, власна база даних та доступ для оновлення програмного забезпечення усіх компонентів. Оскільки при виконанні даного завдання дуже важливу роль також відіграє захищеність персональних даних та інтелектуальних прав власності, кращого способу ніж реалізувати цей процес самостійно з нуля, контролюючи кожен процес імплементації під час її виконання, не знайти.

Мета дослідження. Метою дослідження є аналіз розробки шаблону сайту кафедри на WordPress згідно вимог з можливістю адміністрування.

Основні завдання дослідження:

- провести аналіз наукових публікацій та існуючого програмного забезпечення для аналізу необхідних технологій;
- розглянути історію, мету та шлях розвитку WordPress у сучасному контексті;
- проаналізувати способи реалізації адміністрування, модерації та конфігурування вебсайту;
- розробити та випробувати сайт кафедри на WordPress.

Об’єкт дослідження - процес розробки шаблону вебсайту кафедри.

Предмет дослідження - моделі, методи та засоби представлення та розробки серверної частини вебсайту кафедри на WordPress.

Методи дослідження. Для вирішення поставлених задач використано наступні методи:

- проектування та програмування – при використанні обраних мов програмування, фреймворків та бібліотек для створення програмного забезпечення;
- формалізації – при обґрунтуванні клієнт-серверної архітектури для розробки комп’ютерної системи для аналізу та візуалізації даних;
- аналіз та узагальнення – при проведенні аналізу існуючих конкуруючих технологій для розробки вебсайтів, а також сучасного стану та філософії розробки програмного забезпечення;

Наукова новизна одержаних результатів полягає у вирішенні науково-практичної розробки шаблону сайту кафедри на WordPress згідно вимог з можливістю адміністрування, модерації, конфігурування, при цьому отримано наступні результати:

- детально обґрунтовано засоби реалізації технічних рішень що дають змогу публікувати довільний вміст вебсторінок з розміщенням медіа, аудіо та відео файлів, а також текстових постів чи документів;
- показано доцільність використання найновішої версії програмного забезпечення WordPress та роботи з ним через XML-RPC інтерфейс.

Практичне значення одержаних результатів. Практична цінність отриманих під час дослідження результатів полягає у тому, що спроектована та розроблена система дає змогу робити публікації новинних розсилок, наукових статей, здійснювати різні маркетингові та рекламні компанії для залучення більшої кількості потенційних абітурієнтів, проводити для наукових співробітників та членів команди колективу викладачів та професорів різноманітні семінари, з публікаціями про різноманітні методи та підходи до організації процесу навчання.

Публікації. Результати дослідження апробовано на 7 науково-технічній конференції Тернопільського національного технічного університету імені Івана Пулюя “Інформаційні моделі, системи та технології” (24 грудня 2020 року) у вигляді тез конференцій.

Структура роботи. Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка складається із вступу, трьох розділів, висновків та бібліографії та додатку. Обсяг роботи: пояснювальна записка – 78 аркушів формату А4, графічна частина – 11 аркушів формату А1.

РОЗДІЛ 1
ІСТОРІЯ ВИНИКНЕННЯ ТА АНАЛІТИКА РОЗВИТКУ ТЕХНОЛОГІЙ ПРОЕКТУ WORDPRESS

1.1. Визначення технології WordPress та історія її виникнення

WordPress бере свій початок з технології ‘b2/cafelog’, більш відомої як b2 або cafelog, саме вона стала попередником і натхненником створення WordPress. Станом на Березень 2002 року вона була реалізована шляхом мови програмування PHP для використання разом з базою даних MySQL і налічувала близько двох тисяч активних блогів. Створена ця система з менеджменту контенту (CMS – Content Management System) інженером Міхаелем Валдріні (Michel Valdrighi), який зараз займає посаду одного з головних розробників новішої та покращеної версії цього програмного забезпечення - WordPress.

WordPress вперше почав своє існування у 2003 як результат успішної колаборації зусиль Міхаела Валдріні та Майка Літтла, для створення окремого, покращеного розгалуження цього програмного забезпечення.

У 2004 році відбувся швидкий ріст кількості активних користувачів WordPress, не лише через вдалі продумані рішення які були прийняті в процесі його розробки, також посприяла ситуація зі зміною ліцензії на використання тодішнього конкурентного продукту, програмного забезпечення Movable Type. Компанія власник Six Apart змінила умови використання у безкоштовній версії, що в кінцевому випадку стало причиною для багатьох найбільш авторитетних гравців у середовищі розробки веб застосунків перейти на використання WordPress.

Уже у жовтні 2009 дослідження проведене Open Source CMS MarketShare заключила висновок про домінування технології WordPress на ринку створення та підтримки веб сайтів. За її даними користувачі були набагато більш задоволені можливостями для конфігурації, модерації та адміністрування власних вебсайтів саме з WordPress.

Станом на червень 2019, WordPress використовують 60.8% з усіх відомих вебсайтів. Це складає 27.5% з топ 10 мільйонів вебсайтів. Отже, як висновок можна сміливо підсумувати абсолютне домінування даної технології на ринку, що свідчить про те, що для більшості користувачів воно підходить найкраще. Як саме вдалося добитись таких результатів та які саме особливості функціонування даного програмного забезпечення завойовуються таку палку прихильність її користувачів розглянемо далі.

Отже, WordPress це Content Management System (CMS), система менеджменту контенту, яка допомагає блогерам та веб розробникам модифікувати та модерувати контент у повсякденному користуванні, без потреби у використанні традиційного програмного забезпечення з редагування HTML коду, таких як наприклад Dreamweaver, Frontpage або Sublime.

WordPress функціонує за принципом стандартної ліцензії Open Source, зокрема під GPL2, що означає те що дане програмне забезпечення є безкоштовним та дозволяє розробникам модифікувати та змінювати його код на власний розсуд без жодних лімітацій чи зобов’язань.

Фундаментально існує два способи використання WordPress, або вибирати опцію розміщувати цей пакет програмного забезпечення для створення веб сайту на спеціальній платформі, яка самостійно підтримуватиме, обновлятиме та забезпечуватиме безперебійне функціонування вебсайту на своїх серверах, з використанням власної бази даних (але в такому випадку розробник отримує набагато менше контролю за власним вебсайтом), тоді слід вибирати WordPress.com.

Або самостійно завантажити цей пакет програмного на власний персональний комп'ютер, внести необхідні модифікації, розробити власну тему з можливістю модерування, конфігурації та адміністрування веб сайту (наш випадок), тоді підняти власний сервер у хмарному сховищі, встановити веб сервер, і власну базу даних, в такому випадку слід використовувати ресурс WordPress.org.

WordPress.org містить безкоштовну версію для self-hosted (тобто власного розміщення) серверного програмного забезпечення на таких хостинг клауд платформах як: Digital Ocean (DO), Amazon Web Services (AWS), Microsoft Azure, Google Cloud Software (GSP) або інших схожих операторів, яких на ринку чимало. Вибір конкретного хостинг провайдера залежить від специфікації потреби веб розробника. Базуючись на серверній потужності, орієнтовної очікуваної кількості користувачів, потреби у безперебійному функціонуванні, кількості очікуваного використовуваного інтернет трафіку, географічного регіону розташування серверу, ціни відрізняються. Оптимальним варіантом вибору буде порівняння характеристик ціни і якості наданих послуг з розміщення веб серверу. Найоптимальніший баланс ціни якості можуть запропонувати найбільші гравці на ринку, ціна у яких може бути дещо ви0щоб, зате технічна підтримка та якість у наданні послуг буде на найвищому рівні. що з часом допоможе уникнути потенційних проблем які могли б привести до втрати грошей у випадку більш дешевшого, та менш якісного хостинг провайдера. Ціни на серверні потужності вимірюються переважно у місячній або погодинній підписці функціонування серверу, в кінці платіжного періоду надходить інвойс з необхідними реквізитами для оплати.

Перевагою власного хостину веб сайту являється доступ до код бази серверу та самого WordPress "під капотом", а також приблизно 56 тисяч безкоштовних плагінів та більше 2500 безкоштовних тем, усі з яких розміщені за безкоштовною ліцензією WordPress.org – і навіть декілька з інших комерційних джерел. Усі вони розміщуються під GPL ліцензією. так

само як і WordPress (WP), це надає змогу власникам вебсайтів користуватися великим обсягом доступних ресурсів швидко і безкоштовно. Більше інформації про плагіни і теми для WP подано у наступних розділах роботи.

1.2. Шлях розвитку технології Wordpress та її стан у сучасному контексті розробки програмного забезпечення

В процесі свого розвитку, починаючи з найбазовіших версій, великим товариством розробників та ентузіастів даного програмного забезпечення постійно вносились покращення у існуючий функціонал, а також розроблялись та додавались нові функції. В цьому процесі окрему увагу варто приділити версії **WordPress 3.0** з кодовим ім'ям **Thelonious**, опубліковану у липні 2010 року. Включення у базовий функціонал можливості додавання кастомізованих постів відкрила ворота для усіх користувачів до кастомізації типу контенту який можна розмістити на платформі WordPress. Раніше цей функціонал обмежувався лише постами, сторінками та таксономіями (posts, pages і taxonomies), а з цією зміною стало можливим створювати пости наприклад для ресторанів, використовуючи власно описаних полів з безліччю описаних додаткових структур даних типу словник (ключ: значення) як атрибуту певного обєкту. Ця зміна стала різницею між WordPress та іншими системами з менеджменту коненту.

Шляхом представлення підтримки власного API (Application Program Interface) керування розгорнутими можливостями кастомізації власного вебсайту які надає WordPress позволили програмно, динамічно підганяти цей продукт під специфічні потреби практично будь-якого бізнесу чи організаційної структури.

Оновлення WordPress до версії 3.1 (кодова назва Reinhardt) було опубліковане у лютому 2011 року. В ньому були виправлені певні неточності та помилки, покращений зовнішній вигляд та додана додаткова документація яка відповідала на найбільш часті запитання користувачів. Таки чином

охоплення WordPress вже тоді становило цілих 12% з усіх вебсайтів світу. Дуже швидко кількість піднятих блогів виросла до 50 мільйонів, а згодом, через 3 роки ця частка становила уже 22%.

На момент написання цієї дипломної роботи реліз останньої, найновішої версії відбувся у серпні 2020 року, саме цю найновішу версію програмного забезпечення - **5.5** з кодовою назвою **“Eckistine”** було обрано для розробки шаблону сайту кафедри на WordPress згідно вимог з можливістю адміністрування, модерації, конфігурування для даної дипломної роботи. Таким чином можна стверджувати про актуальність та затребуваність даної теми, а також те що знання здобуті в наслідок її розробки залишатимуться актуальними ще довгий період часу та надають змогу професійного розвитку, а також забезпечать хорошу позицію спеціаліста на ринку праці, який у випадку розробки веб сайтів, являється глобальним.

1.3. Аналітика функціоналу та порівняння в підходах архітектури і дизайну WordPress у зрівнянні з найвагомішими конкурентними технологіями

Оскільки роль яку виконує WordPress у процесі розробки та підтримки вебсайтів є надзвичайно затребуваною в сучасному світі, дана система з менеджменту контенту не являється єдиною технологією на ринку, здатною виконувати цю задачу, навпаки у неї є достатньо широке коло конкуруючих технологій, здатних повторювати як якусь певну частину її функціоналу, так і стати повною заміною для WordPress. У даній дипломної роботі розглянемо основних конкурентів, з найбільшою часткою використання на світовому ринку, з невеликим за обсягом визначенням та характеристикою їх особливостей, таким чином підтверджуючи тезис про вірне використання саме технології WordPress для розробки шаблону сайту

кафедри згідно вимог з можливістю адміністрування, модерації, конфігурування. Основними конкурентами для WordPress являються: Bootstrap, Lavarel, Drupal та Django.

- **Bootstrap** розробили Марк Отто та Джейкоб Торнтон у Twitter. Він був опублікований, як програмний продукт з відкритим вихідним кодом у серпні 2011 року на GitHub (система контролю версій коду). Налаштування компонентів Bootstrap включає менше змінних та плагінів jQuery, для отримання власної кастомізованої версії. Bootstrap містить понад десяток користувацьких плагінів jQuery, які легко використовувати (усі разом або індивідуально). Bootstrap містить понад десяток компонентів багаторазового використання, побудованих для забезпечення навігації, іконографії, попереджень, спливаючих вікон, випадаючих меню, тощо. Bootstrap надає базову структуру із системою сітки, стилями посилань та фоном. Підтримується багатьма популярними браузерами. Порівняльна характеристика WordPress із Bootstrap наведена у табл. 1.1.

Таблиця 1.1

Порівняння Bootstrap і WordPress

Характеристика	**Bootstrap**	**WordPress**
Спосіб використання	Використовується у веб розробці, надає красивого вигляду сторінкам HTML	Для побудови вебсайтів достатньо лише здійснювати кастомізовані налаштування плагінів.
Переваги	Покращує зовнішній вигляд	Сайт можна налаштувати за декілька годин
Практичне застосування	Доповнення до стеку технологій виключно веб розробки.	Повноцінне розміщення блогів та вебсайтів.

- **Lavarel** був створений і розроблений Тейлором Отуеллом. Перша версія даного програмного забезпечення була опублікованою у в 2011 році.

Laravel був випущений із вбудованими функціями для автентифікації користувачів (як звичайних, так і з правами адміністрування), локалізації (тобто можливості швидкого перекладу та підтримки різних мов), зовнішніх моделей, переглядів, сеансів, маршрутизації та інших функцій, таких як інверсія управління та система шаблонів під назвою Blade. Нові можливості інтерфейсу командного рядка під назвою artisan, вбудована підтримка системи управління базами даних, підтримка обробки подій та система упаковки під назвою bundles. Порівняльна характеристика WordPress із Lavarel наведена у табл. 1.2.

Таблиця 1.2

Порівняння Lavarel і WordPress

Характеристика	**Lavarel**	**WordPress**
Визначення	PHP веб фреймворк для веб розробки, з відкритим вихідним кодом	Система менеджменту контенту, з відкритим вихідним кодом
Переваги	Надає набагато більший функціонал, та в це також і недолік через високу складність та заплутаність різних компонентів.	Сайт можна налаштувати за декілька годин
Практичне застосування	Доповнення до стеку технологій виключно веб розробки.	Повноцінне розміщення блогів та вебсайтів.

- **Drupal** - також являється системою з управління вмістом наповнення вебсайту, і також як і WordPress являється пакетом програмного забезпечення з відкритим кодом. Він також називається структурою управління вмістом, програмним забезпеченням для блогів та вебсторінок. Його розробив Дріс Буйтерт. Drupal був випущений ще у 2000 році, а його найактуальніша версія на момент написання дипломної роботи була опублікованою у Червні 2020 року, версія 9.0. Він підтримує як операційну

систему Unix-подібних архітектур, так і Windows. Створений на мові програмування PHP, та доступний для використання багатьма іншими мовами. Основним його функціоналом являється, “Core”, що забезпечує широку підтримку плагінів та функцій. Drupal базується на основних модулях, темах, має підтримку автоматичних повідомлень про доступні оновлення, абстракції бази даних та високій доступності. Основний модуль забезпечує різні функції, такі як розширений пошук, книги, блоги, коментарі, багаторівнева система меню, підтримка декількох сайтів на одному хостів, забезпечення безпеки, надає інструменти робочого циклу, профілі користувачів, статичний доступ та реєстрацію. Основна тема забезпечує професійне відчуття та зовнішній вигляд веб-сайтів та блогів. Порівняльна характеристика Drupal із WordPress наведена у табл. 1.3.

Таблиця 1.3

Порівняння Drupal і WordPress

Характеристика	**Drupal**	**WordPress**
Спосіб реалізації	Drupal написаний за допомогою PHP фреймворку Symphony.	Wordpress реалізований за допомогою PHP та базується на базі даних MySQL.
Архітектура	Архітектура типу Front Controller.	Архітектура типу Presentation abstraction Control.
Підтримка локалізації	Програмний продукт доступний різними мовами світу.	Програмний продукт доступний лише англійською.
Складність	Функціонал набагато ширший та складніший у порівнянні з WordPress	Функціонал спрощений по максимуму.
Практичне застосування	Великі вебсайти зі значним обсягом функціоналу та контентом	Компактні вебсайти або блоги.
Інтерфейс для користувача	Складніший User Interface у порівнянні з WordPress	Простий, User friendly інтерфейс

- **Django** був створений восени 2003 року, коли веб-програмісти газети Lawrence Journal-World, Адріан Головатий і Саймон Віллісон, почали використовувати Python для створення веб додатків. Також достатньо рано у циклі розробки програмного забезпечення до команди розробників приєднався Якоб Каплан-Мосс. Django був опублікований у системи публічні системи керуванням вихідним кодом лише у липні 2005 року, за типом ліцензії відкритого коду BSD (Berkeley Software Distribution). Цей фреймворк був названий на честь гітариста Джанго Рейнхардта. У червні 2008 року було оголошено, що нещодавно утворений Фонд програмного забезпечення Django (DSF) займатиметься підтримкою та адмініструванням фреймворку Django в майбутньому. Django - це система веб-розробки, яка допомагає створювати та підтримувати веб-сайти. Це веб-фреймворк Python високого рівня, який допомагає усунути повторювані завдання, що спрощує веб-розробку. На момент написання даної дипломної роботи, найновіша версія була опублікованою у Серпні 2020 року з маркуванням 3.1

Django є надзвичайно надійним програмним продуктом, що підтверджується тим що його саме використовують найбільші веб-сайти у світі - **Instagram**, **Pinterest**, **Bitbucket** тощо. Django також має вбудований легкий веб-сервер для розробки та тестування додатків на ходу. Він також підтримує Apache та інші популярні веб-сервери. Порівняльна характеристика WordPress із Bootstrap наведена у табл. 1.4.

Таблиця 1.4

Порівняння Django і WordPress

Характеристика	**Django**	**WordPress**
Спосіб використання	Комплексний продукт для високонавантажених систем	Створення простих вебсайтів та блогів

Популярність	Менш популярний оскільки застосовується для досить обмеженої кількості завдань	Надзвичайно популярний з використанням у всіх точках світу
Простота у використанні	Вимагає команди професійних розробників	Сайт можна налаштувати за декілька годин

1.4. Аналіз базового процесу встановлення WordPress

Для швидкого старту з WordPress спершу слід розгорнути його на локальному серидовищі для розробки і тестування, а після повної готовності вебсайт можна перенести в публічний доступ шляхом розгортання готового продукту на публічному хостингу.

Для встановлення WordPress у даному випадку на локальний компютер з операційною системою MacOS необіхдно виконати:

1) Завантажити пакет програмного забезпечення WordPress з вебсайту worpress.org

2) Завантажити простий вебсервер який міститиме в собі тестову базу даних для початку роботи, у даному випадку було обрано XAMPP (абревіатура від Cross-platform, Apache, MariaDB(Mysql), PHP and Perl). Оскільки WordPress не є окремим додатком, XAMPP забезпечує два найважливіші компоненти для його встановлення - Apache, який використовується для створення локального сервера, та MySQL, який ви можете використовувати як базу даних для свого веб-сайту.

