

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри, канд. техн. наук,
доцент _____ Є. О. Давиденко
підпис

«__» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА ВЕБЗАСТОСУНОК ПРОЄКТУ ДИСТАНЦІЙНОЇ ОСВІТИ МАЙБУТНЬОГО У МЕЖАХ ПРОГРАМИ ERASMUS +KA2

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 409. 21910910

Студента

_____ М. О. Колесніков
підпис

«__» _____ 2023 р.

Керівник ст. Викладач, PhD

_____ К. О. Антіпова
підпис

«__» _____ 2023 р.

Консультант к.т.н., доцент кафедри екології

_____ А. О. Алексєєва
підпис

«__» _____ 2023 р.

м. Миколаїв – 2023 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Є. О. Давиденко

« _____ » _____ 2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи бакалавра

Видано студенту групи 409 факультету комп'ютерних наук

Колеснікову Микиті Олександровичу

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

Вебзастосунок проєкту дистанційної освіти майбутнього в межах Erasmus+ KA2

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

2. Строк представлення кваліфікаційної роботи « _____ » _____ 20__ р.

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

Очікуваним результатом є розроблений вебзастосунок з відображенням розвитку проєкту дистанційної освіти майбутнього в межах Erasmus+KA2

4. Перелік питань, що підлягають розробці:

- дослідження предметної області та аналіз існуючих аналогів;
- формування специфікації вимог до ПЗ;
- визначення архітектури для проєктування ПЗ;
- моделювання та проєктування ПЗ;
- розробка ПЗ;
- здійснення тестування роботи ПЗ;
- проведення аналізу результатів розробки ПЗ;

5. Перелік графічних матеріалів

Презентація до КРБ

6. Завдання до спеціальної частини

Аналіз охорони праці на робочих місцях фахівців з інформаційних технологій

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
А. О. Алексеева	Кафедра екології	

Керівник роботи викл. Кафедри ІІЗ Антіпова Катерина Олександрівна

(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

Колесніков Микита Олександрович

(прізвище, ім'я, по батькові)

(підпис)

Дата видачі завдання « » _____ 2023р.

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної роботи

Тема: Вебзастосунок проєкту дистанційної освіти майбутнього в межах Erasmus+ КА2

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	04.04.2023	05.04.2023	виконано
2.	Пошук літератури за темою кваліфікаційної роботи	06.04.2023	07.04.2023	виконано
3.	Складання календарного плану роботи	08.04.2023	08.04.2023	виконано
4.	Аналіз предметної області	09.04.2023	09.04.2023	виконано
5.	Розробка проєктних рішень	10.04.2023	12.04.2023	виконано
6.	Моделювання та конструювання ПЗ	13.04.2023	19.04.2023	виконано
7.	Кодування розробленого ПЗ	20.04.2023	20.05.2023	виконано
8.	Тестування ПЗ, аналіз результатів тестування	21.05.2023	24.05.2023	виконано
9.	Розробка частини КРБ з охорони праці	25.05.2023	01.06.2023	виконано
10.	Відгук керівника КРБ	02.06.2023	03.06.2023	виконано
11.	Оформлення КРБ та презентації КРБ	04.06.2023	06.06.2023	виконано
12.	Попередній захист КРБ	07.06.2023	07.06.2023	виконано
13.	Рецензування КРБ	08.06.2023	18.06.2023	
14.	Завершення оформлення КРБ та презентації	19.06.2023	21.06.2023	виконано
15.	Захист кваліфікаційної роботи	27.06.2023	27.06.2023	

Розробив студент Колесніков Микита Олександрович

(прізвище, ім'я, по батькові)

(підпис)

«__» _____ 2023 р

Керівник роботи ст. викл. каф. ІПЗ, PhD Антіпова Катерина Олександрівна

(посада, прізвище, ім'я, по батькові)

(підпис)

«__» _____ 2023 р

АНОТАЦІЯ

до кваліфікаційної роботи бакалавра
«Вебзастосунок відображення розвитку дистанційної освіти майбутнього у межах
програми Erasmus+ KA2»

Студент 409 групи: Колесніков Микита Олександрович

Керівник: ст. викладач, PhD Антіпова К. О.

Дана робота присвячена розробці вебзастосунку для інформування аудиторії про розвиток проєкту дистанційної освіти майбутнього у межах програми Erasmus+ KA2.

Об'єкт дослідження – процес розробки вебзастосунку для відображення розвитку дистанційної освіти майбутнього у межах програми Erasmus+ KA2.

Предмет дослідження – технології, необхідні для представлення необхідної інформації через вебзастосунок розвитку дистанційної освіти.

Метою роботи є підвищення обізнаності більшої аудиторії людей про розвиток проєкту дистанційної освіти майбутнього у межах програми Erasmus+ KA2, шляхом створення вебзастосунку з розвитку даного проєкту дистанційної освіти.

Кваліфікаційна робота складається зі вступу, 4 розділів, висновків та переліку джерел посилань.

У вступі визначається актуальність теми, мета, предмет та об'єкт дослідження.

У першому розділі проведено аналіз існуючих вебзастосунків-аналогів, визначення функціоналу, переваг, недоліків, технології за допомогою яких було створено вебзастосунок. Також проведено формування та опис специфікації вимог вебзастосунку, що розробляється.

У другому розділі проведено моделювання та розробку структури застосунку, що розробляється.

У третьому розділі проводиться огляд мов програмування, їх бібліотек та інших технологій, що використовуються для розробки вебзастосунку. Також оглядається процес розробки програмного забезпечення.

У четвертому розділі демонструється результат виконаної роботи, тестування вебзастосунку.

У висновках проводиться аналіз виконаних робіт та отриманих результатів.

Кваліфікаційна робота бакалавра викладена на 53 сторінках, містить 4 розділи, 42 ілюстрації, 8 таблиць, 10 джерел в переліку посилань.

Ключові слова: *вебзастосунок Wordpress, розробка вебзастосунку, освітній проєкт, дистанційна освіта, освіта майбутнього.*

ABSTRACT

of the Bachelor's Thesis

«Web application displaying the development of distance education of the future project within the Erasmus+ KA2 program»

Student: Mykyta Kolesnikov

Supervisor: Senior Lecturer, PhD Kateryna Antipova

This work is devoted to the development of a web application to inform the audience about the development of the distance education of the future project within the Erasmus+ KA2 program.

The object of research is the process of developing a web application to reflect the development of distance education of the future within the Erasmus+ KA2 program.