Можливо, вам цікаво, чому і як розробники використовують локальний сервер WordPress. Відповідь проста - це дозволяє їм створити локальну копію сайту, на якому вони зможуть випробувати нові оновлення плагінів, перш ніж впроваджувати їх у його реальній версії. Таким чином вони можуть запобігти та виявити потенційні помилки та проблеми, які можуть виникнути.

3) Використовуючи XAMP створити тестову базу даних та тестового користувача з правами адміністратора, у нашому випадку це:

define('DB_NAME', dplm);

define('DB_USER', 'tester');

define('DB_PASSWORD', 'test123');

define('DB_HOST', 'localhost');

4) Розархівувати файл з пакетом програмного забезпечення WordPress завантажений у першому кроці у папку з веб сервером XAMPP

5) Відкрити веб сторінку за посиланням http://localhost:8080/testwp/wp-admin/setup-config.php, і слідувати інструкціям зображеним на ній, рис. 1.1.

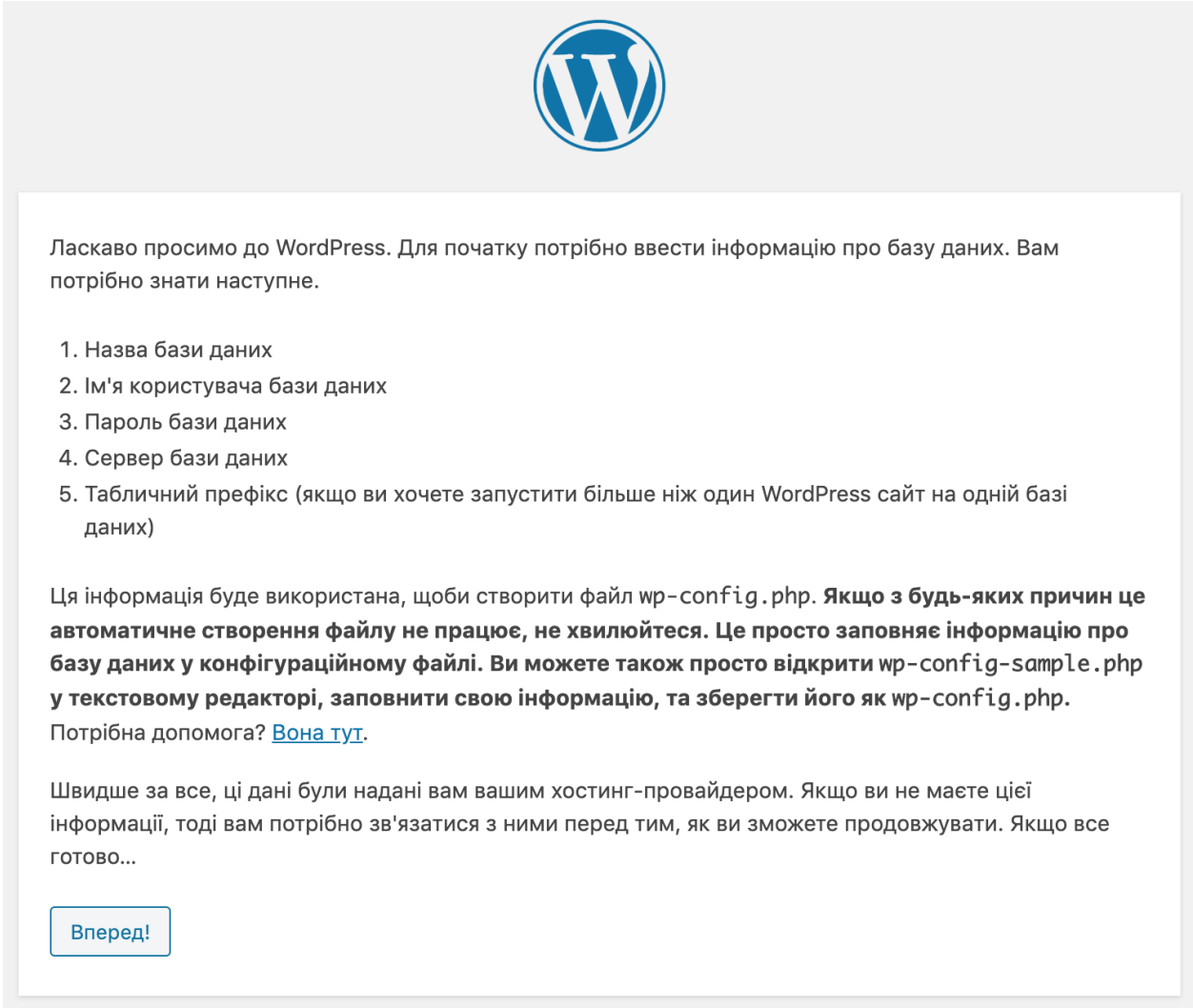

Рис. 1.1. Початкова сторінка WordPress з інструкціями для встановлення

6) Внести наступну конфігурацію в другий крок налаштування WordPress: (як показано на рис. 1.2)

Назва бази даних: **dpml**

Ім’я користувача: **tester**

Пароль: **test123**

Хост бази даних: **localhost**

Табличний префікс: **wp-config.php**

Рис. 1.2. Налаштування WordPress даних

7) Заповніть наступне вікно конфігурації, зображене на рис1.3.:

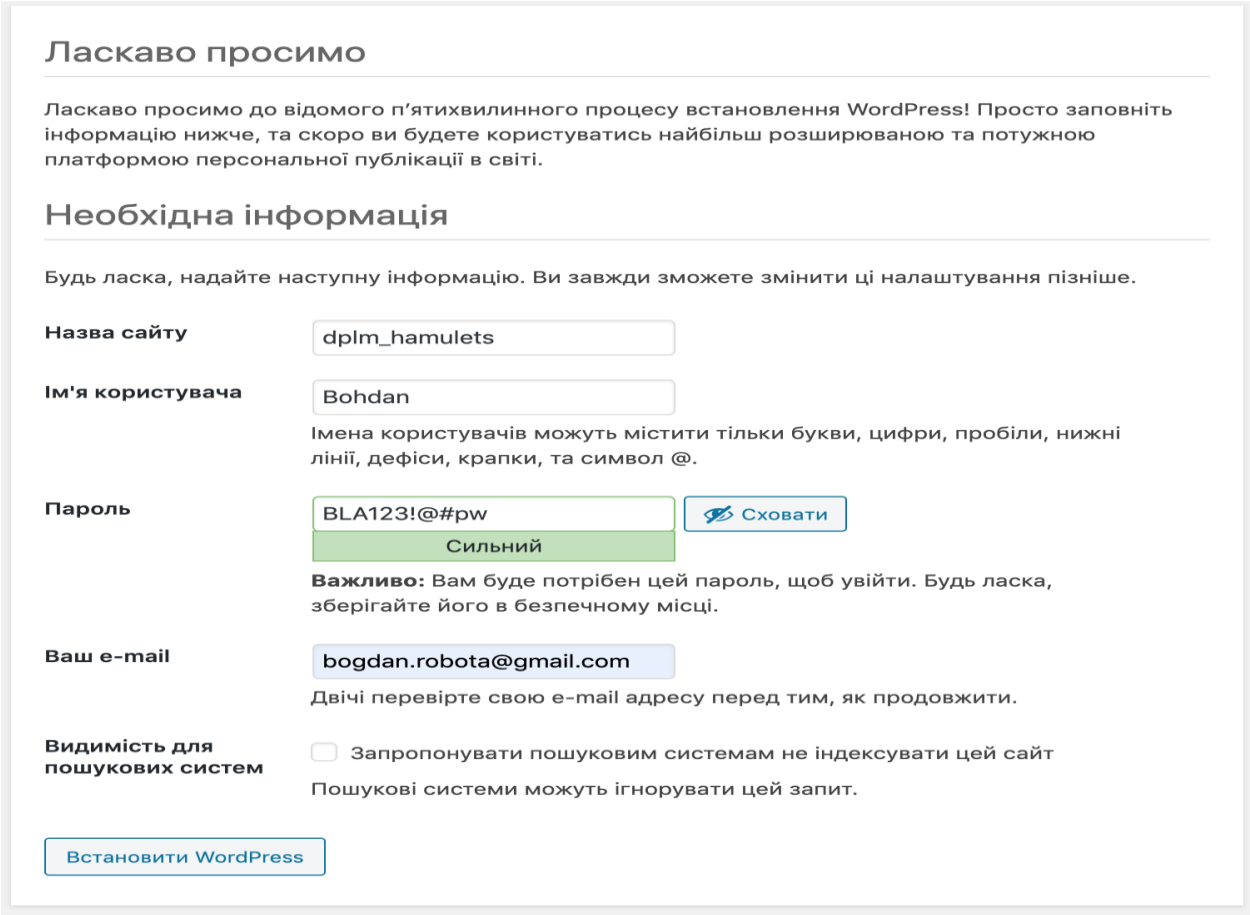

Рис. 1.3. Налаштування вебсайту розміщеного за допомогою WordPress

8) У випадку успішного виконання усіх попередніх кроків, ви зможете залогуватись у систему WordPress (з попередньо вказаними ім'ям користувача та паролю) як адміністратор вебсайту, і перед вами відкриється панель адміністрування, зображена на рис. 1.4.:

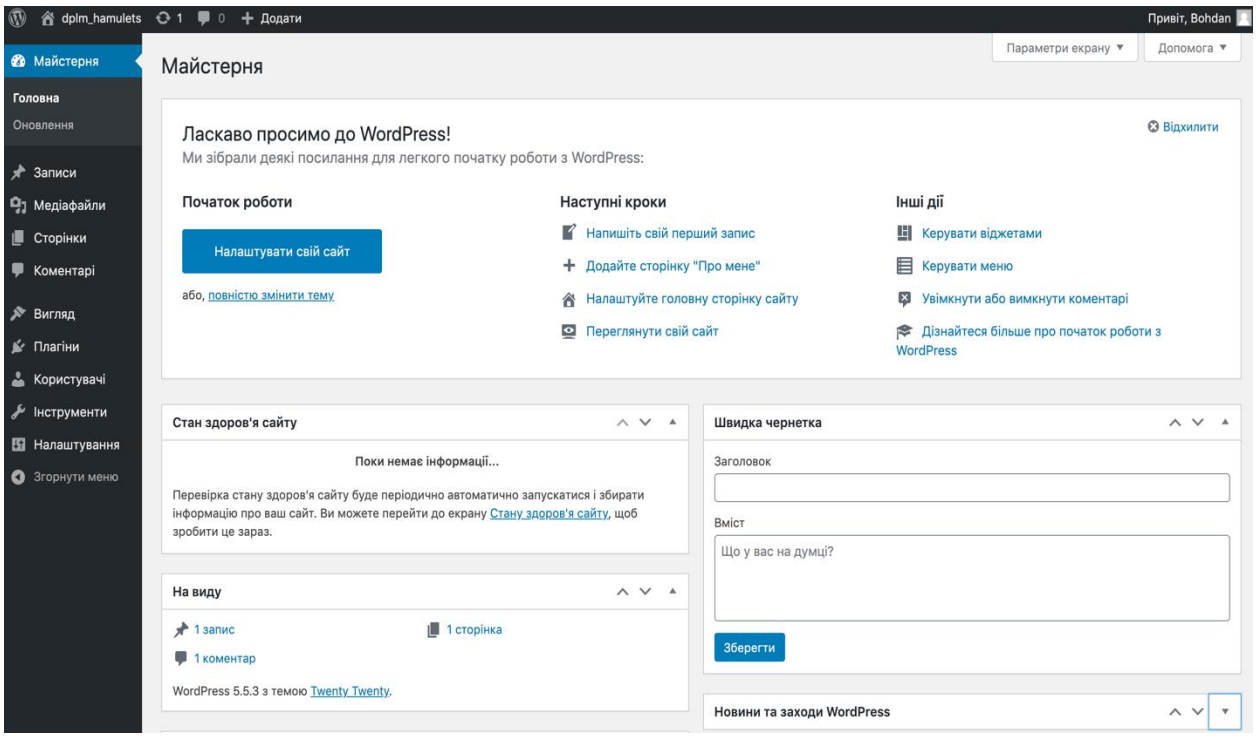

Рис. 1.4. Вікно «Майстерня» WordPress.

1.5 Висновки до розділу

Даний розділ розпочався з визначення програмного забезпечення WordPress, опису історії його виникнення, та хід процесу його розробки. Далі було розглянуто найбазовіші сценарії його використання та дана характеристика типу проблем які дане програмне забезпечення допомагає вирішити. Проаналізувавши відповідну літературу та зібрану статистику з використання було доведено актуальність та затребуваність роботи на дану тему. Для доведення правильності вибору саме WordPress для розробки шаблону сайту кафедри згідно вимог з можливістю адміністрування.

Далі було розглянуто декілька найбільш поширеніших аналогів WordPress: Bootstrap, Lavarel, Drupal та Django для визначення особливостей кожної з даних технологій, аналізу переваг та недоліків кожної з них, що надало нам змогу впевнено стверджувати у правильному виборі, і повному задоволенню технічним вимог для даної дипломної роботи саме WordPress.

Наступним логічним кроком став аналіз ресурсів та допоміжних програм для встановлення WordPress щоб мати можливість розпочати розробку. Детально наведені кроки, необхідні для конфігурації налаштувань, розглянуті вище з успішним результатом – WordPress встановлено на локальний комп'ютер, веб сервер піднятий на локальному хості, база даних працює коректно. У нас є всі необхідні ресурси для продовження роботи.

РОЗДІЛ 2
ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ ШАБЛОНУ САЙТУ

2.1. Мова програмування Python: історія, версії та переваги.

Мова програмування **Python** (пайтон) була зароджена наприкінці 1980 року, а її імплементація (втілення в життя) почалось у 1989 році інженером Guido van Rossum з Нідерландів, як логічне продовження кар'єрного шляху після роботи з іншими мовами програмування на роботою над спеціалізованою операційної системою Amoeba operating system.

У лютому 1991 року Van Rossum опублікував першу версію з маркуванням 0.9.0 до alt.sources. В ній уже були присутніми такі найбазовіші типи даних як list, dict, str та loop. Для виняткової моделі архітектури пайтон був створений інтернет форум comp.lang.python де відбувалась початкова дискусія стосовно Python, способу його розвитку та стала першим значним стрибком у суттєвому збільшенні кількості користувачів цієї мови програмування.

Першим найбільш суттєвим релізом для цієї мови програмування стала версія Python 2.0, що вийшла у Жовтні 2000 року в моделі безкоштовної мови програмування (Open Source програми, тобто програми з відкритим джерелом та вихідними файлами з кодом які доступні для перегляду та інспекції кожному користувачеві), з багатьма ключовими функціональними принципами, включаючи системний процес відстеження циклів garbage collector яка спростила менеджмент пам'яті операційною системою та підтримувала Unicode стиль кодування символів. Та найбільш суттєвою зміною в процесі розробки цієї мови програмування стала зміна процесу розробки на повністю відкриту, прозору систему з залученням будь яких зацікавлених професіоналів-колег які мали бажання її використовувати.

В процесі написання різного роду програм та користування цією мовою програмування серед користувачів, природньо, виникали нові запити на розширення функціоналу, спрощення реалізації певних задач, покращення синтаксису та швидкодії програми. З роками використання багато змін та покращень були імплементовані у дану мову програмування, які виходили у форматі все нових та нових її версій.

Публікація **Python 3.0** у грудні 2008 року стала чи не найголовнішим релізом цієї мови програмування після багатьох років тестування та випробувань у працюючих онлайн системах. Ця версія на початках була настільки зміненою та покращеною, що не мала backward-compatibility для програм написаних на версії Python 2.0, а це означало що програми не могли легко мігрувати на цю версію, а радше мали бути переписані повністю на нову версію. Згодом багато функціональних частих були імплементовані у версіях 2.6 та 2.7, та оскільки прогрес не стоїть на місці ці зміни допомогли виключно тим програмним продуктам які зупинились у своєму розвитку на етапі функціонування і підтримки робочого стану системи, для розробки нового функціоналу і функцій очевидні переваги останніх найновіших версій як от Python 3.5 зробили рішення про розробку саме на найновішій версії очевидним.

Оскільки мова програмування Python 2.0 була в публічному просторі ще з 2000 року, 1 Січня 2020 року середовищем розробників пайтон було анонсовано зупинка підтримки цієї версії, на користь Python 3.0, та активно заохочується міграція на цю версію, адаптація та модернізація існуючих комп'ютерних систем для відповідності найновішим стандартам.

Причин для використання саме цієї мови програмування чимало, зокрема те що Python являється загальною, high level мовою програмування, яка спрощує інтеракцію програміста з комп'ютером, без вникання у бітні операції, компілятори, чи самостійне ручне керування пам'яттю яку використовує програма під час виконання на операційній системі. Також варто віддати належне місце тому факту, що пайтон являється підходящим інструментом для розробки програм на пристроях найбільш широкого спектру. Зокрема підтримуються настільні комп'ютерні системи, серверні частини, ітерація з базами даних, виконання на мобільних платформах а також розробка як утилітарних програм на Windows, Linux чи MacOS, так і для створення та підтримки web applications у мережі інтернет. Також python ддозволяє використовувати функціонал, і був спроектований таким чином,

який спрощує розробку функціоналу для дата аналітики, візуалізації та навіть для нейромереж штучного інтелекту.

Python зручно використовувати для швидкого створення прототипів програмного забезпечення, через простоту у використанні, а також підтримки TDD (test driven development), яке позволяє тестувати певні компоненти програми відразу під час її створення. Тести для окремих функцій чи класів об’єктів програмування можуть бути швидко описаними та використаними для процесу CI/CD (Continues Integration / Continues Development – система створення програмного забезпечення в якій час від внесення змін у програмний продукт до часу її публікації для використання серверною частиною або користувачами, пришвидшений до максимально можливих значень). Також ці тест функції можуть бути використаними для перевірки і гарантії того що модулі програми дотримуються поставлених до них вимог якісного характеру, базуючись на вихідному коді.

Однією з особливостей розробки у середовищі Python являється її модульна система в якій не всі функції та фреймворки є доступними відразу, а радше повинні бути імпортованими з інших модулей. Це являєтеся одним з мінусів цієї мови програмування, яка робить її швидкодію виконання процесів трохи повільнішою ніж у таких мов програмування як C++ або Java. Цей мінус можливо обійти шляхом використання власноруч розробленими модулями які користуються швидкодією спеціально написаних модулів ядра операційної системи, або специфічного пріоритету під час запуску (custom runtime).

Згідно дослідження проведеного на початку 2020 року інтернет ресурсом statista.com Python являється однією з найбільш популярних та використовуваних мов програмування у світі, зокрема він посідає 4 місце і уперше перегнав за популярністю одного з своїх конкурентів Java, рейтинг зображено на рис. 2.1:

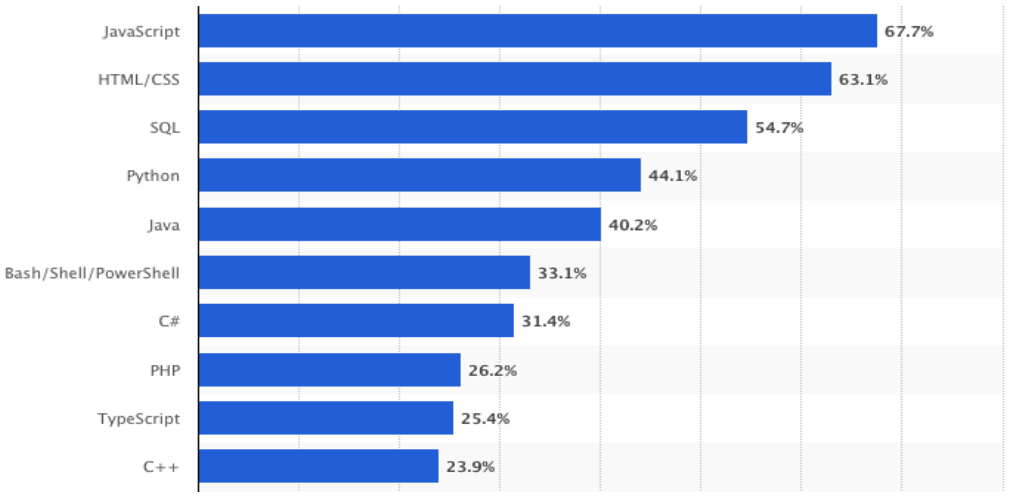

Рис. 2.1. Рейтинг популярності мов програмування

Згідно з TIOBE Index за Листопад 2020 року, основні 7 причин які виділяють Python серед інших мов програмування являються:

1) Читабельність та легкість у підтримуванні код бази.

Під час написання коду програмного забезпечення основний фокус повинен іти на якість виконуваного коду для спрощення підтримки та оновлень програмного забезпечення. Прості синтаксичні правила пайтон та концепція її структури, що максимально схожа на англійську мову, дозволяє “читати” код максимально ефективно.

2) Підтримка різних парадигм програмування.

В той час коли більшість мов програмування фокусуються і підтримують лише одну з парадигм, як от функціональне або об'єкто-орієнтоване програмування, Python підтримує обидва. В додачу до цього він також позволяє користувачеві мови самостійно вирішувати яку модель типізації даних використовувати: статичну чи динамічну. А також, як уже було згадано, вбудований менеджер пам’яті спрощує роботу інженера, беручи ці процеси на себе, дозволяючи за бажанням перекласти цю сферу повністю на вбудований функціонал.

3) Підтримка різноманітних платформ.

Python підтримується на великому різноманітті операційних систем: Linux, Unix, MacOS, Windows, Ubuntu, CentOS, Debian, Android. А отже код не

потрібно компілювати окремо під кожен тип операційної системи, радше лише сам пайтон інтерпретатор повинен бути встановленим на даній операційній системі і в подальшому файли з кодом можуть бути скопійовані на будь яку з переліку і успішно запущеними та виконаними.

4) Надійна робота стандартної бібліотеки.

Велика за обсягом та практично безпомилкова під час виконання стандартна бібліотека Python надає широкий доступ до найрізноманітніших ресурсів операційної системи та цілого ряду стандартних методів роботи з файлами, веб сервісами, та API які не потребують самостійної повторної розробки інженером.

5) Великий обсяг додаткових бібліотек і інструментів.

Оскілки мова Python являється надзвичайно поширеною серед професіоналів, великий обсяг пророблені роботи поширюється у відкрити (Open Source) доступ для використання усіх бажаючих інженерів, відповідно багато шматків функціоналу з інтеграції з різними програмними продуктами являється доступним для імпорту та миттєвого використання, в якому лише окремі, специфічні для даної програми особливості потрібно буде імплементувати самому інженеру.

6) Спрощення складності розробки програмного забезпечення.

Високорівнева архітектура і простий синтаксис python спрощує розробку функціоналу для дата аналітики, візуалізації даних, статистики, розпізнавання мови у письмовому чи аудіо форматі, та навіть для розробки нейромереж штучного інтелекту (AI – Artificial intelligence). В силу своєї популярності python породив багато спеціалістів які віддають перевагу саме цій мові програмування, а отже знайти необхідного фахівця на ринку праці простіше.

7) Адаптація принципів TDD.

TDD (test driven development), позволяє тестувати певні компоненти програми відразу під час її створення а отже ці функції можуть бути використаними для перевірки і гарантії того що модулі програми

дотримуються поставлених до них вимог якісного характеру, та швидкодії базуючись на вихідному коді.

2.2. Алгоритми та інструкції для інсталяції Python на різних операційних системах

Алгоритм дій та інсталяція для Windows:

- Слід визначити архітектуру наявної операційної системи (x86-64 або x64), та зробити відповідний вибір версії інтерпретатора мови програмування Python. На офіційному сайті www.python.org представлений повний перелік доступних версій. Базуючись на архітектурі операційної системи слід вибрати файл інсталяції x86-64 або x64, як показано на рис. 2.2.:

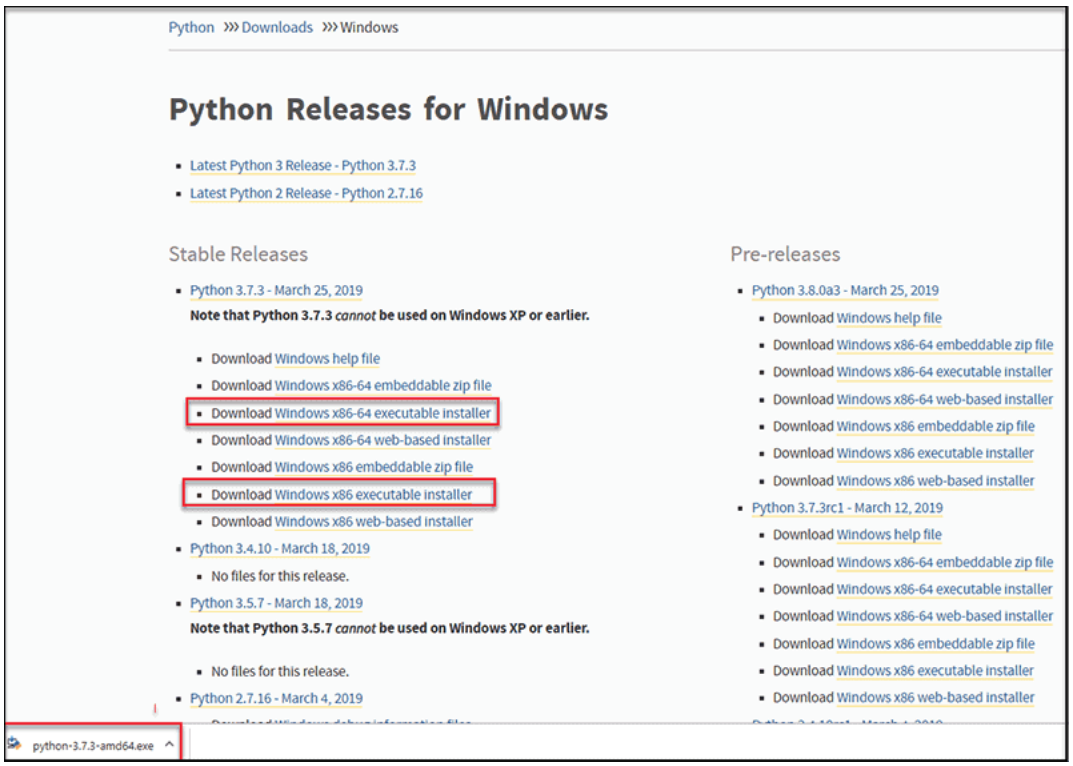

Рис. 2.2. Процес скачування файлу з вибраною версією python

- Запуск інсталяційного бінарного файла python3 з правами адміністратора зображено на рис. 2.3.:

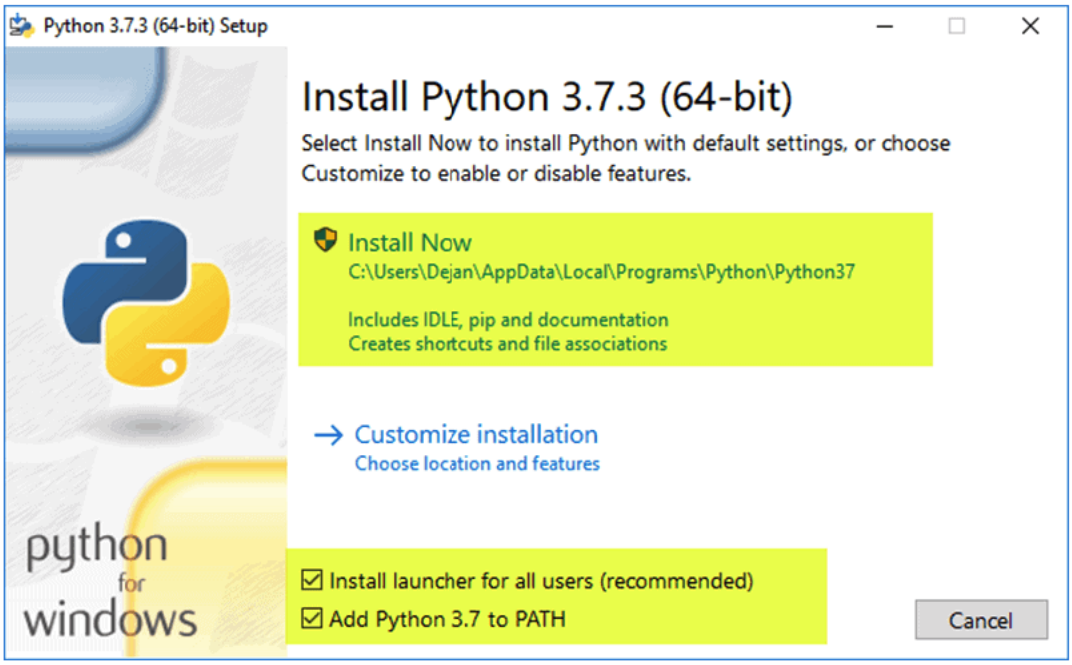

Рис. 2.3 Інсталяція бінарного файлу Python

- Необхідно перейти за шляхом:

C:\Users\Username\AppData\Local\Programs\Python\Python37 (оскільки нами було встановлено саме python3.7) та запустіть python.exe, у випадку коректної установки інтерпретатор буде запущено, що вказує про те що встановлення відбулось успішно. Робота пайтон інтерпретатора зображена на рис. 2.4.:

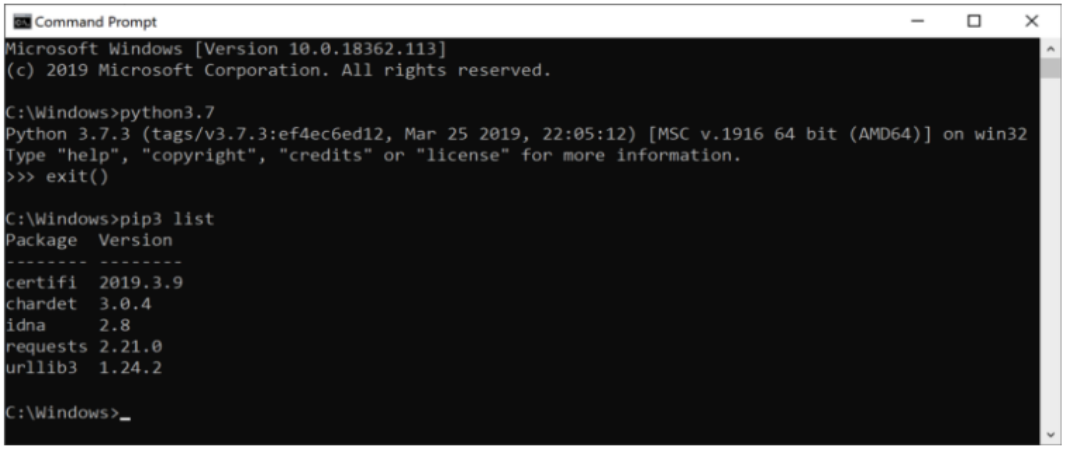

Рис. 2.4. Робота інтерпретатора Python на Windows 10

Алгоритм дій та інсталяція Pythion для MacOS:

- Операційна система MacOS включає у себе встановлений інтерпретатор python 2.x*. Та оскільки дана версія являється застарілою, а отже для встановлення Python3 необхідно відкрити термінал та виконати команду зображену на рис. 2.5.:

```
$ pyenv install 3.7.3
python-build: use openssl 1.0 from homebrew
python-build: use readline from homebrew
Downloading Python-3.7.3.tar.xz...
-> https://www.python.org/ftp/python/3.7.3/Python-3.7.3.tar.xz
Installing Python-3.7.3...
## further output not included ##
```

Рис. 2.5. Встановлення Python3 використовуючи terminal

- Успішний результат виконання цієї команди повинен мати результат встановленого Python3 на операційній системі, далі для глобального виставлення цієї версії для використання іншими програмами встановленими на даній операційній системі слід виконати команди зображену на рис. 2.6.:

```
$ pyenv global 3.7.3
# and verify it worked
$ pyenv version
3.7.3 (set by /Users/mbbroberg/.pyenv/version)
```

Рис. 2.6. Встановлення глобального середовища Python3 на MacOS

- Далі для переходу у сам інтерпретатор мови програмного забезпечення python достатньо буде виконати наступну команду у MacOS терміналі:

 # python3 –version

Результатом успішного виконання цієї команди буде вивід у консоль поточної встановленої версії: **Python3.7.3**

Алгоритм дій та інсталяція Python для Linux (Ubuntu 20):

- Першим кроком зі встановлення python3 являється обновлення індексу доступних репозиторіїв операційної системи, як показано на рис. 2.7:

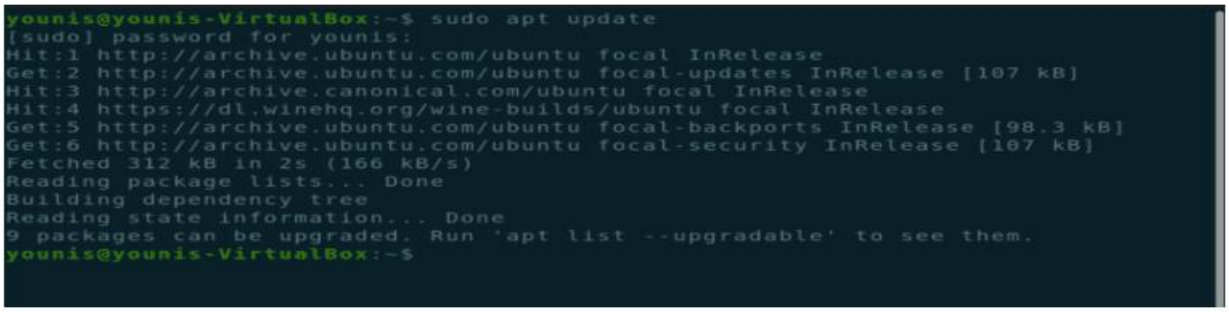

Рис. 2.7. Оновлення індексів репозиторіїв операційної системи

- Оновлення усіх існуючих пакетів операційної системи для підготовки інсталяції останньої версії Python3, як показано на рис. 2.8:

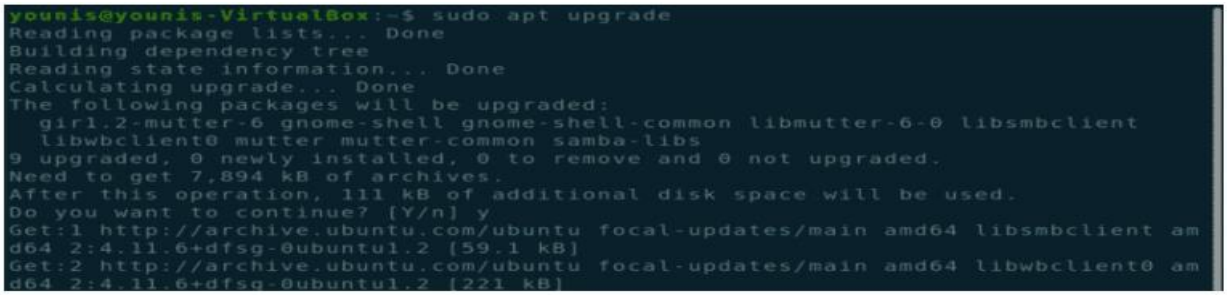

Рис. 2.8. Оновлення пакетів операційної системи

- Слід додати глобальний репозиторій для пакетів зовнішнього середовища інсталяторів програм для Ubuntu20, як показано на рис. 2.9:

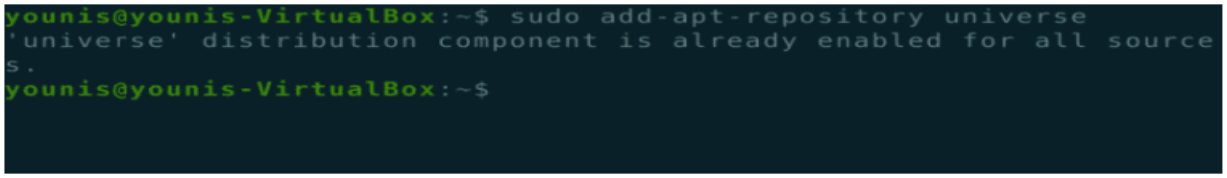

Рис. 2.9. Додавання глобального репозиторія

- Встановлюємо пакет інсталяції найновішої версії python3-pip використовуючи термінал операційної системи як зображено на рис. 2.10:

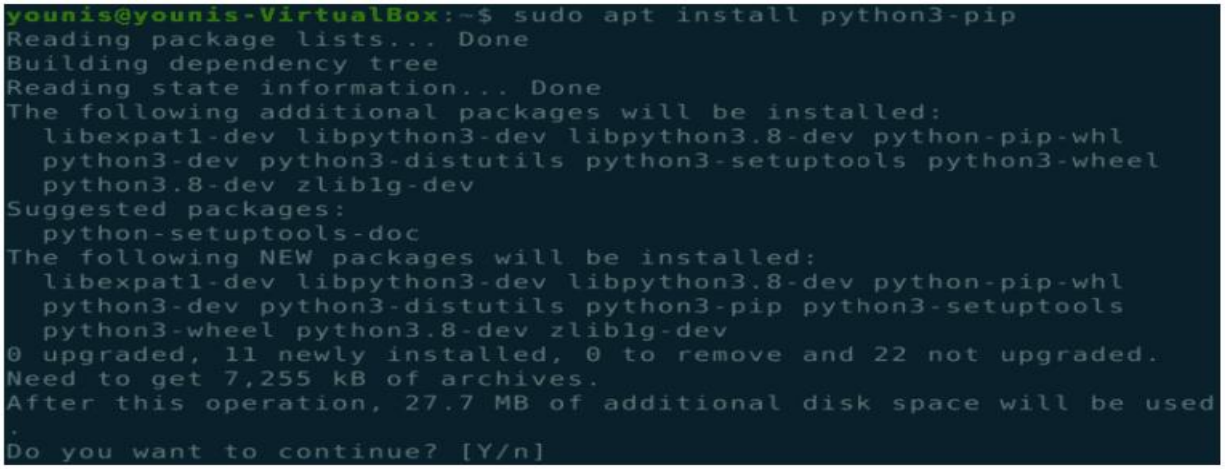

Рис. 2.10. Встановлення python3-pip у терміналі Ubuntu20

- Для перевірки правильної роботи інтерпретатора слід виконати команду у терміналі Ubuntu 20 яка зображена на рис. 2.11:

Рис. 2.11. Перевірка коректності роботи python3

2.3. Детальний аналіз синтаксису Python для відповідності вимогам реалізації шаблону сайту

Для того щоб мати змогу визначити відповідність мови програмування Python до поставлених вимог, зокрема реалізації шаблону сайту на WordPress з можливістю адміністрування, модерації та конфігурування слід розглянути синтаксис мови, та розібратись з типовими вбудованими функції найбільш поширених типів даних. У Python наявні

наступні типи даних: list, tuple, string, dictionary, set, class, function. Розглянемо їх опис, мету та призначення, на конкретних прикладах використання.