The subject of research is the technologies necessary to present the necessary information through the web application for the development of distance education.

The aim of the work is to raise awareness of a larger audience of people about the development of the distance education project of the future within the Erasmus+ KA2 program, by creating a web application for the development of this distance education project.

The qualification work consists of an introduction, 4 chapters, conclusions and a list of reference sources.

The introduction determines the relevance of the topic, purpose, subject and object of research.

The first section analyzes existing web applications-analogues, determines the functionality, advantages, disadvantages, technologies with the help of which the web application was created. Also, the formation and description of the specification of the requirements of the developed web application was carried out.

In the second section, modeling and development of the structure of the developed application is carried out.

The third section provides an overview of programming languages, their libraries and other technologies used to develop a web application. The software development process is also reviewed.

The fourth section demonstrates the result of the work done, testing the web application.

The conclusions analyze the work performed and the results obtained.

The qualification work of the bachelor consists of 53 pages, contains 4 sections, 32 illustrations, 8 tables, 10 sources in the list of references.

Keywords: *Wordpress web application, web application development, educational project, distance education, education of the future.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	5
1.1 Огляд застосунків-аналогів.....	5
1.2 Аналіз системи, що розробляється.....	7
1.3 Специфікація вимог	8
Висновки до розділу 1	13
2 МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	14
2.1 Етапи реалізації проєкту.....	14
2.2 Створення use case	15
2.3 Діаграми взаємодії	17
2.4 Діаграми станів та переходів	19
2.5 Діаграми діяльності	21
Висновки до розділу 2	24
3 ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ ТА ОГЛЯД СТЕКУ ЗАДІЯНИХ ТЕХНОЛОГІЙ	25
3.1 Розробка UML-діаграм.....	25
3.2 Огляд технологій.....	29
Висновки до розділу 3	35
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ ТА ПРОВЕДЕННЯ ТЕСТУВАННЯ	36
4.1 Огляд дизайну вебзастосунку	36
4.2 Розробка функціоналу адмін-панелі	41
4.3 Додавання мультимовності на сайт	44
4.4 SEO-оптимізація	45
4.4 Тестування	47
Висновки до розділу 4	50
ВИСНОВКИ.....	51
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	52
ДОДАТОК А АКТ ВПРОВАДЖЕННЯ	53

ПЕРЕЛІК СКОРОЧЕНЬ

БД – База даних

КРБ – Кваліфікаційна робота бакалавра

ПЗ – Програмне забезпечення

CSS – Cascading Style Sheets

JS – JavaScript

ВСТУП

Актуальність теми кваліфікаційної роботи бакалавра зумовлена ситуацією в сучасному світі, де протягом декількох минулих років сталась пандемія COVID-19 та розпочалось повномасштабне вторгнення в нашу країну. Внаслідок цього багато людей покинули свої домівки та їм довелося розпочинати життя з чистого листа.

З цієї ж причини довелося переїхати і певній кількості навчальних закладів і перевести студентів на онлайн-навчання. Даний проєкт має в собі ціль допомогти абітурієнтам та студентам інформацією щодо дистанційного навчання.

Об'єкт роботи: процес розробки вебзастосунку для відображення розвитку дистанційної освіти майбутнього у межах програми Erasmus+ KA2.

Предмет роботи: технології, необхідні для представлення необхідної інформації через вебзастосунок дистанційної освіти.

Метою роботи є підвищення обізнаності більшої аудиторії людей про розвиток проєкту дистанційної освіти майбутнього у межах програми Erasmus+ KA2, шляхом створення вебзастосунку з розвитку даного проєкту дистанційної освіти.

Для досягнення поставленої мети необхідно вирішити наступні **завдання:**

- аналіз предметної сфери;
- обговорення деталей проєкту;
- дизайн та розробка логотипу проєкту;
- дизайн зовнішнього вигляду сайту проєкту;
- верстка сайту проєкту;
- розробка та встановлення необхідного функціоналу адмін-панелі;
- тестування та фікс багів;
- відправлення проєкту на production server та його запуск в мережу Інтернет.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Огляд застосунків-аналогів

Для проведення аналізу аналогів обрано наступні вебзастосунки: MTU (табл. 1.1), Coursera (табл. 1.2), Stanford Online (табл. 1.3).

MTU [1]

Застосунок є офіційним вебсайтом Манстерського технологічного університету, розташованого в Ірландії. Вебзастосунок має сучасний та зручний інтерфейс, а також активно оновлюється. Головним же недоліком вебзастосунку є відсутність мультимовності.

Таблиця 1.1 – Опис MTU

Назва	MTU
Розробник	Munster Technological University
Архітектура	Web application
Мова реалізації	CMS Shopify, PHP, Ruby
Функції	1. Додавання новин; 2. Додавання наукових публікацій; 3. Контактна інформація; 4. Актуальна інформація про університет.
Переваги	1. Сучасний та зручний дизайн; 2. Активне оновлення сайту; 3. Велика кількість згрупованої інформації.
Недоліки	1. Відсутність мультимовності. 2. Баги в анімаціях.
Вебсайт	https://www.mtu.ie/

Coursera [2]

Coursera є одним з найбільших у світі провайдерів відкритих онлайн курсів, який був заснований у 2012 році. Наразі встановлено, що більше 150 університетів пропонують більше 4000 курсів через дану платформу.

Таблиця 1.2 – Опис Coursera

Назва	Coursera
--------------	----------

Кінець таблиці 1.2

Розробник	Coursera Inc.
Архітектура	Web application
Мова реалізації	CMS Gatsby, CMS Wordpress, PHP, JavaScript
Функції	<ol style="list-style-type: none"> 1. Перегляд доступних курсів. 2. Додавання нових курсів. 3. Отримання професійних сертифікатів. 4. Отримання ступенів навчання онлайн. 5. Проходження доступних курсів.
Переваги	<ol style="list-style-type: none"> 1. Можливість використання сайту для онлайн навчання. 2. Велика кількість інформації та можливих курсів. 3. Можливість проходження спеціальних безкоштовних курсів
Недоліки	<ol style="list-style-type: none"> 1. Трохи застарілий дизайн. 2. Перевантажений header.
Вебсайт	coursera.org

Stanford Online [3]

Stanford Online це вебсайт створений членами Стенфордського університету, який пропонує користувачам проходження як платних, так і безоплатних курсів різних тематик та напрямлень.