Вивід інформації, що є результатом виконання програми, можна реалізувати шляхом метода print() який приймає як аргумент необхідний для виводу об’єкт. Приклад виводу знаменитої у всьому світі фрази, яка оголосила про настання ери комп’ютерів, реалізувати шляхом python можливо так:

print(“hello world!”)

Даний елемент коду виведе у standard out строку “hello world!”, відповідно вибраного каналу стандартного аутпуту результат буде зображений у терміналі, або записаний у текстовий файл.

“hello world!” це строка. Строки, як об’єкти мають методи які виводять результат дії функції на стрічці.

Python синтаксис позволяє присвоювати значення строки змінній для наступних функціональних маніпуляцій з нею. Отже, оголосимо строкову змінну: hw = “hello world!”

Ця змінна утримує значення нашої строки і позволяє використовувати методи стандартної бібліотеки у поєднанні з нею, як показано на рис. 2.12.:

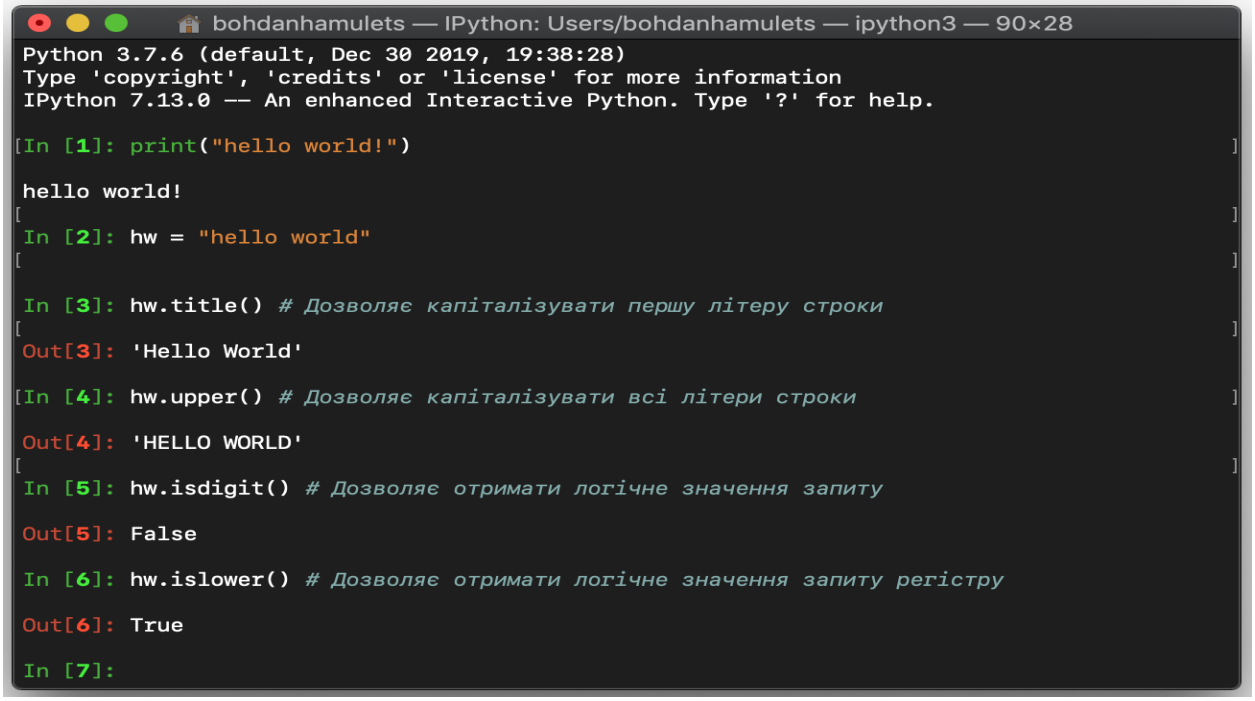

Рис. 2.12. Стандартні методи строки у Python

Значення строки в оголошенні змінній залишається тим самим (не міняється). Метод строки повертає альтернативне значення строки або Boolean logic variable – True або False. Зміна самого значення змінної потребує оголошення, як це зображено на рис. 2.13.:

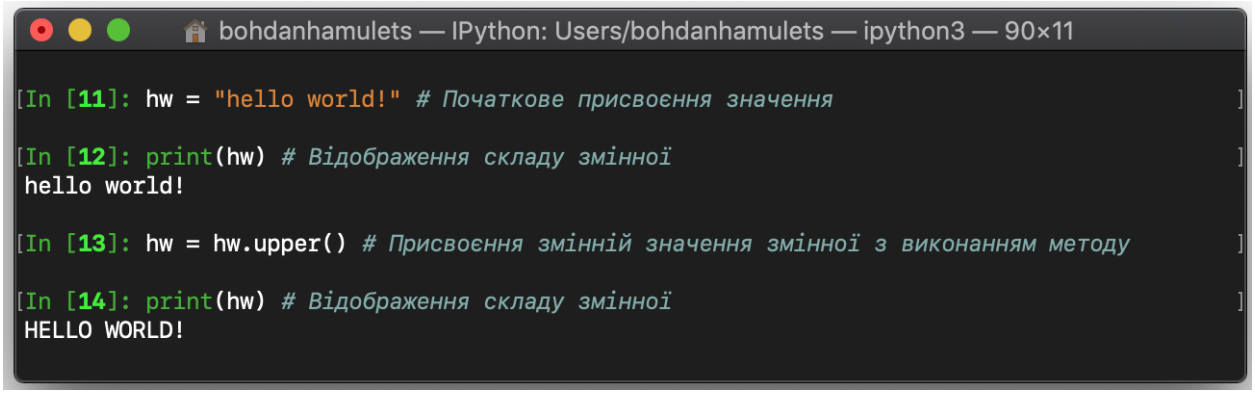

Рис. 2.13. Модифікація збереженої змінної шляхом присвоєння їй виклику методу стандартної бібліотеки

Інші види об’єктів у пайтон та їх методи зображені та описані на рис. 2.14:

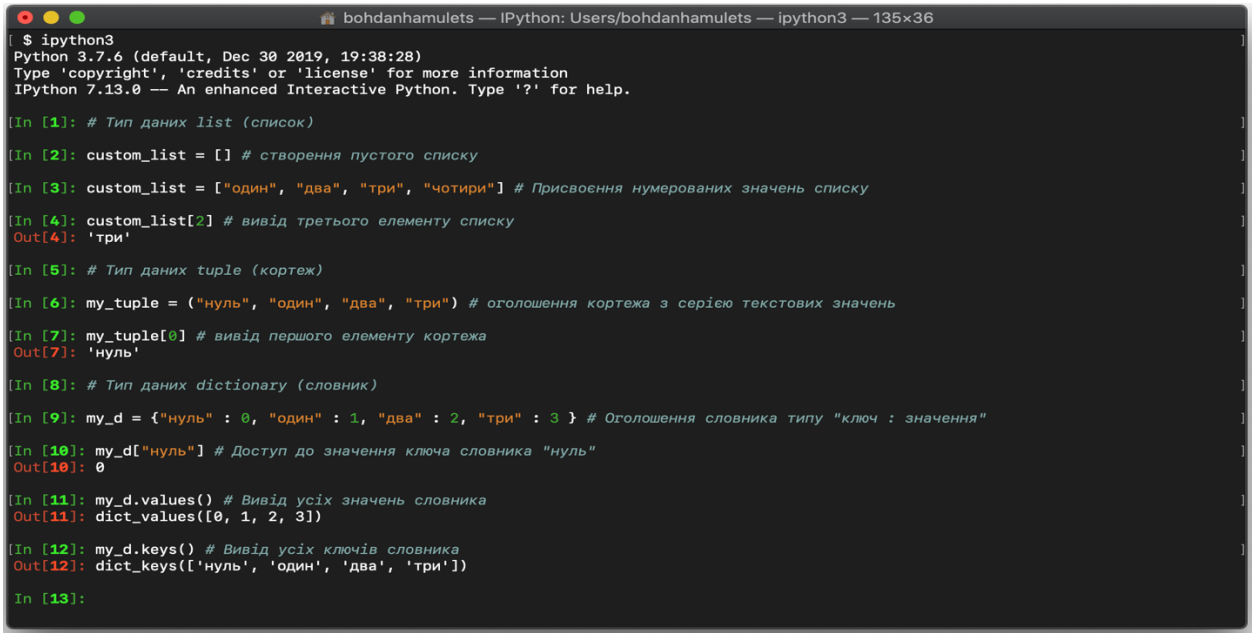

Рис. 2.14. Оголошення та робота з типами даних список, кортеж та строка у python, виклик вбудованих методів

Для описання типу даних “список” підійде аналогія з колодою карт, на кожній з яких описана певна інформація. Дані залишаються у тому ж порядку в якому вони були внесені, якщо не відбувалось жодних змін. Методи списку вертають один або більше елементів списку, додають, видаляють, модифікують поточні значення даних в списку. Індекс списку, написаний у форматі list[index] повертає значення елементу, починаючи відлік від нуля (0). Список являється iterable, тобто по його елементах можна проходитись у списку та виконувати певні функції, базуючись на логічних операторах, необхідних для досягнення поставлених цілей. Методи списку:

- Додавання до списку:

variable[n] = object

(змінює елемент n значенням object)

variable.append(object)

(доповнює список елементом object в найостаннішу позицію)

- Видалення зі списку:

variable [n] = []

(обнулює значення елементу n, але залишає поточний порядок елементів)

variable.remove(n)

(видаляє елемент n, і модифікує поточний порядок елементів)

variable.pop(n)

(видаляє елемент n і виводить його значення у stdout або присвоює його іншій змінні)

Для описання типу даних кортеж (tuple) підійде аналогія з списком дат календаря, основною відмінністю від типу даних список являється його властивість immutable – тобто значення кортежу не може бути зміненим. На базі роботи з ним стандартних методів бібліотеки python можуть бути створені нові змінні, або проведена інша маніпуляція з його даними, та сам склад кортежу залишатиметься незмінним завжди. Як і список, цей тип даних являється iterable, тобто по його елементах теж можна проходитись у циклі. В загальному цей тир даних застосовується для випадків коли від даних не очікується жодних змін, для гарантування правильної роботи програмного забезпечення та превентивного уникнення допущення помилок. Приклади: перелік назв країн, поштові індекси, перелік місяців, упорядкований список доступний функцій.

Для описання типу даних словник (dictionary) підійде аналогія з українсько-англійським словником, де у форматі “ключ”:”значення” перелічені слова рідною та зарубіжними мовами. Даний тип даних може бути використаним для: рахунку кількості повторень певних слів у тексті, відображення позицій на карті у форматі x:y, або у контексті розробки шаблону сайту на WordPress значеннями словника можуть бути поле веб елементу та його вміст. Словник позволяє ідентифікувати значення визначених ключів за їх описаним іменем (на відміну від наприклад числової позиції у типі даних список). Ключі зберігаються в словнику неупорядкованим чином, якщо не були явно посортованими. Ключі мусять мати унікальні значення та бути hashable. Спроба створити два одинакові ключі в словнику видасть помилку: SyntaxError: invalid syntax у випадку

дублювання ключа, або TypeError: unhashable type у випадку якщо ключ не може бути прохешованим.

За допомогою операторів Conditional Branching: if and else відбувається контроль виконання коду, і забезпечується робота з даними. Наприклад:

- if та else

if variable == condition:

виконає наступний код базуючись на результаті порівняння змінної з очікуваним логічним твердженням рівності значень

else:

виконає наступний код базуючись на тому що результат порівняння описаний вище повернув нерівність значень елементів

- elif

позволяє створювати додаткові гілки коду, в архітектурі:

if логічний оператор:

elif: інший логічний оператор

elif: третій логічний оператор

…

else:

виконається код за умови що жоден з трьох умов описаних вище, не були виконані.

Додатковим різновидом керівної конструкції у Python, призначеним для організації виконання багаторівневої множини інструкцій (команд) являється цикл. Ця командна інструкція виконується певну, попередньо зазначену кількість раз, або поки певна умова не буде виконана. У мові програмування Python представлені два види циклів for loop і while loop.

Розглянемо більш детальніше цикл for. Він дозволяє проходитись по блоку елементів коду задану кількість раз:

for x in range(3):

print(“Line number ” + str(x))

виведе (оскільки нумерація у пайтон починається з нульового елементу):

Line number 0

Line number 1

Line number 2

Або проходить по кожному елементі заданого типу даних, наприклад list:

```
country_list = ["Canada”, “Poland”, “Ukraine”]
   for element in country_list:
      print(element, len(element))
```

виведе по черзі кожну країну зі списку, вирахуючи кількість букв у назві:

Canada 6

Poland 6

Ukraine 7

В рамках цієї дипломної роботи, на тему розробка шаблону сайту кафедри на WordPress згідно вимог з можливістю адміністрування, цикли повинні бути використаними для веб елементів сайту, наприклад для виводу на головну сторінку кожної статті кафедри із зазначенням дати публікації. Приклади такого використання будуть описані у наступних розділах з безпосередньою розробкою шаблону. Підсумовуючи тему циклів, основним випадком їх використання являється функція виконання певного набору завдань для кожного з елементів переданого типу даних, якими можуть бути строка, словник, кортеж або інші.

Додатковою, суттєвою частини інфраструктури пайтон являється можливість підключати у файли з виконуваним кодом додаткові, самописні або елементи стандартної бібліотеки пайтон, модулі. Модулі це додаткові розділи коду які позволяють розширювати уже наявні функціональні можливості. Типово модуль заключає в собі специфічний функціонал взаємодії з певним компонентом програми, бази даних, веб серверу, мережі інтернет або API (Aplication Program Interface) іншого програмного продукту.

Мова програмування Python включає багато корисних модулів, розглянемо для прикладу використання модулів os та sys, оскільки робота з ними ведеться достатньо часто.

Модуль os надає портативний спосіб використання ресурсів операційної системи на якій відбувається запуск програми. Вищенаведена процедура встановлення мови програмування пайтон на операційні системи Windows, MacOS та Linux можлива не в останню чергу завдяки цьому модулю який позволяє читати та записувати інформацію у файли операційної системи, знаходити ресурси за вказаним шляхом файлової системи, позволяє створювати окремі папки (директиви) операційної системи та модерувати правила доступу до них.

Модуль sys надає доступ до змінних середовища в якому програма була запущена і до функцій які взаємодіють зі самим інтерпретатором. Цей модуль завжди присутній у стандартній бібліотеці та не потребує окремого встановлення на операційну систему. Взаємодія коду програми з модулями відбувається наступним чином:

import os, import sys

Це командні інструкції пайтон (import statements) які вказують інтерпретатору на необхідність підтягування цих модулів програми у виконуваний файл. Модулі мають у собі визначені сабсети функцій, з їх описом та інструкцією використання, як кількість аргументів які повинні бути передані функції, та кількість і тип змінних які ця функція модуля поверне у випадку успішного виконання. Доступ до функцій модулів відбувається через використання крапки після виклику їх назви.

Прикладами для модуля os являються:

os.name() – повертає назву операційної системи на якій відбувається виконання програми, даючи змогу змінювати поведінку наступних виконуваних інструкцій коду, враховуючи особливості різних операційних систем.

os.path() – повертає значення поточного шляху на файловій системі виконуваної програми Python, даючи змогу зберігати результати виконання або логувати помилки під час виконання у ту саму папку (директиву).

Прикладами функцій модуля sys являються:

sys.exec_info() – повертає кортеж з трьома значеннями в яких збережена інформація про перебіг та результат виконання переданої як аргумент команди. Якщо жодних помилок під час виконання не відбулось, то ця функція поверне лише одне значення None, інакше буде повернуто type, value і traceback з типом помилки, її вмістом та трейсбеком до місця в коді де помилка відбулася для можливості перевірки.

sys.exit() – ця функція завершує роботу програми Python на даній операційній системі. Використовується для випадків коли виникла небажана помилка під час виконання інструкцій коду, повертає значення SystemExit яке позволяє або миттєво завершити роботу програми, або спершу зробити очищення результатів виконання програми, а тоді завершити роботу.

Отже розглянувши базові можливості які надає мова програмування Python та її модулі можна перейти до аналізу модулів за допомогою яких завдання зі створення шаблону сайту WordPress може бути реалізованим.

2.4. Мови стилів та розмітки CSS/HTML

Cascading Style Sheet (CSS) або каскадна таблиця стилізації це – спеціальна мова, за допомогою якої стилізуються веб-аплікації, розроблені мовами розмітки веб-сторінок, яка може працювати незалежно від мови програмування яка застосовується на серверній частині.

Найчастіше CSS використовується для сторінок, які розмічені мовами Hyper Text Markup Langugage (HTML), Extensible Hyper Text Markup Langugage (XHTML) та Extensible Markup Lanuguage (XML). Специфікації CSS створено Консорціумом Всесвітньої мережі (World Wide Web Consortium або W3C), котрий їх підтримує та розвиває. Раніше розробка різних специфікацій CSS відбувалась синхронно що позволяло утримувати

версійність цієї спеціальної мови, нерідко можна зустріти CSS1, CSS2, CSS3 та навіть CSS4, остання так і не здобула статусу офіційної версії.

На сучасному етапі розвитку інформаційних наук HTML без стилів містить доволі малий набір інструментів, який не завжди надає можливість відповідати теперішнім вимогам до дизайну та вирішувати завдання сучасних веб-застосунків. Саме для вирішення цих проблем і використовується CSS, який вирішує основні завдання, які безпосередньо стосуються до стильового та логічного оформлення веб сторінок. Прикладами стилю та вирішення питань пов'язаних з дизайном служать зміни кольору, фону, шрифту, кольору, інтервалу між елементами, впорядкування вмісту веб сторінки та інші декоративні функції.

Такий поділ за зонами відповідальності у CSS/HTML має на меті покращити візуальне сприйняття та зручність користування контентом, забрати повторення елементів коду, забезпечити управління та нагляд за відображенням коду в різних станах, краще структурувати та спростити контент (вміст) тощо. CSS дозволяє пристосувати веб-сайт до най різноманітних умов відвідування (монітори, планшети, смартфони і т.д.).

До переваг використання CSS відносять:

- відображення різного дизайну для одного сайту;
- кешування стилів для мінімізації часу завантаження;
- легкість і прямолінійність в оновленні дизайну;
- створення майже досконалу верстку вебсайту;
- покращення зручності користування;
- спрощення структуризації та впорядкованості сторінок

Сучасні браузери, такі як Opera, Mozilla, Google Chrome, Safari, Internet Explorer, Microsoft Edge та Tor Browser, дозволяють швидко опрацьовувати стилі що покращує User Interface-дизайн веб-сайтів або веб застосунків. Такий же самий принцип обробки цієї мови функціонує на безлічі різних операційних систем, включаючи в себе як операційні системи для десктоп

комп’ютерів та планшетів, так і сучасних мобільних телефонів і смартфонів, отже до переваг цієї спеціальної мови розмітки також можна віднести її розповсюдженість. Використовуючи її в побудові та розробці власних веб-сайтів на WordPress забезпечить відмінну інфраструктурну поширеність а отже гарантуватиме те що користувачі матимуть коректний доступ з правильним відображенням елементів.

CSS-фреймворки – являються потужними інструменти, які допомагають удосконалити процес розробки та проектування.

Для прикладу, один з таких, Bootstrap, (спочатку називався Twitter Blueprint), був створений lдвома видатними експертами в галузі Mark Otto і Jacob Thronton а його випуск відбувся у серпні 2011 року. Це зовнішній інтерфейс з відкритим вихідним кодом, який складався з HTML, CSS і JavaScript. Він має модульну структуру та використовує препроцесор SassCSS. Також варто додати те, що він містить у собі не лише CSS, але й JavaScript-фреймворк. У Bootstrap написані готові скрипти та стилі, для їх використання розробнику достатньо лише створити необхідні стильні атрибути та класи HTML-елементів.

Bootstrap фреймворк підтримує останні стабільні версії усіх основних гравців на ринку браузерів і платформ.

Hyper Text Markup Langugage (HTML) або гіпертекстова мова розмітки) - це синтаксис тегів, який застосовується для розробки, формування та візуалізації веб-сайтів та мобільних застосунків. для розміщеного у них контенту. Перша версія була написана Тімом Бернс-Лі у 1993 році. З цього часу було випущено багато різних версій цієї мови, та на сьогоднішній час найбільшою популярністю користується версія HTML 4.01 яка набула статусу офіційного стандарту для вебу у 1999 році. Тому на сьогодні у світу абсолютна більшість веб сторінок та застосунків використовує саме версію HTML 4.01. Активно ведеться модифікація та покращення існуючого стандарту HTML до версії 5.0 яка була

опублікованою у 2011 році, та станом на сьогодні її поширення ще не набуло такого суттєвого прийняття.

Робота HTML полягає у логічному розділі інформації, як таблиця, меню, заголовок та інші частини, в залежності від необхідності її донесення до кінцевого споживача.

HTML, за допомогою використання таблиць стилізації CSS та програмним функціям, являється головним засобом з побудови веб-додатків.

Ця комбінація позволяє:

- отримувати інформацію із World Wide Web через різноманітні гіперпосилання (URL);
- вбудовувати звук, відео статичні зображення, та інші об'єкти безпосередньо у текст;
- створити формалізований структурний компонент, позначенням особливостей контенту або тексту (меню, заголовки, таблиці)
- додавати елементи JavaScript для зміни наповнення сторінки з статичного на динамічний;
- організовувати вміст сторінки шляхом надання headings та subheadings на додачу до кожної секції веб сторінки.
- створювати інтерактивні форми.

2.5. Висновки до розділу

У цьому розділі проведено вибір і детальний опис усіх складових елементів для успішного практичного виконання дипломної роботи на тему "Розробка шаблону сайту кафедри на WordPress згідно вимог з можливістю адміністрування, модерації, конфігурування", а саме, було розглянуто:

- мова програмування, вибрана для реалізації специфічних функцій з керування спеціалізованими можливостями наповнення та редагування вмісту веб сайту, розміщеного на WordPress, шляхом взаємодії з вбудованим розширенням XML-RPC інтерфейсу – Python. Аналітичний опис синтаксису,

характеристика застосувань та підтримувані можливості доводять правильність цього вибору для виконання поставленої задачі.

- слідуючи інструкціям з попереднього розділу було встановлено пакет програмного забезпечення WordPress на локальний хост, що забезпечило наступний логічний крок - детальну інструкцію встановлення, та базовий варіант використання Python у цьому розділі.
- розглянуто основи взаємодії Python XML-RPC модуля для динамічної інтеграції з відповідним компонентом на WordPress, для забезпечення базового функціоналу з: створення постів, сторінок, завантаження медіа контенту, завантаження та модерації thumbnails, автентифікації користувачів, та ін.
- проаналізований базовий функціонал стилів та розмітки CSS/HTML, та способи взаємодії з ними для розробки шаблону сайту кафедри.

Отже можна стверджувати про наявність усіх необхідних елементів системи для практичної розробки шаблону сайту кафедри у наступному розділі.

РОЗДІЛ 3
РОЗРОБКА ШАБЛОНУ САЙТУ КАФЕДРИ ЗГІДНО ДО ФУНКЦІОНАЛЬНИХ ВИМОГ

3.1 Огляд Python WordPress XML-RPC специфікації

Мова програмування Python реалізує взаємодію з WordPress через спеціальну бібліотеку wordpress-xmlrcp яка, як випливає з назви, використовує у своїй роботі функціонал XML-RPC API.

Дана бібліотека, яка може бути підключена до мови програмування Python була спроектована та розроблена для спрощення інтеграції цих двох елементів. Надаючи змогу користуватись самописними, власноруч створеними XML-RPC API методами, ця бібліотека робить доступними систему plugins (доповнень).

Дана специфікація надає групу класів та об'єктів які працюють над стандартними методами типів даних WordPress як обгортка, забезпечуючи функціонування різних, необхідних під час розробки об'єктів (наприклад, Blog, Post, User). Надані методи повертають описані значення у випадку успішного виконання програми.

Для інсталяції цього модуля достатньо упевнитись в тому що у використанні знаходиться найновіша версія WordPress у якій увімкнута підтримка XML-RPC, якщо вона не була увімкнута по налаштуваннях за замовчуванням, то увімкнути її можна за наступним шляхом Settings -> Writing -> Remote Publishing і відмітити чек бокс для XML-RPC.

Далі у робочому терміналі буде достатнім увести команду:

pip3 install python-wordpress-xmlrpc

Результатом її виконання буде завантаження необхідних пакетів з репозиторія та їх установки на операційну систему.

Для швидкого старту використання бібліотеки необхідно створити об'єкт класу Client з URL необхідного WordPress XML-RPC ресурсу, використовуючи захищене з'єднання, передаючи для автентифікації ім'я користувача та пароль. Тоді необхідно передати XmlrcpMethod об'єкт у запит метода для виконання команди на віддаленому ресурсі та отриманні результату, рис. 3.1.:

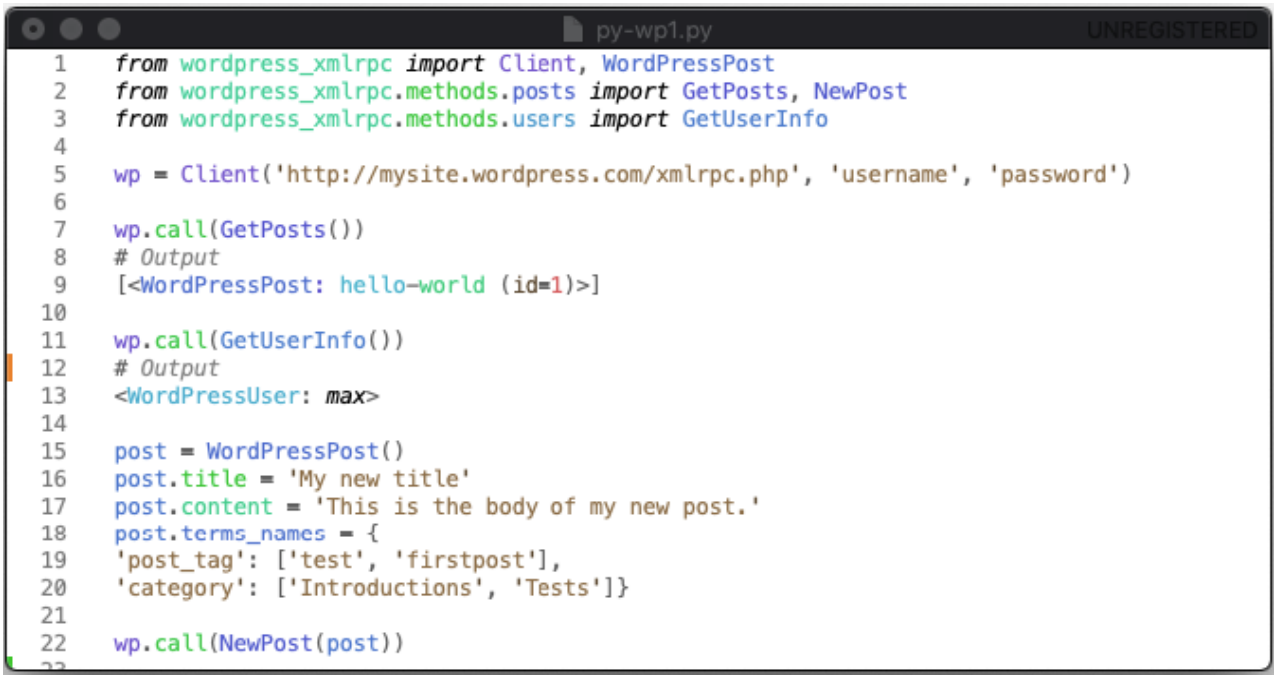

Рис. 3.1. Створення об’єкту класу Client

Варто зауважити що характеристики об’єкта WordPress можуть бути доступними напряму, без оголошення атрибутів визначених у вихідному коді. Коли WordPress об’єкт використовується як метод з параметром його struct параметр автоматично видається для використання XML-RPC, але якщо використати цей об’єкт у списку або іншому типі даних, як параметр, слід впевнитись у дотриманні синтаксису та використати obj.struct бо інакше WordPress не отримає дані у тому форматі, в якому він їх очікував.

Для взаємодії з не стандартними методами XML-RPC методами (такими що їх роблять доступними розширення plugins) метод повинен бути продовженим до wordpress_xmlrpc.XmlrpcMethod або один з його під класів (AnonymousMethod або AuthenticatedMethod).

Прикладом нестандартного, кастомізованого під особисті потреби методу може бути наприклад такий, рис. 3.2.:

Рис. 3.2. Кастомізований метод

Для роботи з окремими публікаціями (Posts) у бібліотеці python-wordpress-xmlrpc було реалізовано всі види типів постів. У WordPress “під капотом” всі пости базуються на єдиній post таблиці бази даних, уся функціональність якого відкрита через post методі у XML-RPC API. Для послідовності такий самий підхід було використано і у бібліотеці python-wordpress-xmlrpc.

Спочатку розглянемо типові пости. По налаштуваннях за замовчуванням усі пости надсилаються як чернетки, які не будуть опубліковані на веб-сайті відразу, якщо є потреба опублікувати їх відразу то слід використовувати атрибут post.post_status=’publish’.

Отже спершу давайте отримаємо існуючі WordPress пости, рис. 3.3.:

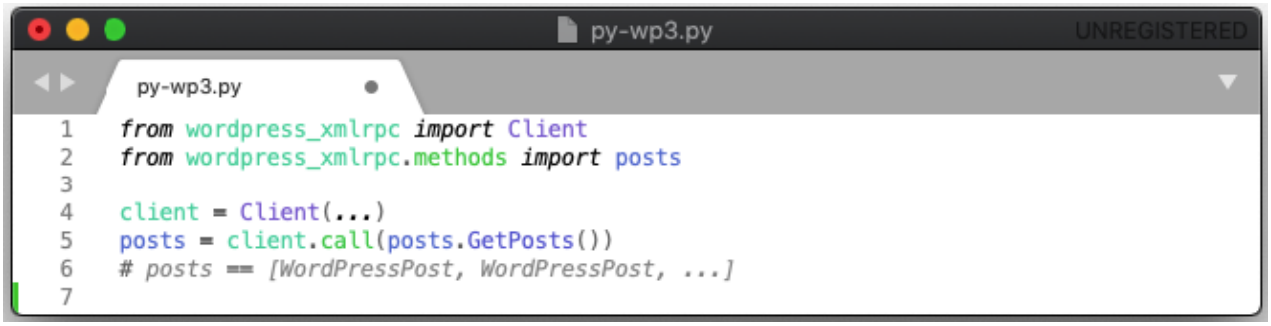

Рис. 3.3. Отримання опублікованих постів

Процес створення нового посту та його публікацію на контент менеджмент систему WordPress зображено на рис. 3.4.:

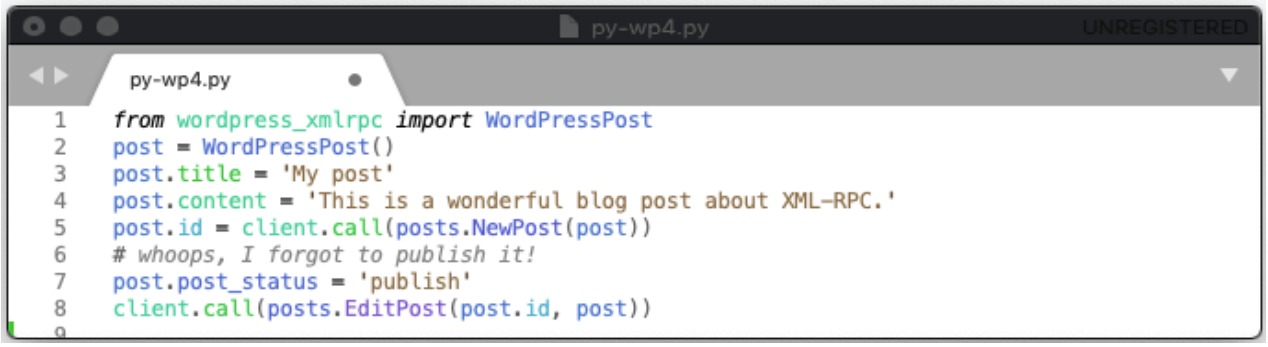

Рис. 3.4. Створення та публікація нового посту

Наступними ідуть операції зі сторінками. По налаштуваннях за замовчуванням WordPress підтримує типу посту ‘page’, для статичних, не блог сторінок на веб-сайті, які це реалізувати для сторінок зображено на рис. 3.5.:

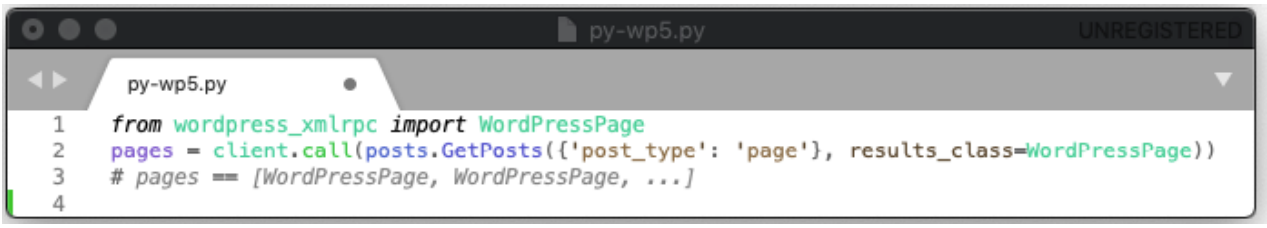

Рис. 3.5. Вибірка для операцій з сторінками

Варто занотувати дві важливі різниці:

1) Параметр filter опції post_type використовується для лімітації запиту до об’єкта з бажаним post_type.

2) Конструктор який був переданий як results_class, іменний аргумент, вказує який клас використовувати для інтерпретації запиту та повернення значень успішного виконання.