Таблиця 1.3 – Опис Stanford Online

Назва	Stanford Online
Розробник	Stanford University
Архітектура	Web application
Мова реалізації	CMS Wordpress, CMS Drupal, PHP, JavaScript
Функції	<ol style="list-style-type: none"> 1. Отримання інформації про університет. 2. Публікація новин. 3. Публікація інформації про спеціальні програми навчання. 4. Додавання безкоштовних онлайн-курсів. 5. Додавання безкоштовних онлайн-лекцій.
Переваги	<ol style="list-style-type: none"> 1. Швидкодія сайту. 2. Дизайн та зручність сайту.

Кінець таблиці 1.3

Переваги	3. Присутність онлайн-підтримки. 4. Присутність пошуку по сайту.
Недоліки	1. Анімації елементів сайту «дерев'яні». 2. Відсутність мультимовності.
Вебсайт	https://online.stanford.edu/

1.2 Аналіз системи, що розробляється

Основною метою розробки даної системи є донесення до користувачів інформації про можливості дистанційної освіти.

Таблиця 1.4 – Опис системи що розробляється

Основні задачі	1. створення новин, пов'язаних з діяльністю проєкту; 2. авторизація для адміністраторів та редакторів проєкту; 3. створення списку учасників проєкту; 4. додавання з часом нових учасників проєкту; 5. створення інформації про результати проєкту, розділені по певним категоріям;
Користувачі системи	1. Відвідувач; 2. Адміністратор; 3. Редактор.
Сценарії роботи	1. Пошук та зміна новин. Адміністратор може шукати новини за темою та датою. Після цього адміністратор може або переглянути знайдені новини, або змінити інформацію про них. 2. Пошук та зміна результатів проєкту. Адміністратор може шукати результати за категорією та датою. Після цього адміністратор може або переглянути знайдені результати, або змінити інформацію про них. 3. Перегляд новин. Користувач може переглядати існуючі новини проєкту. 4. Перегляд результатів проєкту. Користувач може переглядати існуючі результати проєкту.

Кінець таблиці 1.4

Засоби апаратної та програмної реалізації	<ol style="list-style-type: none">1. IDE Visual Studio Code;2. PHP;3. Wordpress;4. JavaScript;5. CSS;6. HTML;7. JQuery;8. phpMyAdmin;9. MySQL;10. Open Server Panel;
Вихідні дані	<ol style="list-style-type: none">1. представлення користувачам інформації щодо проєкту через сторінки вебсайту;2. представлення користувачам інформації про учасників проєкту.

1.3 Специфікація вимог

ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

Призначення системи, для якої розробляється програмне забезпечення

Призначенням застосунку є подання інформації про дистанційну освіту абітурієнтам та студентам.

Погодження, ухвалені в програмній документації

Було узгоджено, що для створення ПЗ буде використовуватись CMS Wordpress та мови програмування PHP, JS.

Межі проєкту ПЗ

Дата завершення роботи над проєктом – 20.05.2023 р.

ЗАГАЛЬНИЙ ОПИС

Сфера застосування

Основною сферою застосування даного ПЗ – є освітня сфера, для забезпечення інформацією студентів.

Характеристики користувача

Для користування вебсайтом, користувачу необхідно мати гаджет (смартфон, ноутбук, ПК) зі стабільним доступом до мережі Інтернет.

Загальна структура та склад системи

Система складається з таких модулів:

– Модуль авторизації користувачів – цей модуль відповідає за авторизацію адміністраторів та редакторів в адмін-панелі, для внесення необхідних на сайті, наприклад зміни в сторінках в ролі редактора, або додаванні нових облікових записів у якості адміністратора.

– Модуль адміністрування – цей модуль відповідає за функціонал адмін-панелі, з якою будуть працювати адміністратори застосунку. У адміністраторів наявний весь доступ до інформації на вебсайті з можливістю її зміни.

– Модуль новин – цей модуль відповідає за надання відвідувачам вебсайту можливості переглядати новини. Новини мають право редагувати редактори, адміністратори, а також додаткові ролі, які створені спеціально адміністратором.

– Модуль результатів – цей модуль надає можливість відвідувачам вебзастосунку переглядати результати проєкту. Аналогічно з модулем новин, редагувати результати можуть, як редактори, так і адміністратори. Модуль дозволяє розміщення файлів результатів проєкту на сторонніх ресурсах, як наприклад Google Drive.

– Модуль активностей – цей модуль дозволяє відвідувачам переглядати активності проєкту, серед яких події та зустрічі та дозволяє отримати інформацію по ним. Аналогічно з модулем новин та модулем результатів – даний модуль мають можливість редагувати редактори та адміністратори.

Загальна структура системи має наступний вигляд:

1. Модуль авторизації користувачів

- Форма логіну для існуючих користувачів.
- Форма додавання нових користувачів адміністратором вебсайту в адмін-панелі.

2. Модуль адміністрування

- Редактор для додавання нових сторінок, постів, новин, результатів, активностей.
- Вкладка додавання та редагування користувачів системи, створення нових ролей в системі.
- Вкладка налаштування плагінів.
- Вкладка налаштування теми вебзастосунку.
- Інструменти з імпорту та експорту вебсайту.
- Вкладка з загального налаштування WordPress.

3. Модуль новин

- Додавання новин на вебсайт.
- Редагування новин.
- Видалення новин.
- Додавання категорій та тегів до новин.

4. Модуль результатів

- Додавання результатів проєкту на вебсайт.
- Редагування результатів проєкту.
- Видалення результатів проєкту.
- Додавання спеціальних тегів до результатів проєктів.
- Розподіл результатів проєктів по категоріям.

5. Модуль активностей

- Додавання активностей проєкту на вебсайт.
- Редагування активностей проєкту.
- Видалення активностей проєкту.
- Розподіл активностей між двома категоріями – події та зустрічі.

ФУНКЦІЇ СИСТЕМИ ВЕБЗАСТОСУНКУ

Функція перегляду новин

Опис функції

Дана функція дозволяє користувачу переглядати новини та інформацію про них.

Вхідні і вихідна інформація

Вхідна інформація – назва новини, дата створення.

Вихідна інформація – інформація новини.

Функціональні вимоги

База даних з інформацією про новини та наявний доступ до інтернету.