Приклад створення та модифікації сторінки зображено на рис. 3.6.:

Рис. 3.6. Створення та модифікація нової сторінки

Створення кастомізованих видів Post Types. Поки сторінки прикладів використовують свої власні results_class, це являється унікальною ситуацією через те що сторінки у WordPress мають спеціальні поля прямо у post tables. Більшість кастомізованих post types натомість використовують інші,

спеціально створені поля для зберігання додаткової інформації, а ці поля в свою чергу доступні через WordPressPost, приклад на рис. 3.7.:

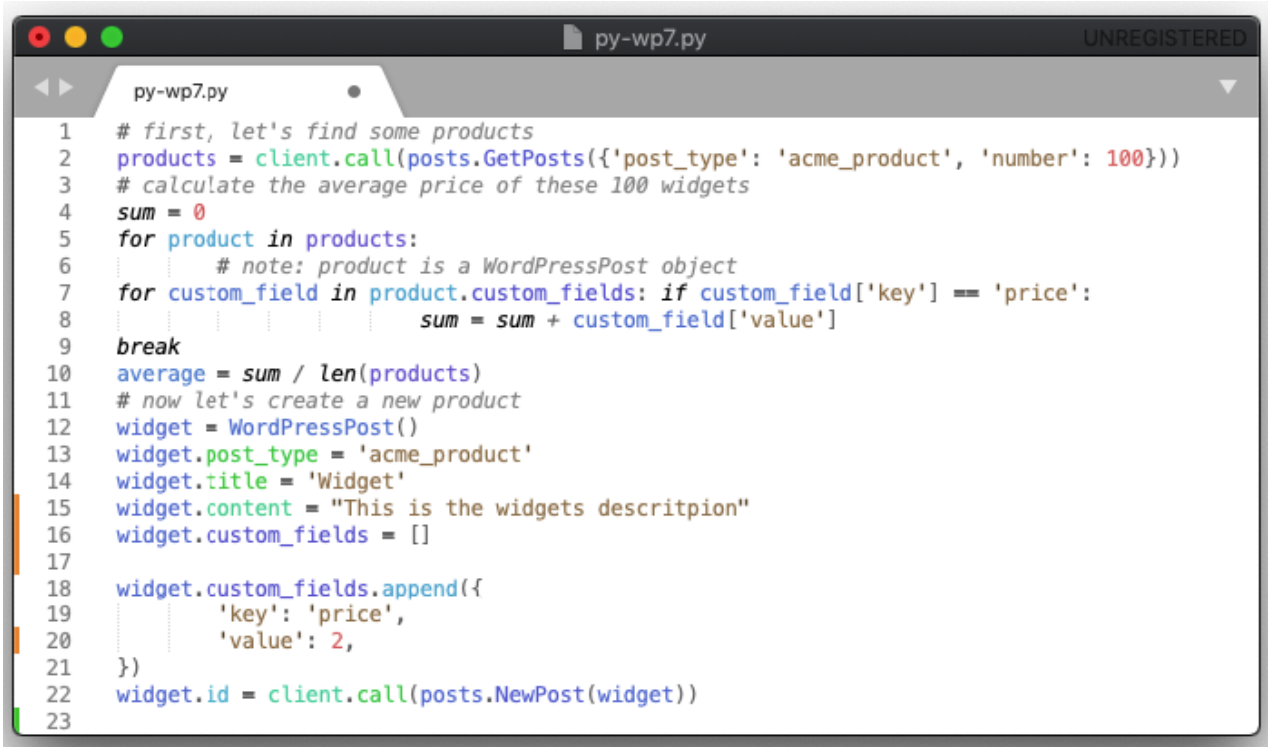

Рис. 3.7. Доступ до спеціальних полів через WordPressPost

Для реалізації більш складних запитів до WordPress у налаштуваннях за замовчуванням wordpress_xml_rpc.methods.posts.GetPost повертає як результат виконання 10 постів, посортованих за принципом в якому найостанніший пост займає першу позицію (базуючись на полі publish date). Використовуючи параметр filter цю поведінку можна змінити і створити запит на повернення постів за іншими принципами. Якщо до прикладу необхідно пройтись у циклі по усі постах WordPress блогу, зручним для серверної частини набором сторінок буде результат з використанням опцій number та offset, зображений на рис. 3.8.:

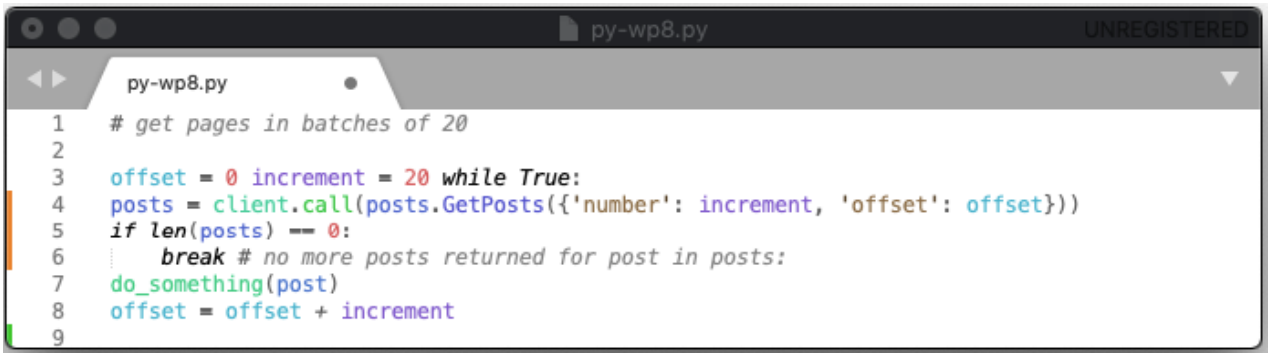

Рис. 3.8. Використання опцій number та offset

Упорядкування постів, за умови якщо не потрібно сортувати за post_date, може бути використано orderby та order опції для зміни поведінки, як зображено на рис. 3.9.:

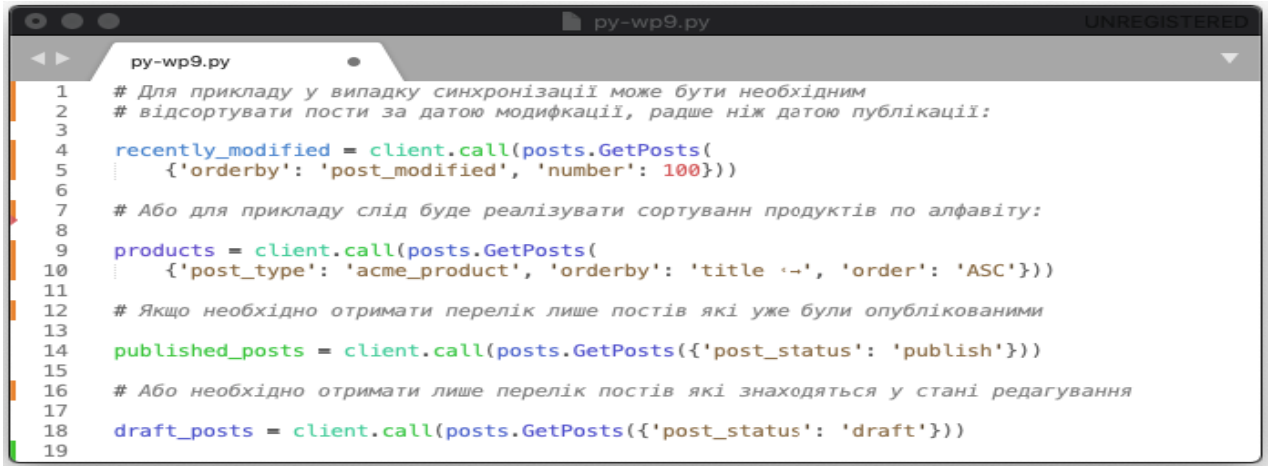

Рис. 3.9. Використання orderby та order опції

Робота з Taxonomies у WordPress здійснюється за принципом захищених даних. По налаштуваннях за замовчування WordPress налічує дві основні taxonomies: категорії (category) та теги (pos_tags). Встановлені плагіни та кастомізовані теми можуть містити додаткові taxonomies. Для отримання типу даних список (list) з taxonomies WordPress слід використовувати wordpress_xmlrpc.methods.taxonomies.GetTaxonomies, рис. 3.10.:

```python
from wordpress_xmlrpc import Client
from wordpress_xmlrpc.methods import taxonomies
client = Client(...)
taxes = client.call(taxonomies.GetTaxonomies())
# taxes == [WordPressTaxonomy, WordPressTaxonomy, ...]
```

Рис. 3.10. Отримання типу даних список з taxonomies WordPress

Одинарна taxonomies може бути отримана через атрибут її імені: *category_tax = client.call(taxonomies.GetTaxonomy('category')).*

Також у taxonomies індивідуальними записами являються terms, для створення пошукового запиту по усіх блогах цієї категорії слід виконати: *categories = client.call(taxonomies.GetTerms('category')).*

Операції зі створення нового тегу зображено на рис. 3.11.:

```python
from wordpress_xmlrpc import WordPressTerm
tag = WordPressTerm()
tag.taxonomy = 'post_tag'
tag.name = 'My New Tag'
tag.id = client.call(taxonomies.NewTerm(tag))
```

Рис. 3.11. Створення нового тегу

Приклад створення підпорядкованої категорії у WordPress зображено на рис. 3.12.:

```python
parent_cat = client.call(taxonomies.GetTerm('category', 3))
child_cat = WordPressTerm()
child_cat.taxonomy = 'category'
child_cat.parent = parent_cat.id
child_cat.name = 'My Child Category'
child_cat.id = client.call(taxonomies.NewTerm(child_cat))
```

Рис. 3.12. Створення підпорядкованої категорії

Індивідуально від terms користі небагато, тому вони мають бути записаними до posts. Якщо об’єкт WordPresTerm уже був створеним, слід використовувати terms атрибут об’єкта WordPressPos, рис. 3.13.:

Рис. 3.13. Запис terms до posts

Якщо необхідно додати категорію до вже опублікованого посту необхідно виконати код зображений на рис. 3.14.:

Рис. 3.14. Модифікації опублікованого посту з додаванням категорії

Та у випадку якщо terms ще не були отриманими і необхідно створити нові terms то слід використовувати атрибут WordPressPost, рис. 3.15.:

Рис. 3.15. Створення нових terms

Варто звернути увагу що terms_names являється словником з taxonomy names у форматі ключів та списку строкових значень. WordPress шукатиме існуючі terms з цими значеннями або створить нові, у випадку якщо задані не являтимуться присутніми. З ієрархічними taxonomies такого типу як category слід поводитись особливо обережно, оскільки потенційно вказане дублікат ім'я (різні terms можуть мати різні імена, якщо вони наслідують атрибути різних батьківських об'єктів), WordPress зауваживши таке співпадіння може видати помилку та вимагати використання tearms натомість, з коректним WordPressTerm.

Для знаходження довільного номеру найбільш використовуваних тегів. може бути використаним наступний пошуковий запит у циклі for, описаний на рис. 3.16.:

Рис. 3.16. Цикл для отримання номеру найбільш використовуваних тегів

Для знаходження term names незалежно від регістру написання їх назви, слід використовувати метод search з опцією для вибірки filter, як показано на рис. 3.17.:

Рис. 3.17. Метод пошуку для тагів шляхом використання list comprehension

3.2. Прикладе використання Python модуля WordPress XML-RPC для розробки шаблону сайту

Отже, після того як у попередніх розділах ми розглянули поетапну взаємодію модуля wordpress_xmlrpc з окремими, базовими елементами функціоналу WordPress, час переходити до комплексної, всеосяжної роботи з розробки шаблону сайту кафедри.

Для роботи з медіа файлами та їх завантаження на сайт використовується метод wordpress_xmlrpc.methods.media.UploadFile, він надає можливість розміщувати зображення або інші медіа на WordPress blog, як це показано на рис. 3.18.:

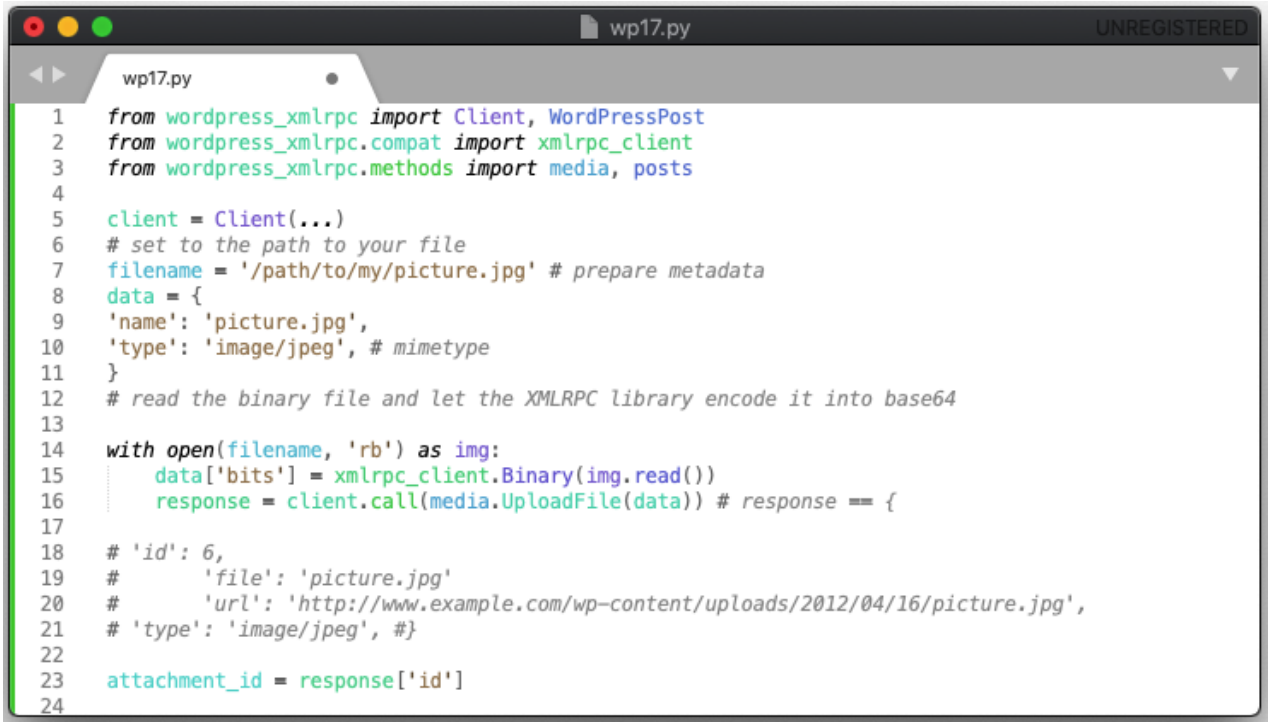

Рис 3.18. Завантаження медіа файлу

Цей новостворений файл у подальшому можна використати в якості thumbnail посту, як показано на рис. 3.19.:

Рис. 3.19. Створення thumbnail посту

Зверніть увагу, якщо mimetype не був відомим заздалегідь, то знайти його значення можна за допомогою вбудованого методу wordpress_xmlrpc.methods.media.GetMediaItem().

Наявних методів XML-RPC, які доступні по налаштуванням за замовчування у WordPress, може не завжди вистарчати для створення усього необхідного спеціалізованого функціоналу, для такого випадку підтримується створення нових, спеціалізованих методів XML-RPC, які розробник програмного забезпечення можу створити сам. Усі деталі по роботі з цим функціоналом описані у довідці до WordPress Codex, типовий приклад зі створення та використання такого методу зображено на рис. 3.20:

Рис. 3.20. Створення та виклик спеціалізованого методу XML-RPC

Також для забезпечення кращої безпеки веб сайту рекомендовано використовувати власно-написаний, спеціалізований, метод автентифікації для адміністрування WordPress, оскільки даний продукт користується

великою популярністю і присутній на значній частині вебсайтів та блогів в інтернеті, коли дослідники або зловмисники знаходять спосіб обійти стандартний метод автентифікації то ця інформація публікується у відкриті джерела інтернету з описом проблеми та списком необхідних дій для зменшення або усунення наявної загрози у безпеці. З цього випливають два висновки: слід завжди оновлювати програмне забезпечення до найновіших версій, в яких неполадки та недоліки в безпеці уже були виправленими, а також захищати свій вебсайт від автоматичних скриптів зловмисників, які постійно проходять масивом інтернет веб сайтів у пошуках вразливостей. Саме з цією метою розробимо спеціалізований метод автентифікації, як показано на рис. 3.21.:

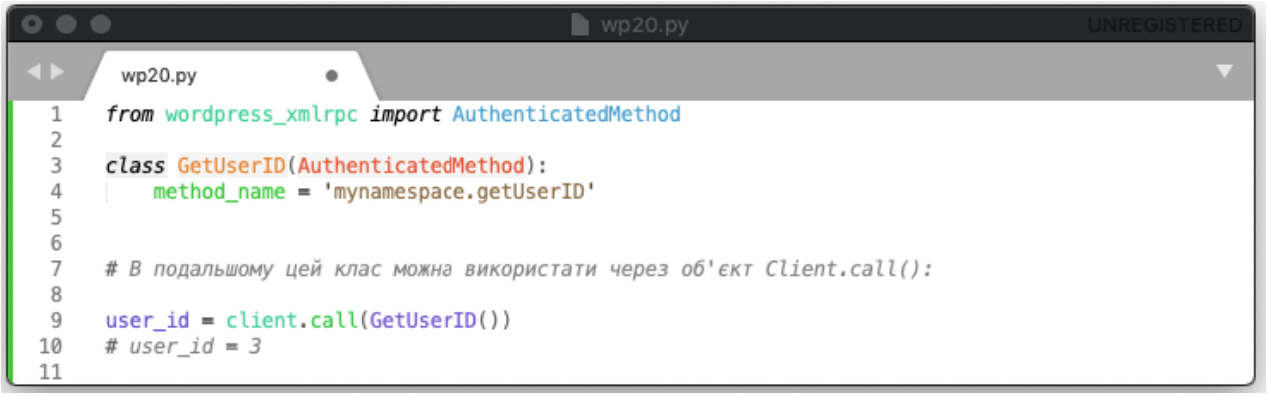

Рис. 3.21. спеціалізований метод автентифікації

Варто звернути увагу на те що в даному прикладі не потрібно передавати аргументів blog_id, username чи password, оскільки ці дані будуть автоматично підтягнуті через передання класу AuthenticateMethod. Отже, спеціалізовані методи автентифікації вимагають лише передання method_agrs або, опціонально, optional_args.

Отже, наявної інформації достатньо для розробки базового класу Client який служитиме вхідними воротами (gateway) для усіх інтерацій з WordPtess XML-RPC інтрефейсу. Після ініціалізації блогу з заданими параметрами посилання (URL) та інформацією для автентифікації (логін та

пароль, указані у другому розділі даної дипломної роботи), цей об’єкт готовий до виконання завдань WordPress через Client.call() метод:

class Client(url, username, password[, blog_id, transport])

Атрибути:

- ***url*** – посилання на сайт кафедри (наприклад, https://dl.tntu.edu.ua/)
- ***username*** – Ім’я існуючого користувача блогу WordPress
- ***password*** – Його пароль
- ***blog_id*** – ідентифікатор блогу (оскільки WordPress може підтримувати відразу декілька блогів одночасно, цей параметр можна ігнорувати для конкретно нашої задачі)
- ***transport*** – Спеціально створений метод XML-RPC (наприклад створений нами раніше метод автентифікації на рис. 3.4)

Call (method)

- Параметри методу method – один з бібліотеки ***wordpress_xmlrpc.XmlrpcMethod.***

Наявний перелік класів у бібліотеці WordPress XML_RPC API для роботи з вищенаведеним методом Client:

wordpress_xmlrpc.XmlrpcMethod – базовий клас для XML-RPC методів. Наслідувані класи можуть переписувати існуючі методи та атрибути, за бажанням, для досягнення спеціалізованих цілей.

Атрибути:

- ***method_name*****:** XML-RPC назва методу (напр, ‘wp.getUserInfo’)
- ***method_args*****:** Метод-специфічний кортеж для обов’язкових параметрів.

- ***optional_args*:** Кортеж з опціональними (необов’язковими).
- ***results_class*:** Python клас який переведе специфічну XML-RPC

відповідь у об’єкт для зручної обробки з типом даних словник.

default_args(client)

Вибудовує сет неспецифічних для методу аргументів

get_args(client)

Витягує фінальний сет XML-RPC методів аргумента, базуючись на будь-яких налаштуваннях методу за замовчуванням, або їх задекларованого порядку.

process_result(raw_result)

Виконує дії над сирим (необробленим) результатом відповіді XML-RPC. Якщо results_class є визначеним то відповідь буде серіалізована у одному або декількох об’єктів інстансів цього класу.

class wordpress_xmlrpc.AnonymousMethod: метод для роботи з даними які не потребують попередньої автентифікації (прикладом у нашому випадку буде функціонал типу «задати запитання» або «залишити коментар»)

class wordpress_xmlrpc.AuthenticatedMethod: метод для роботи з даними які вимагають попередньої автентифікації (прикладом у нашому випадку буде функціонал типу «завантажити кваліфікаційну роботу» або «пройти тестування по курсу»).

3.3. Робота з обов’язковими елементами шаблону сайту на WordPress

У попередніх розділах було налаштоване середовище для розробки Python з використанням WordPress XML-RPC модуля, HTML/CSS, та піднято вебсервер на якому був встановлений базовий функціонал пакету програмного забезпечення, системи менеджменту контенту WordPress. Для

реалізації розробки шаблону усі необхідні базові елементи повинні бути присутні у темі сайту WordPress, наведені у табл. 3.1.