Функціональні вимоги:

- Користувачі повинні мати змогу переглядати опубліковані новини;
- Користувачі повинні мати змогу переглядати учасників проєкту;
- Користувачі повинні мати змогу переглядати опубліковані результати проєкту;
- Адміністратор та редактор повинні мати змогу додавати нові новини до системи, оновлювати інформацію в новинах та видаляти їх із системи.
- Адміністратор повинен мати змогу додавати нових користувачів у систему, змінювати інформацію про них та їхні привілеї.
- Адміністратор та редактор повинні мати змогу додавати нові результати до системи, оновлювати інформацію в існуючих результатах, розділяти результати по категоріям та видаляти їх із системи.

Нефункціональні вимоги:

- ПЗ повинно працювати на платформах Windows, MacOS, Linux;
- ПЗ повинно працювати у сучасних браузерях (Google Chrome, Opera та інші).
- Швидкість реакції розробленої системи повинна бути не менше 2 секунд на запит або дію користувача.
- Система має бути доступною цілодобово, за виключенням проблем з електропостачанням та серверами.
- Система повинна мати хороший рівень безпеки для передбачення DDoS-атак та зламу адмін-панелі.
- Інтерфейс системи має бути однаково зрозумілим як для користувача, так і для адміністратора.

Нормативно-довідкова інформація (класифікатори, довідники тощо)

В цьому пункті вимоги відсутні.

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Для роботи вебзастосунку необхідно, щоб технічний засіб користувача відповідав мінімальним системним вимогам для використання браузера

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура програмної системи

Система складається з бази даних, серверу де розташований вебзастосунок, а також клієнтської частини.

Системне програмне забезпечення

Вебзастосунок реалізовується за допомогою CMS WordPress. В якості бази даних системи виступає MySQL.

Програмне забезпечення ведення інформаційної бази

Всі операції виконуються через MySQL, з використанням інтерфейсу розроблюваного програмного забезпечення.

Мова і технологія розробки програмного забезпечення

PHP, JavaScript, CMS WordPress.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

Інтерфейс користувача

Інтерфейс має бути максимально комфортним для користувача, а також інтуїтивно зрозумілим. Кольорова гамма інтерфейсу не має приносити дискомфорт користувачеві.

Апаратний інтерфейс

В якості апаратного інтерфейсу має виступати будь який пристрій, який дозволяє робити вихід в мережу Інтернет.

Програмний інтерфейс

CMS WordPress – система менеджменту контенту, що дозволяє максимально ефективно та швидко додавати та змінювати вебзастосунок, а також розширюючи його функціонал.

Висновки до розділу 1

У першому розділі КРБ було проведено аналіз існуючих застосунків-аналогів та розглянуто їх функціонал, переваги та недоліки. Для цього було обрано застосунки, пов'язані з освітньою сферою.

В даному розділі також було описано специфікацію вимог до ПЗ вебзастосунку проєкту розвитку дистанційної освіти майбутнього. В ньому представлено функціональні та нефункціональні вимоги, які описують вимоги до користувача даного ПЗ.

Також було створено аналіз системи, що розробляється. Було поставлено основні задачі даної системи, а також було виділено користувачів даного застосунку, а саме: користувач, редактор та адміністратор. Було виділено сценарії роботи майбутнього вебзастосунку. Також було обрано засоби апаратної реалізації проєкту. В цій частину було обрано як ПЗ для роботи з застосунком (VS Code, OpenServerPanel), так і мови програмування для розробки застосунку (PHP, JS).

2 МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Етапи реалізації проєкту

В процесі розробки програмного продукту важливим етапом є планування. Планування дозволяє команді розробників поставити завдання, створити розуміння того, що потрібно зробити в проєкті, які для цього необхідні інструменти та технології, скільки часу потрібно на виконання програмного продукту, його бюджет. Планування формується на підставі вимог та очікувань замовника, а також на силах створеної команди розробників.

Для створення програмного продукту формується план дій та список майбутнього функціоналу продукту.

Для виконання поставленої мети проєкту необхідно виконати наступні завдання:

- аналіз предметної сфери;
- обговорення деталей проєкту;
- дизайн та розробка логотипу проєкту;
- дизайн зовнішнього вигляду сайту проєкту;
- верстка сайту проєкту;
- розробка та встановлення необхідного функціоналу адмін-панелі;
- тестування та фікс багів;
- відправлення проєкту на production server та його запуск в мережу Інтернет.

Планування розробки включає в себе наступні етапи:

- створення сценаріїв;
- побудова діаграми станів;
- побудова діаграми взаємодії;
- побудова діаграм компонентів та розгортання;
- побудова діаграми класів.

2.2 Створення use case

Use case або сценарії використання – це перелік дій, які необхідно виконати користувачу для взаємодії з системою.

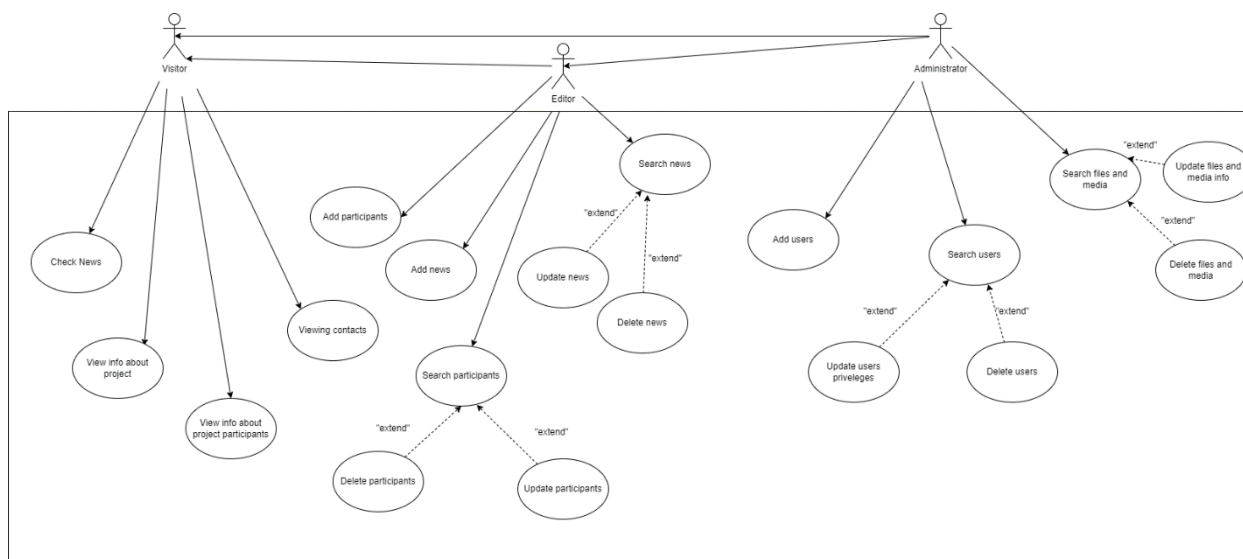


Рисунок 2.1 – Use case діаграма

Короткий use case (створення новини)

Адміністратор вебзастосунку заходить на сайт. Переходить по спеціальному посиланню. Виконує вхід у свій аккаунт. Після цього відкривається панель адміністрування. В панелі адміністрування, адміністратор додає новину про проведену зустріч, захід. Далі переходить на сторінку відображення новин та перевіряє успішність додавання новини. Потім переходить до панелі адміністрування та додає зображення, відео та інші ресурси. Все коректно додано. Адміністратор переходить на сторінку новин для перегляду коректності відображення новини. Все коректно відображається.

Поверхневий use case (створення новини)

Головний сценарій (успішний):

Адміністратор вебзастосунку заходить на сайт. Переходить по спеціальному посиланню. Виконує вхід у свій аккаунт. Після цього відкривається панель адміністрування. В панелі адміністрування, адміністратор додає новину про проведену зустріч, захід. Далі переходить на сторінку відображення новин та перевіряє успішність додавання новини.

Потім переходить до панелі адміністрування та додає зображення, відео та інші ресурси. Все коректно додано. Адміністратор переходить на сторінку новин для перегляду коректності відображення новини. Все коректно відображається.

Альтернативні сценарії:

- 1) Адміністратор ввів неправильні дані користувача при вході.
- 2) Перевищений розмір зображення, відео або іншого ресурсу.
- 3) Збій системи на сервері, внаслідок чого вебзастосунок не працює.
- 4) Збій додавання новини в базу даних, повідомлення про помилку.
- 5) Перевищений розмір БД, новину не додано, отримання помилки.

Таблиця 2.1 – Повний use case (відправлення пакетів на сервер)

Primary actor	Адміністратор
Scope	Вебзастосунок проєкту дистанційної освіти
Level	Мета адміністратора
Preconditions	Адміністратор вказав коректні дані.
Stakeholders and interests	Відвідувач: Зацікавлений у перегляді новин про проєкт та отримання на їх основі нової інформації. Адміністратор: Зацікавлений у коректному додаванні новини та подальшій перевірці коректності її додавання. Керівник проєкту: Зацікавлений у коректності відображення даних.
Main success scenario	<ol style="list-style-type: none"> 1. Після коректного вводу даних для входу, адміністратор опинився в панелі адміністрування. 2. Перейшов на вкладинку перегляду доданих новин. 3. Натиснув на кнопку «додати новину» та перейшов в редактор нової новини. 4. Ввів необхідну інформацію новини. 5. Зберіг та опублікував новину. 6. Перейшов на сайт, переглянув коректність додавання новини.

Кінець таблиці 2.1

Main success scenario	<ol style="list-style-type: none"> 7. Повернувся до редагування новини. 8. Додав необхідні ресурси до новини (зображення, відео і тд.). 9. Зберіг оновлену інформацію та ресурси новини. 10. Перевірив коректність відображення новини. 11. Вийшов з панелі адміністрування.
Result	Адміністратор додав новину
Extensions	<ol style="list-style-type: none"> 1. Адміністратор ввів неправильні дані користувача при вході. 2. Перевищений розмір зображення, відео або іншого ресурсу. 3. Збій системи на сервері, внаслідок чого вебзастосунок не працює. 4. Збій додавання новини в базу даних, повідомлення про помилку. 5. Перевищений розмір БД, новину не додано, отримання помилки.
Special requirements	<ol style="list-style-type: none"> 1. Система має виконувати запити з максимальною затримкою в 2 секунди. 2. Підтримка мультимовності.
Frequency of occurrence	Система працює безперервно, за виключенням моментів її оновлення, або перебоїв з електропостачанням та інтернетом.

2.3 Діаграми взаємодії

Діаграми взаємодії використовуються для розуміння підходу до того як користувач має взаємодіяти із застосунком [5]. Під час моделювання застосунку було розроблено 3 діаграми взаємодії (рис. 2.2-2.4). Діаграми, що моделюють дії користувача під час отримання інформації про учасників

проєкту, а також створення учасників проєкту є схожими, лише з декількома відмінностями.