Таблиця 3.1

Назва Файлу	Короткий опис та застосування
style.css	Основний файл для стилів CSS.
rtl.css	Цей файл стилів буде встановлений автоматично у випадку якщо контент розміщений “right-to-left”.
index.php	Основний публічний документ для входу на сайт.
comments.php	Шаблон для наслідування стилів коментарів.
front-page.php	Шаблон наслідування для сторінки фасаду.
home.php	Домашня сторінка, (за замовчуванням - front page).
single.php	Шаблон одиночного посту, використовується для усіх постів, крім тих де використовуватиметься index.php
single-{post-type}.php	Спеціально налаштовані одинарні пости.
page.php	Шаблон наслідування для одинарної сторінки.
category.php	Шаблон наслідування для категорії.
tag.php	Шаблон наслідування для базових тегів.
taxonomy.php	Шаблон наслідування для терміну.
author.php	Інформація про автора теми шаблону.
date.php	Шаблон дати у визначеному форматі.
archive.php	Шаблон архівів. Використовується у випадку коли категорія, автор, або певна дата була затребувана клієнтом.
search.php	Шаблон пошуку інформацій на веб-сайті.
attachment.php	Шаблон вкладень, використовується для усіх вкладень.
image.php	Шаблон для налаштувань зображень.
404.php	Шаблон сторінки з помилкою HTTP 404.

Кожен з розглянутих вище файлів, як описано, має свою зону відповідальності і визначає спосіб, зовнішній вигляд та стиль кожного з елементів, відповідно. Отже для конкретних точкових змін у веб-сайті існує свій файл, в якому, відповідно до його специфіки зміни вносяться у форматі синтаксису HTML/CSS, PHP, JavaScript, або Python і після внесення цих змін та збереження файлу зміни передаються усім наслідуваним об’єктам.

3.4 Адміністрування шаблону сайту кафедри на WordPress

Можливість адміністрування шаблону сайту кафедри на WordPress здійснюється за допомогою меню Налаштування> Загальні, що зображено на рис. 3.22.:

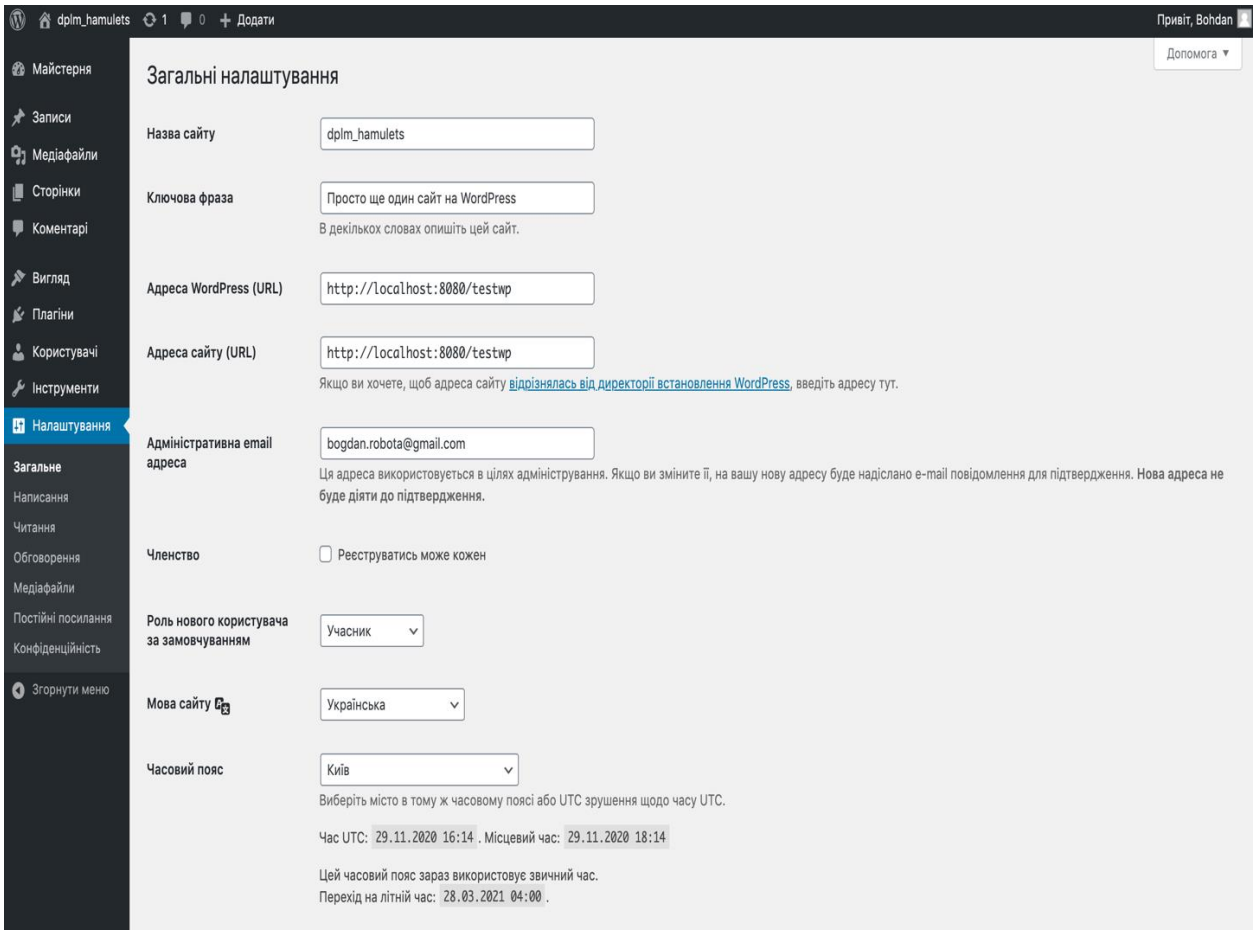

Рис. 3.22. Загальні налаштування Адміністрування WordPress

Отже, можливості адміністрування включають:

- Вибір назви сайту (позволяє налаштувати назву для відображення у веб браузері);
- Ключові фрази для індексації пошуковиками;
- Адреса сайту для користувачів (публічна адреса, розміщена у мережі інтернет) ;
- Адреса сайту для адміністраторів (як правило використовується приватна адреса, розміщена у локальній мережі установи, або на локальному комп'ютері адміністратора) ;
- Адміністративна поштова скринька (скринька на яку надходитиме електронна пошта з деталями про публікацію нових версій та іншої інформації) ;
- Налаштування можливості реєстрацій (відкрита для кожного, або лише за членством, тобто запрошенням) ;
- Роль користувачів;
- Мова сайту;
- Часовий пояс сайту (для відображення дати публікацій матеріалів).

За допомогою даних інструментів, які присутні у пакеті програмного забезпечення WordPress, можна детально налаштувати саме ті права доступу та особливості, які являтимуться необхідними для виконання поставлених цілей. У нашому випадку - розробки шаблону сайту кафедри на WordPress - ці налаштування повинні бути влаштовані таким чином, щоб забезпечити максимальну безпеку від потенційних зловмисників, та разом з тим зробити сайт дружнім для використання абітурієнтів, студентів та професорського складу кафедри, отже слід дотримуватись балансу в налаштуваннях, який буде сприятливим для усіх зазначених випадків.

3.5 Модерування шаблону сайту кафедри на WordPress

Для безперебійного функціонування сайту необхідна модерація. Вона фокусується на контролі інформації, яка розміщується користувачами. У роботу входить періодичне очищення від спаму та обмеження прав доступу відвідувачів, що порушують правила.

Модерування шаблону сайту кафедри складається з двох частин:

1) Модерування дозволів на Написання (публікацію) на веб-сайті, зображено на рис. 3.23.:

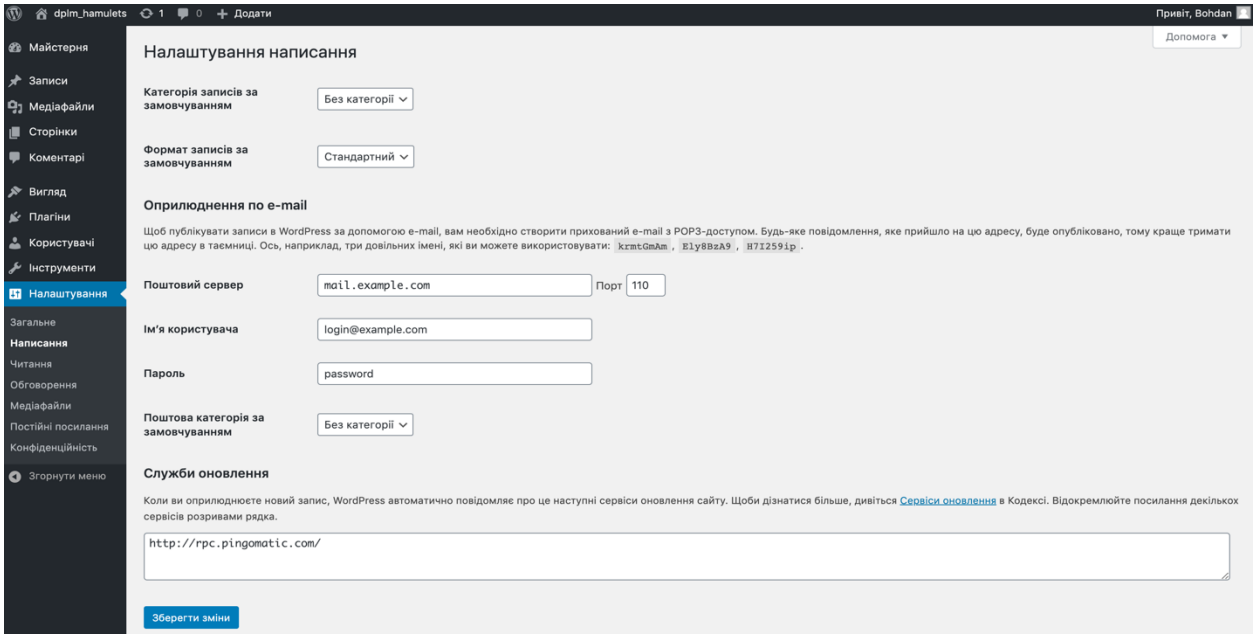

Рис. 3.23. Модерування Написання WordPress

Можливості написання включають:

- Вибір категорії записів за замовчуванням;
- Вибір формату записів за замовчуванням;
- Оприлюднення по поштовій адресі (щоб публікувати дописи шляхом надсилання емейлу необхідна попередня конфігурація POP3 поштового серверу) ;
- Налаштування поштового серверу;
- Ім’я користувача для прав написання;
- Пароль користувача для прав написання;

- Поштова категорія за замовчуванням;
- Служба оновлень (WordPress може автоматично повідомляти вибрані сервіси про оновлення на веб-сайті).

2) Модерування дозволів для Читання, зображено на рис. 3.24.:

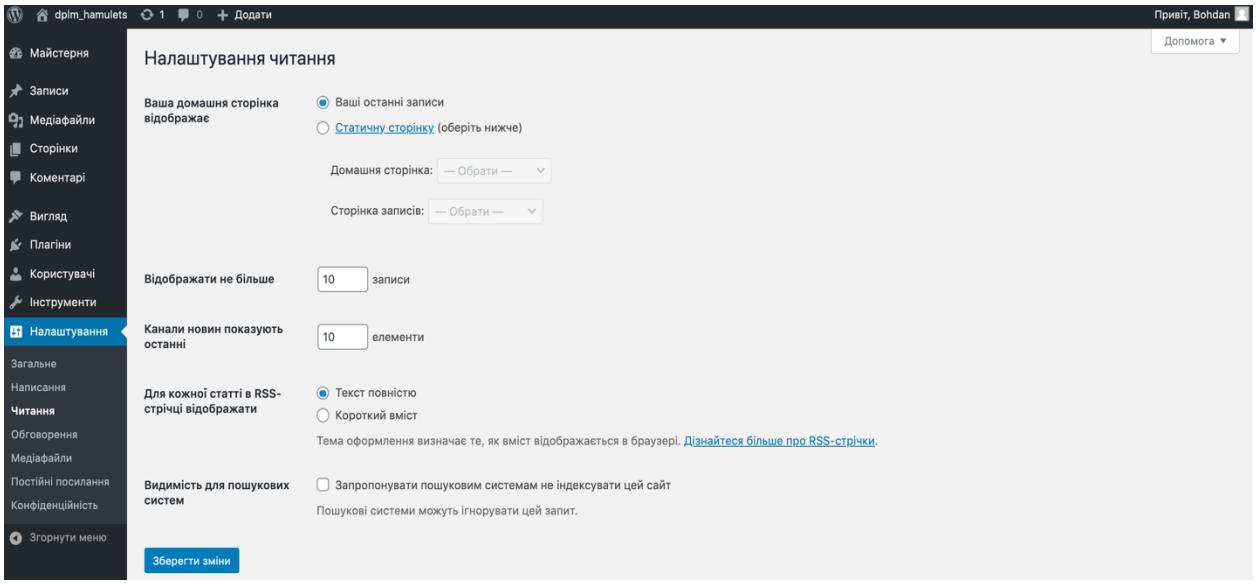

Рис. 3.24. Налаштування читання WordPress

Можливості читання включають:

- Вибір відображення контенту для домашньої сторінки (останні дописи, або статична домашня сторінка) ;
- Налаштування відображення заданих кількості останніх дописів;
- Налаштування каналу новин, з вибором кількості останніх елементів;
- Налаштування для кожної статті в RSS-стрічці (позволяє відображати кроткий, або повний вміст стрічки, варто відзначити що для різних веб браузерів вигляд може відрізнятися;
- Видимість для пошукових систем (пошукові системи можуть індексувати цей вебсайт за вказаними параметрами, або ж ігнорувати).

3.6 Конфігурування шаблону сайту кафедри на WordPress

Окремо можна виділити наступні дві частини конфігурування шаблону сайту на WordPress:

1) Конфігурування меню сайту, зображено на рис. 3.25.:

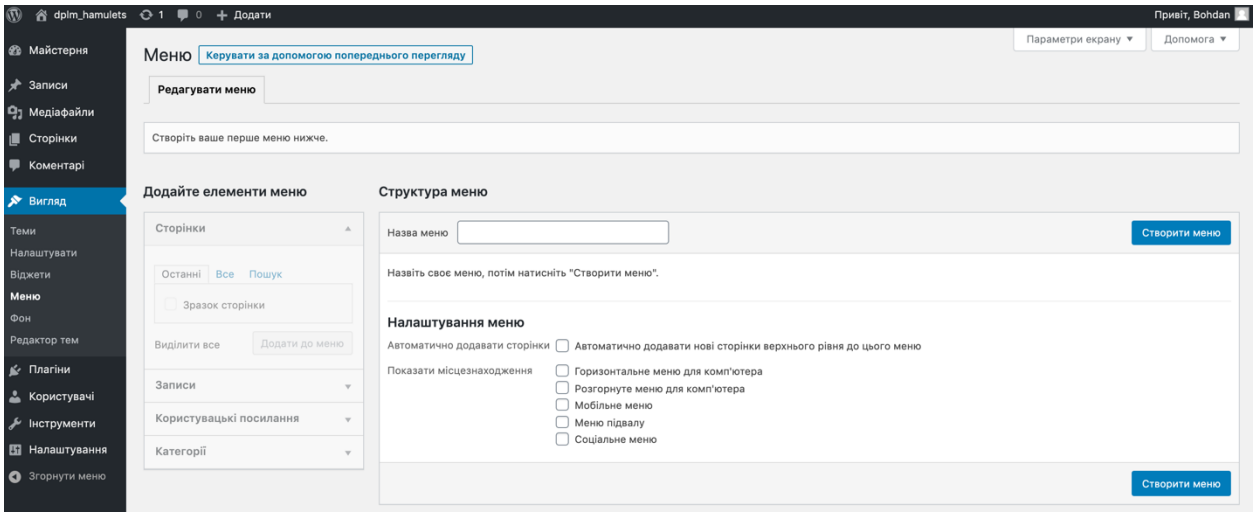

Рис. 3.25. Меню конфігурування WordPress “Вигляд”

Яке позволяє:

- Автоматично додавати нові сторінки верхнього рівня;
- Показувати місцезнаходження;
- Вибирати тип дизайну базуючись на характеристиках пристрою, з якого було здійснено вхід на сайт (горизонтальне меню для комп’ютера, мобільне меню, тощо) ;
- Додати окремі сторінки для меню з дизайном який відрізнятиметься від стандартного дизайну наслідування;
- Додати кастомізовані записи;
- Розміщувати користувацькі посилання (URLs) ;
- Додавати часто використовувані категорії до меню вебсайту.

WordPress Content Management System не обмежує адміністратора до лише одного типу меню, натомість є можливість додати декілька,

налаштування у яких матимуть змогу відрізнятися, та можуть бути відредагованими у будь який момент за бажанням.

2) Конфігурування теми – чи не найважливіший розділ налаштувань WordPress, зображений на рис. 3.26.:

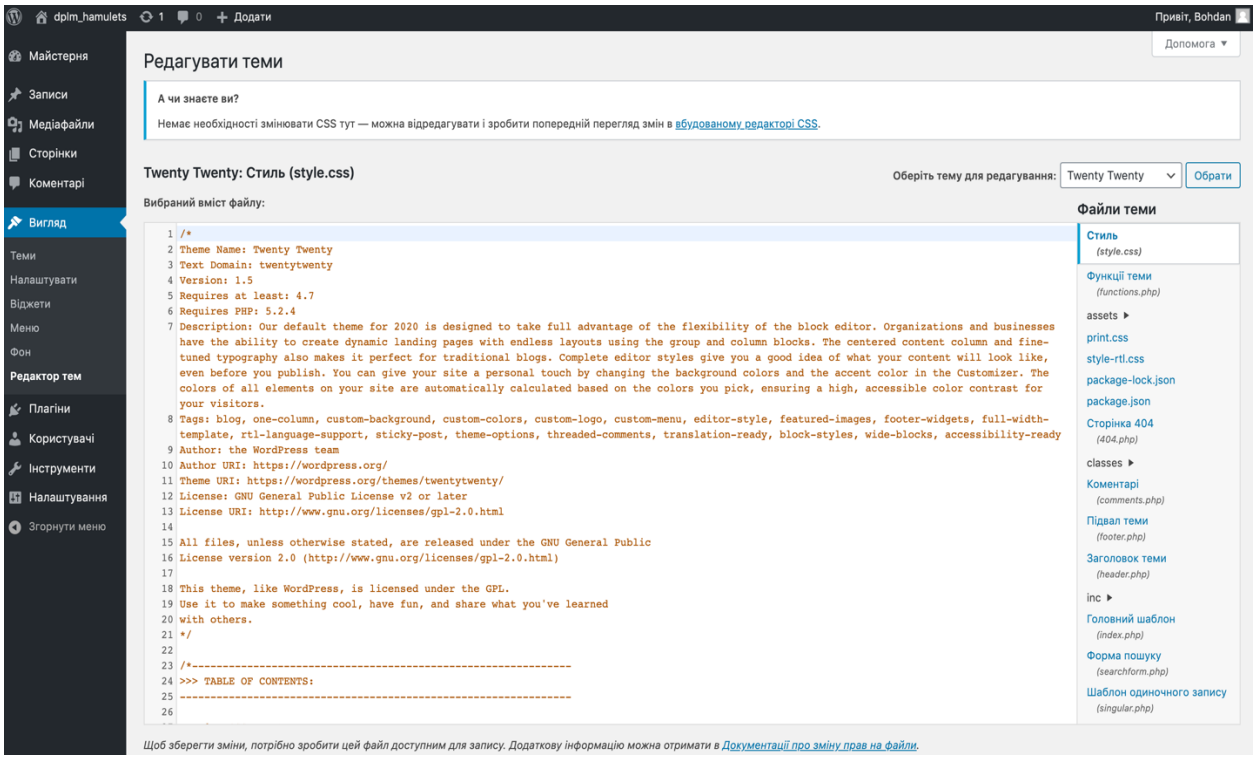

Рис. 3.26. Конфігурація теми WordPress

Даний розділ позволяє вибрати як саму тему (одну з стандартних, наявних за замовчуванням, або зроблену самостійно), так і керування існуючою темою.