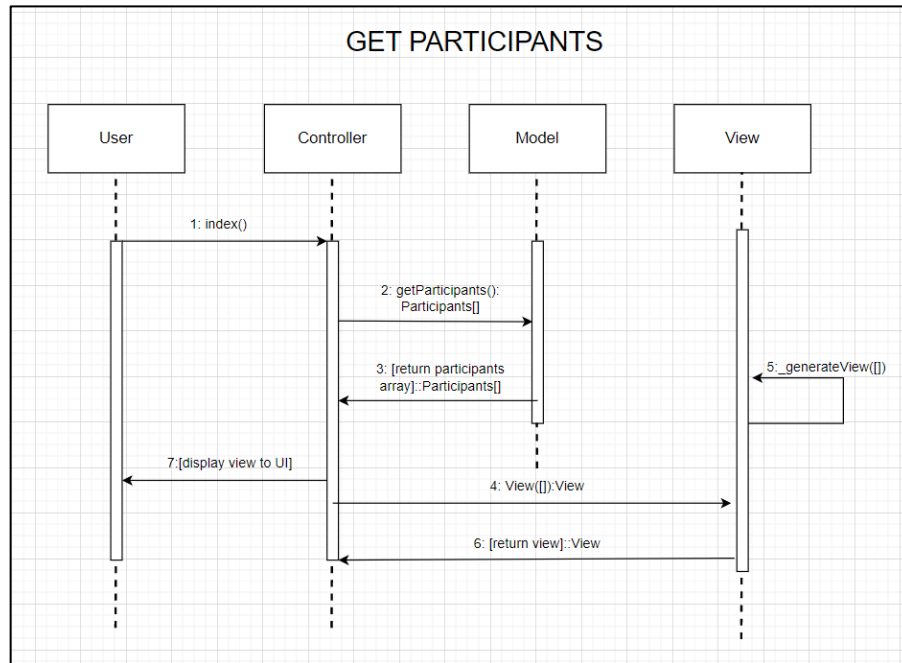


Рисунок 2.2 – Діаграма взаємодії перегляду учасників проєкту

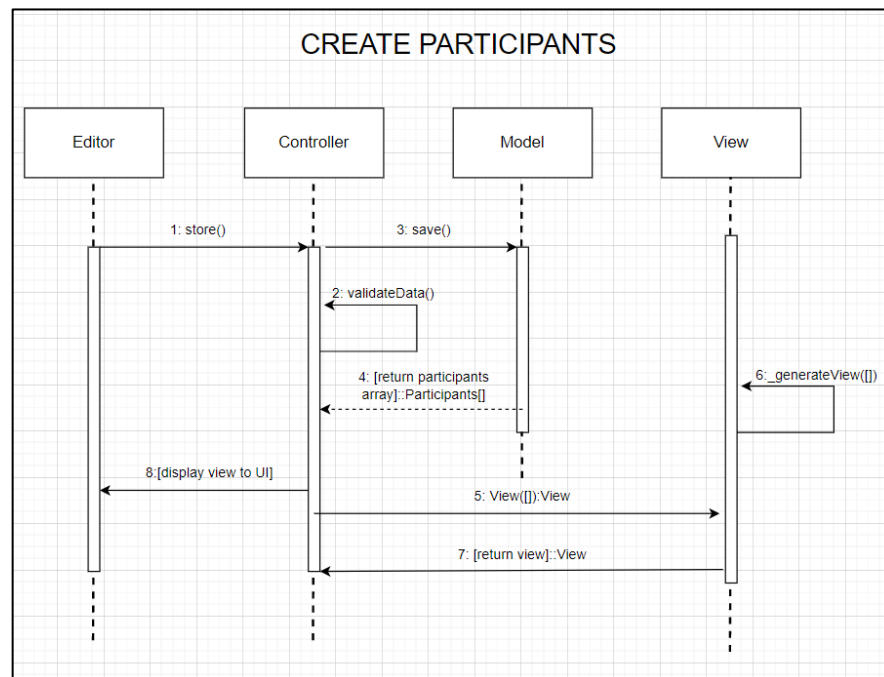


Рисунок 2.3 – Діаграма взаємодії створення нового учасника проєкту

Діаграма взаємодії користувача, який виконує вхід у систему відрізняється від минулих діаграм, оскільки вона пов'язана не з отриманням і роботою з даними про учасників проєкту, а роботою з даними користувача.

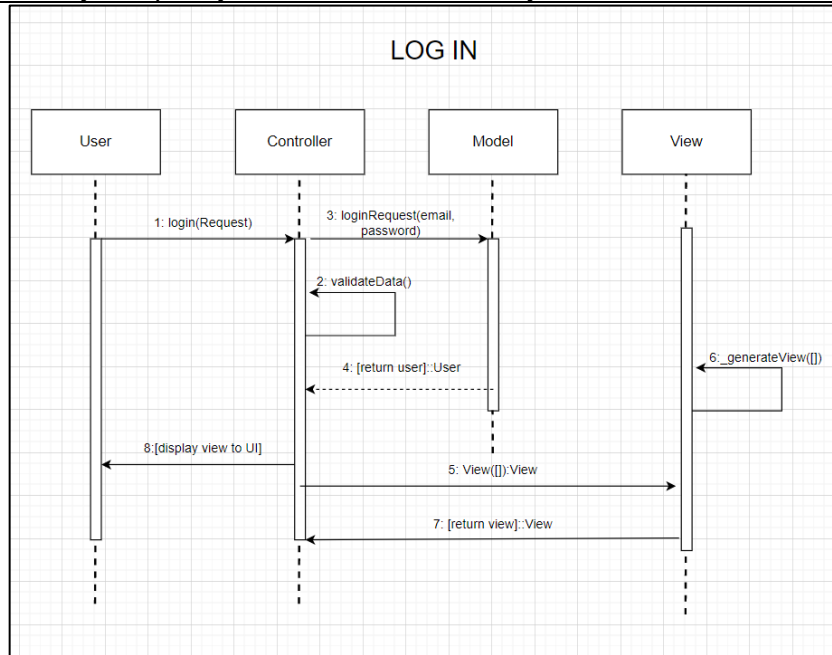


Рисунок 2.4 – Діаграма взаємодії відправлення інформації про пакети

Тут можна побачити, що в залежності від правильності введених даних під час логіну залежить, чи виведе користувачеві сторінку після логіну, чи ні.

2.4 Діаграми станів та переходів

Діаграми станів та переходів (statechart diagrams) показують різні стани, в яких може перебувати об'єкт або система, а також переходи між цими станами на основі різних подій або умов [6]. Вони використовуються під час моделювання поведінки складних систем.

Для майбутньої системи було розроблено 3 діаграми станів та переходів:

- 1) діаграма станів учасників проєкту (рис. 2.5);
- 2) діаграма станів результатів проєкту (рис. 2.6);
- 3) діаграма стану логіну в систему (рис. 2.7).