У правому куті цієї сторінки розміщений список фалів доступ до яких може бути здійсненим. При виборі необхідного фалу він відкривається по центрі сторінки і вбудований функціонал системиWordPress Content Management System надає можливість редагувати вміст цих файлів. У перелік файлів з якими можна здійснювати ці зміни входять усі файли з наведеної вище таблиці 3.1, у розділі “Робота з обов’язковими елементами шаблону сайту на WordPress” цієї дипломної роботи.

3.7 Висновки до розділу

У даному розділі було детально розглянуто спосіб розробки шаблону сайту кафедри згідно з функціональними вимогами, а зокрема було проведено

огляд Python WordPress XML-RPC специфікації, прикладе використання Python модуля WordPress XML-RPC для функціональної розробки шаблону сайту кафедри, та динамічного керування вмістом та атрибутами розміщеного матеріалу на ньому, перелічено обов’язкові елементи шаблону сайту та описана робота з ними.

Окрему увагу було присвячено темам адміністрування, модерування та конфігурування шаблону сайту кафедри, для забезпечення усіх функціональних вимог його роботи.

РОЗДІЛ 4
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Система з управління охороною праці

Система управління охороною праці (СУОП) — це сукупність органів з управління організацією або підприємством, які на підставі комплексної нормативної документації контролюють, планомірну та цілеспрямовану діяльність щодо здійснення завдань та функцій управління з метою влаштування безпечних, здорових та високопродуктивних умов праці. Створення СУОП здійснюється визначенням мети і об'єкта управління, функцій і методів управління, завдань і заходів щодо охорони праці, складання нормативно-методичної документації та побудови організаційної структури управління. Головною метою управління з охорони праці є створення безпечних умов праці, покращення виробничого побуту, запобігання травматизму і профзахворюванням.

Охорона праці базується на нормативно-технічних документах та законодавчих статей. При управлінні охороною праці не можуть прийматись рішення чи здійснюватися заходи, які суперечать діючому законодавству України, державним нормативним актам про охорону праці, правилам та нормам охорони праці або стандартам безпеки праці

До основних функцій управління охороною праці належать:

- прогнозування і планування робіт, їх фінансування;
- контроль за станом охорони праці та функціонуванням СУОП;
- облік показників, аналіз та оцінка стану умов і безпеки праці;
- організація та координація робіт;
- стимулювання діяльності з охорони праці.

Функція з планування, за основу якої служить прогностичний аналіз, має вирішальне значення в СУОП. Планування та виконання роботи з охорони праці поділяється на: поточне, перспективне та оперативне.

Поточне планування здійснюється у межах календарного року через здійснення розробки та включення відповідних заходів в розділ "Охорона праці" колективного договору.

Перспективне планування охоплює найбільш трудомісткі, важливі й довгострокові за терміном впровадження заходи з охорони праці, впровадження яких, вимагає спільної роботи декількох структурних підрозділів підприємства або організації. Можливість з виконання заходів цього перспективного плану повинна бути підтверджена та обґрунтована розрахунком необхідного матеріально-технічного забезпечення а також фінансових витрат з зазначенням їх джерел фінансування. Основним різновидом з перспективного планування роботи по охороні праці являється розробка комплексного плану підприємства (на 3-5 років) щодо дотримання існуючих правил та покращення стану охорони праці.

Оперативне планування діяльності по охороні праці забезпечується за підсумками з контролю стану охорони праці у структурних підрозділах організації і на підприємстві в цілому або перевірки органів з державного нагляду. Оперативні заходи з усунення виявлених недоліків зазначаються у опублікованому наказі роботодавця.

Оперативне планування діяльності по охорони праці здійснюється за підсумками контролю поточного стану охорони праці в структурних підрозділах організації та на підприємстві у цілому. Заходи зі усунення або виправлення виявлених недоліків чи порушення вимог зазначаються безпосередньо у наказі власника підприємства, який видається за висновками контролю, як додаток до наказу.

Функція СУОП по організації та координації робіт забезпечує формування органів управління охороною праці у всіх рівнях управління та на всіх стадіях процес виробництва або надання послуг з, визначення

обов'язків, прав, відповідальності та порядку взаємодії з особами, які приймають участь у процесі з управління, а також прийняття та реалізації управлінських рішень.

Діюче управління охороною праці можна сформувати лише при наявності комплексної, своєчасної і правдивої інформації про стан охорони праці. Таку інформацію можливо одержати або виявити можливі відхилення від норм безпеки, а також перевірити виконання планів та управлінських рішень можна лише на підставі регулярних та об'єктивних перевірок (контролю).

До основних форм контролю (перевірок) за станом охорони праці належать:

- оперативний контроль - контроль, що проводиться службою охорони праці підприємства;
- громадський контроль - адміністративно-громадський трьох ступеневий контроль;
- відомчий контроль – контроль спеціалізованих вищих органів.

Необхідно зазначити, що крім контролю, здійснюється також і нагляд за охороною праці з боку державних та профспілкових інспекцій.

Адміністрація (роботодавець) для створення безпечних і нешкідливих умов праці працівників і для власної безпеки зобов'язана керуватися переліком таких основних нормативно-законодавчих актів і документів з охорони праці**:**

- Закон України «Про охорону праці» (остання редакція 27 грудня 2020 року);
- Перелік робіт з підвищеною небезпекою (Постанова КМУ №1107 від 3 березня 2020 року);
- Типове положення про службу охорони праці;
- Порядок розробки і затвердження власником нормативних актів про охорону праці, чинних на підприємстві;
- Типове положення про навчання з питань охорони праці;

- Положення про розробку інструкцій з охорони праці;
- Граничні норми підняття та переміщення важких речей неповнолітніми;
- Граничні норми підняття та переміщення важких речей жінками;
- Положення про медичний огляд працівників окремих категорій;
- Перелік посад посадових осіб, які зобов’язані проходити попередню і періодичну перевірку знань з охорони праці;
- Типове положення «Про кабінет охорони праці».
- Типове положення з комісії з питань охорони праці;

Стимулюванням діяльності по охороні праці на підприємстві чи організації є також створення зацікавленості у працівників в забезпеченні здорових та безпечних умов праці. Стимулювання передбачає моральні та матеріальні заохочення, а також покарання за невиконання чи неналежне виконання покладених на конкретну відповідальну особу зобов'язань стосовно забезпечення безпеки з праці або порушення вимог щодо охорони праці. До числа стимулювання належать: премії, винагороди за виконану конкретну роботу, винахідництво та раціональні пропозиції на тему охорони праці. Джерелом з забезпечення фінансового стимулювання діяльності з охорони праці є фонд охорони праці.

4.2 Вимоги до робочого місця користувача ЕОМ: освітлення, рівень шуму, мікроклімат та електромагнітне випромінювання

Приміщення з ЕОМ (Електор-обчислювальними машинами) повинні бути укомплектованими системою з автоматичної пожежної сигналізації у випадку встановлення факту впливу небезпечних чинників або спрацювання відповідних датчиків, згідно з вимогами переліку однотипних за призначенням об'єктів, що підлягають обладнанню автоматичними установками з пожежогасіння та пожежної сигналізації, затвердженого

наказом Міністерства Внутрішніх Справ України та зареєстрованого у Міністерстві Юстиції України, з димовими пожежними сповіщувачем та мобільними вуглекислотними вогнегасниками у розрахунку з 2 шт. на кожні 20 квадратних метрів площі приміщення, з урахуванням границі допустимих концентрацій вогнегасної рідини, згідно з вимог Правил пожежної безпеки в Україні.

У виробничих приміщеннях, та на робочих місцях працівників повинні забезпечуватись оптимальні значення з параметрів мікроклімату: температури, відносної вологості і циркуляції повітря.

Гігієнічні вимоги щодо параметрів робочого середовища включають вимоги до конкретних параметрів мікроклімату, таких як: освітлення, рівень шуму та електромагнітного випромінювання.

Правила з експлуатації ЕОМ встановлюють правила та вимоги по безпеці та санітарно-гігієнічні вимоги щодо обладнання робочих місць користувачів ЕОМ а також працівників, що виконують їх обслуговування, ремонт або налагодження ЕОМ, та роботи з застосуванням ЕОМ, відповідно до сучасного поточного стану технічного забезпечення і найкращих практик у конкретних галузях, та наукових досліджень у сфері організації безпечної роботи з експлуатації ЕОМ та з урахуванням положень міжнародних нормативно-правових актів з цього питання.

Приміщення з ЕОМ мусять мати в наявності як природнє, так і штучне освітлення. Природне джерело освітлення повинно проникати крізь бічні світлопрорізи, зорієнтовані, як правило, на північ чи північний схід, та забезпечувати коефіцієнт природної освітленості не нижче 1,5%. При наявній виробничій потребі дозволяється також використовувати ЕОМ у приміщеннях без природного освітлення за умови узгодження з органами державного нагляду за охороною праці та органами чи установами з санітарно-епідеміологічної служби.

Загальне освітлення повинне бути виконане у формі суцільних або інтервальних ліній світильників, які розміщуються збоку від робочих місць паралельно до лінії зору працівників.

Рівень шуму на робочому місці осіб, які працюють з ЕОМ, визначені постановою Головного Санітарного Лікаря України, Міністерства Охорони Здоров'я України, постановою № 37 від 01.12.1999 року: Санітарні норми виробничого шуму, ультразвуку та інфразвуку ДСН 3.3.6.037-99.

Для забезпечення контролю наявності нормованих рівнів шуму у виробничих приміщеннях, в загальному, та на індивідуальних робочих місцях працівників, застосовуються шумопоглинальні матеріали та засоби, затвердження яких базується на спеціальних інженерно-акустичними розрахунках.

Рівень електромагнітного випромінювання та магнітних полів має відповідати наказу Міністерства Охорони Здоров'я України № 239 від 01.08.1996 року «Про затвердження санітарних правил та норм». Відповідальність за контроль та дотримання цих рівнів несе роботодавець.

4.3 Цивільний захист на обєктах промиловості та запобігання виникненню надзвичайних ситуацій

Згідно з принципами побудови цивільного захисту в Україні слід підкреслити, що територіально-виробничий принцип знайшов втілення в організації цивільного захисту на об'єктах приватного господарства, а також на територіях областей, міст та районів, у тому числі міських та сільських.

Відповідно до Кодексу Цивільного Захисту України та з метою запобігання виникненню надзвичайних ситуацій техногенного характеру (надзвичайні ситуацій), забезпечення стійкого функціонування та господарювання об'єктів підприємницької та управлінської діяльності, в умовах особливого періоду, Кабінет Міністрів України постановив, що дія

цієї постанови поширюється на центральні органи виконавчої влади, міські, обласні, районні державні адміністрації, військово-цивільні адміністрації, органи місцевого самоврядування та об’єкти незалежно від форми власності, порушення виконання яких може завдати шкоди життєво важливим національним інтересам, які провадять діяльність з надання послуг у галузях хімічної промисловості, енергетики, або підлягають охороні та обороні в умовах введення надзвичайного стану чи особливого періоду, являються об’єктами підвищеної небезпеки.

Для координування поточної планової роботи з забезпечення цивільного захисту на об'єкті економіки створюється основний орган управління - штаб цивільного захисту. У складу штабу цивільного захисту входять: начальник штабу та його заступники (помічники) з оперативно-розвідувальної частини, бойової підготовки, житлового сектора.

Керівникам функціональних та територіальних підсистем єдиної державної системи цивільного захисту на підприємствах, установах та організаціях незалежно від їхньої форми власності, на та які поширюється дія цієї постанови, належить забезпечити:

- уточнення планів реагування на надзвичайні ситуації і планів локалізації та ліквідації наслідків аварій, здійснення заходів щодо запобігання їх виникненню;
- готовність до здійснення оповіщення органів управління та сил цивільного захисту, населення про загрозу виникнення або виникнення надзвичайної ситуації та інформування їх про межі поширення, наслідки, способи та методи захисту, а також дії у зоні можливої надзвичайної ситуації;
- готовність наявних сил і засобів цивільного захисту, можливість залучення додаткових сил і засобів у разі виникнення надзвичайних ситуацій;
- створення і використання матеріальних резервів для запобігання виникненню надзвичайних ситуацій і ліквідації їх наслідків;

- спостереження та контроль за ситуацією на об’єктах, на які поширюється дія цієї постанови, територіях цих об’єктів та/або за їх межами, а також здійснення постійного прогнозування можливості виникнення надзвичайних ситуацій, їх масштабів.

Остаточне рішення щодо рівня надзвичайної ситуації з подальшим її відображенням у даних наведення статистики, у тому числі при відсутності повних відомостей для контролю надзвичайної ситуації, приймає спеціально уповноважений центральний орган виконавчої влади, в компетенцію якого входить контроль та забезпечення вирішення питання захисту працівників, населення та територій, від настання надзвичайних ситуацій техногенного чи природнього характеру, з погодженням у разі необхідності із зацікавленими міністерствами та іншими центральними органами виконавчої влади.

4.4 Висновки до розділу

В даному розділі було розглянуто актуальні теми з забезпечення охорони праці працівників та алгоритм з забезпечення коректної реакції з реагування на можливість виникнення надзвичайної ситуації. Були розглянути питання системи з управління охороною праці, вимоги до робочого місця користувача ЕОМ: освітлення, рівень шуму, мікроклімат та електромагнітне випромінювання та цивільний захист на об’єктах промисловості і запобігання виникненню надзвичайних ситуацій. Проаналізовані та наведені у приклад діючі Постанови та Законодавчі акти України які регулюють та встановлюють вимоги з забезпечення охорони праці у даних питаннях. На базі цього аналізу та огляду були отримані знання які забезпечать коректну методику з управління цими ризиками у майбутньому працевлаштуванні.

ВИСНОВКИ

Сайт кафедри призначений для представлення інтересів кафедри та університету в цілому у мережі інтернет, і висвітлення інформації щодо діяльності кафедри; створення комплексної системи інфраструктурного й інформаційного забезпечення навчальної, та наукової діяльності.

Сайт кафедри сприяє вирішенню таких завдань:

- створення позитивного враження від університету;
- оперативне інформування про важливі події у житті кафедри;
- підвищення конкурентоздатності вузу;
- обміну інформацією поміж структурними підрозділами,
- підвищення рівня освіченості університету шляхом застосування новітніх інформаційних технологій.

Будь-яка особа може бути користувачем веб-сайту, необхідні лише технічні можливості для виходу в мережу інтернет, а тому розробка сайту кафедри повинна відбуватися згідно з дотриманням найсуворіший стандартів якості та безпеки.

Функції веб-сайту поділяються на зовнішні, що виконують презентаційну мету та інформаційне обслуговування користувачів, та внутрішні, ті які забезпечують поширення необхідної інформації серед всіх учасників навчального процесу або наукових досліджень.

Вебсайт було побудовано шляхом використання системи управління вмістом WordPress (CMS – Content Management System) з відкритим вихідним кодом та відкритою ліцензією (Open Source), вибір на його користь був зроблений завдяки його наступним основним характеристикам:

- ефективного поділу адміністративної частини на приватну та публічну;
- повний контроль над елементами HTML та CSS з можливістю редагування та попереднього перегляду дизайну;
- легку конфігурацію та зручність;
- суттєву розповсюдженість та затребуваність;
- інсталятор з графічним інтерфейсом;
- підтримку Python XML-RPC модуля;
- контроль доступу ACL (Access Control List).

В ході даної дипломної роботи були детально розглянуті усі необхідні кроки та налаштування, необхідні для реалізації поставленої задачі у Анотації, а саме: «Розробка шаблону сайту кафедри на WordPress згідно вимог з можливістю адміністрування»

БІБЛІОГРАФІЯ

1. І. Бородкіна, Г. Бородкін, Інженерія програмного забезпечення. Посібник для студентів вищих навчальних закладів: Центр навчальної літератури, 2018. 156 с.
2. Р. Мельник, Програмування Веб Застосувань: Львівська Політехніка, 2018. 232 с.
3. В. Гайдаржи, І. Ізварін, Бази даних в Інформаційних системах: Університет «Україна», 2018. 311-355 с.
4. С. Забара, М. Дехтярук, Програмне забезпечення комп'ютерних мереж: Університет «Україна», 2012. 115 с.
5. Ю. Грицюк, Аналіз вимог до програмного забезпечення: Львівська Політехніка, 2018. 401-440 с.
6. І. Хвищун, Програмування і математичне моделювання, Видавництво Ін Юре, 2007. 511 с.
7. О. Васильєв, Програмування мовою Python: Навчальна книга «Богдан», 2019. 206 с.
8. Стандартні методи бібліотеки Python WordPress XML RPC URL: https://python-wordpress-xmlrpc.readthedocs.io/en/latest/ (дата звернення 01.11.2020 р.)
9. Хостинг власного вебсайту URL: https://wordpress.com/features/ (дата звернення 01.11.2020 р.)
10. Аналіз історії розвитку та інформація з публікування нових версій WordPress URL: https://wordpress.org/news/ (дата звернення 02.11.2020 р.)

11. M. MacDonald, WordPress: The Missing Manual: O'Reilly Media, 2020. 56 с.

12. K. Krol, WordPress 5 Complete: Build beautiful and feature-rich websites from scratch: Packt Publishing, 2019. 26-88 с.

13. Офіційна документація по мові програмування Python, URL: https://docs.python.org/3/ (дата звернення 03.11.2020 р.)

14. J. Casabona, HTML and CSS Visual Quick Start Guide: Peachpit Press. 2020. 37-45 с.

15. B. Williams, Professional WordPress: Design and Development: Wrox, 3-d edition, 2015. 187 с.

16. Аналіз розповсюдженості на типових випадків використання WordPress, URL: https://uk.wikipedia.org/wiki/WordPress (дата звернення 03.11.2020 р.)

17. Порівняльна характеристика сучасних веб фреймворків, URL: https://www.educba.com/tutorials/ (дата звернення 03.11.2020 р.)