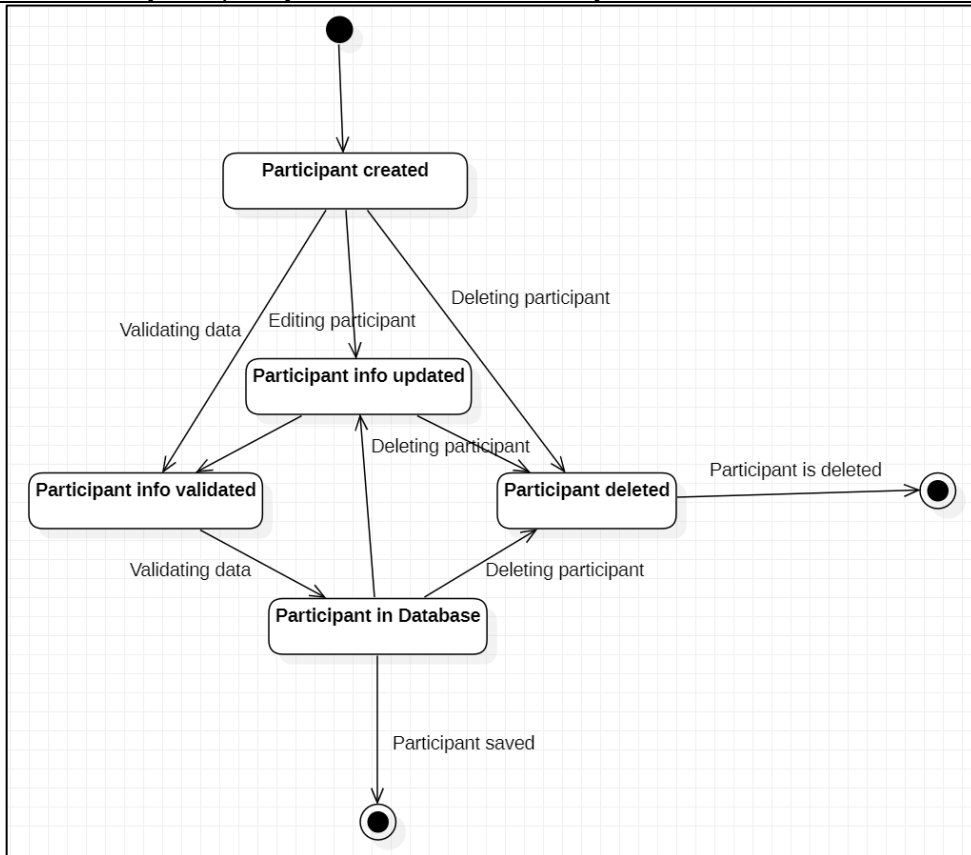


Рисунок 2.5 – Діаграма станів учасників проєкту

На діаграмі станів учасників проєкту видно, що вхідною точкою є створений учасник проєкту. Після цього відкривається простір для дій з ним, як наприклад його видалення, валідація або оновлення.

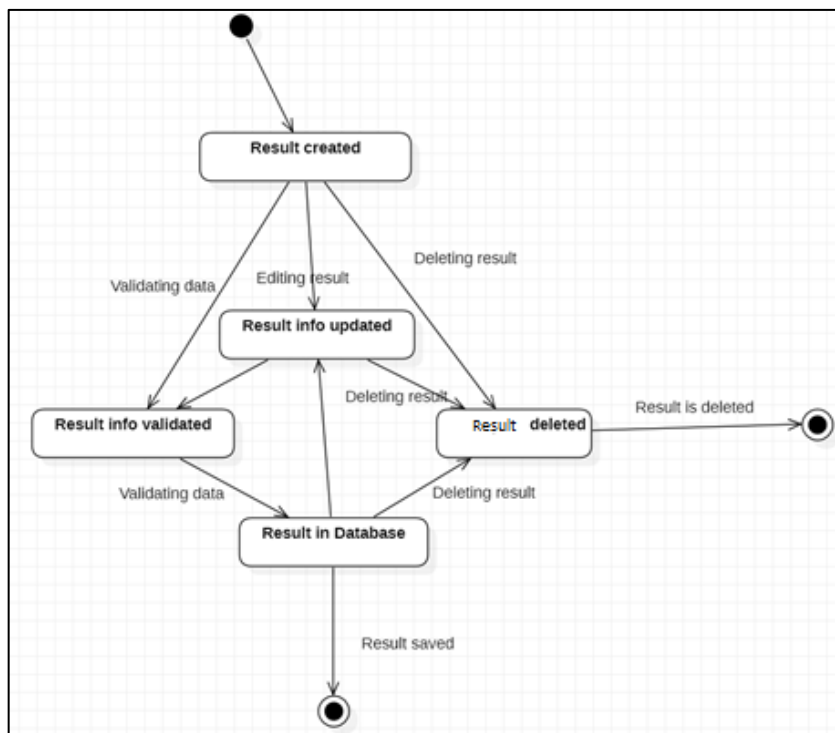


Рисунок 2.6 – Діаграма станів результатів проєкту

Аналогічно діаграмі станів учасників проекту – діаграма станів результатів проекту має вхідну точку у вигляді створеного результату і в подальшому його можна видаляти, валідувати та редагувати.

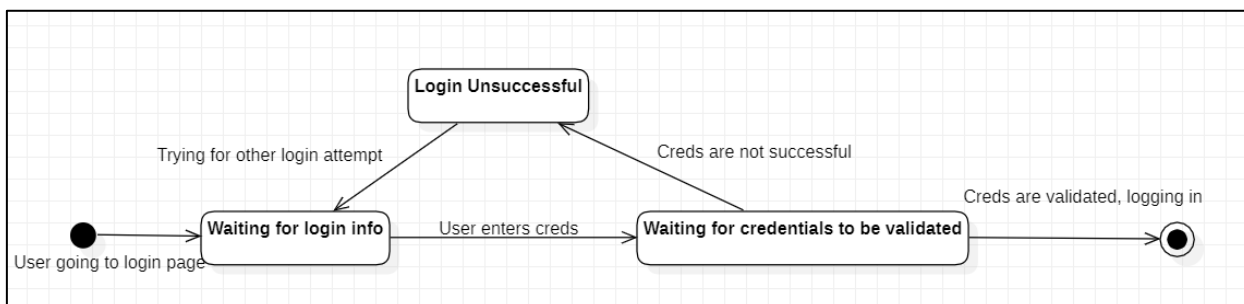


Рисунок 2.7 – Діаграма стану логіну в систему

З діаграмою стану логіну в систему ситуація зовсім інша. Вхідною точкою для неї є перехід користувача на сторінку логіну в систему, після чого система очікує ввід інформації від користувача для логіну в систему. Такою інформацією слугує електронна пошта та пароль. Після вводу інформації та відправки її на сервер для перевірки можливі два результати. Перший результат – логін успішний і користувача переносить в адмін-панель. Другий же – логін провалено, оскільки не вистачає інформації в одному з полів або в самій введеній інформації є помилка. Тоді цикл повторюється і користувачеві треба знову ввести інформацію про свій обліковий запис, але на цей раз вже з урахуванням помилки, яка виникла.

2.5 Діаграми діяльності

Діаграми діяльності використовуються для моделювання широкого діапазону систем і процесів, від простих бізнес-процесів до складних програмних систем [7]. Вони складаються з набору символів і позначень для представлення різних дій, таких як завдання, дії, рішення та цикли, і показують зв'язки та залежності між ними.

Створено три діаграми діяльності з наступним функціоналом:

1) пошук результатів проекту (рис. 2.8). Діаграма містить такі кроки як: введення пошукового значення, пошук значення, надсилання знайдених значень на клієнт.

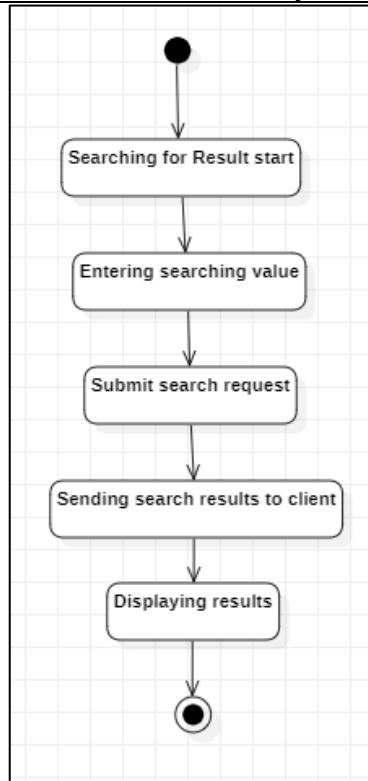


Рисунок 2.8 – Діаграма діяльності пошуку результатів проєкту

2) додавання учасника проєкту (рис. 2.9). Діаграма містить такі кроки як: введення інформації про учасника, публікація учасника, перевірка успішності публікації учасника тощо.

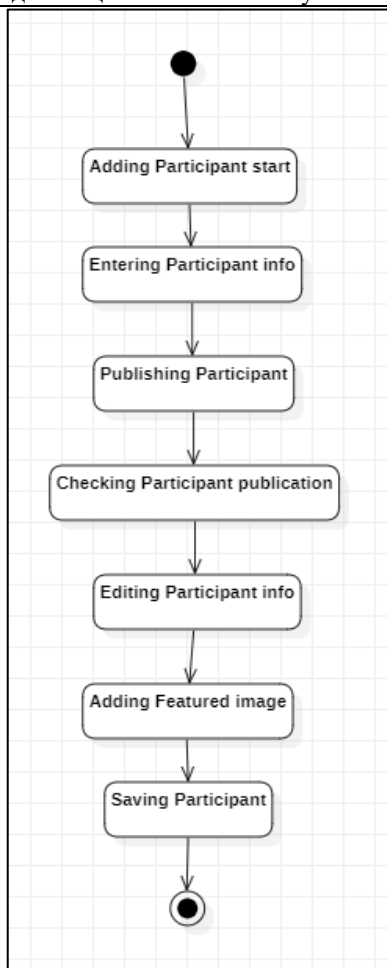


Рисунок 2.9 – Діаграма діяльності додавання учасника проєкту

3) логін у систему (рис. 2.10).

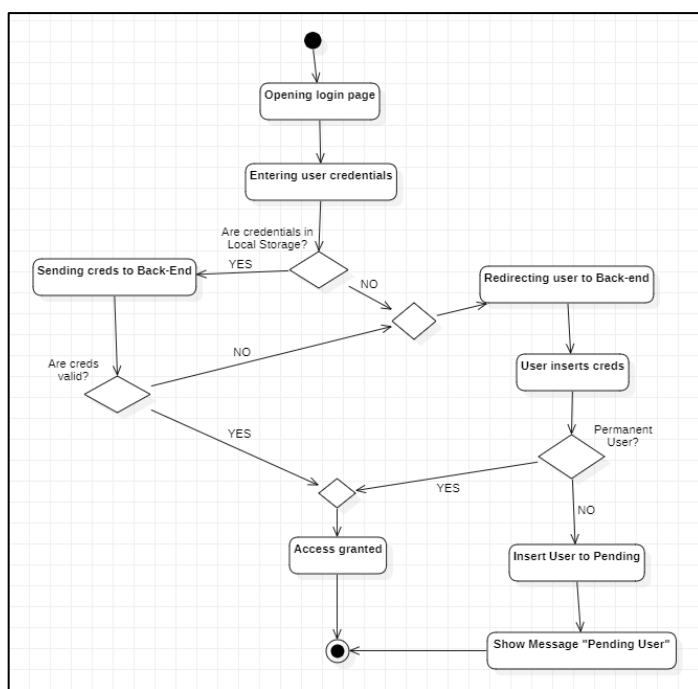


Рисунок 2.10 – Діаграма діяльності додавання пакетів

На даній діаграмі видно, що користувач вводить інформацію від свого облікового запису, після чого відбувається цикл подій та умов, які призводять до одного чи іншого результатів. Наприклад у разі, якщо інформація правильна – користувача переносить в адмін-панель. Але у разі помилки або відсутності інформації – його переносить назад на сторінку логіну з певною помилкою, на основі якої він має виправити введені дані, та спробувати залогінитись ще один раз.

Висновки до розділу 2

У другому розділі було описано процес моделювання ПЗ, що розробляється. Було створено діаграму класів, діаграму пакетів, діаграму розгортання, діаграму пакетів тощо. Для опису взаємодії користувача з ПЗ, було розроблено сценарії використання, основою яких стала діаграма використання.

3 ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ ТА ОГЛЯД СТЕКУ ЗАДІЯНИХ ТЕХНОЛОГІЙ

3.1 Розробка UML-діаграм

В більшості випадків під час моделювання та проєктування ПЗ використовуються діаграми. Серед найпоширеніших та найзручніших діаграм виділяються UML-діаграми. Вони надають змогу описати різноманітні бізнес-процеси, проєктування системи, а також дозволяють найбільш точно зобразити структуру програмного забезпечення.

UML – це спеціальна мова моделювання, яка надає стандартний набір графічних символів та правил для створення моделей програмних систем.

Однією з особливостей таких діаграм є їх варіативність. UML включає в себе такі типи діаграм, як: діаграми варіантів використання, діаграми класів, діаграми послідовностей, діаграми станів тощо. Кожен такий тип має власну специфікацію та використовується для моделювання конкретних аспектів системи.

Окрім цього, UML надає можливість розробнику створювати власні типи діаграм, використовуючи спеціальні механізми розширення. Завдяки цьому вдається адаптувати цю мову моделювання до специфічних потреб при розробці проєкту, розширюючи її функціональність.

Також перевагою використання даних діаграм є їх взаємозв'язок. Це організовано за допомогою можливості переходу від однієї діаграми до іншої, використовуючи представлені розробнику інструменти. Такий підхід дозволяє уникнути повторення інформації та детально описати функціональність системи, формулюючи її більш повну картину взаємодії.

Але певно найважливішою особливістю UML діаграм – є їх стандартизованість. Через це покращується комунікація між розробниками, оскільки вони вивчають одні стандарти. UML стабільно підтримується та з часом формулює нові стандарти для побудови діаграм, а тому важливо слідкувати за її оновленнями та вивчати нову інформацію.

Для визначення основних сутностей застосунку та зв'язків між ними використовується діаграма класів [4]. Створена діаграма має 11 основних та 6 допоміжних сутностей (рис. 3.1).

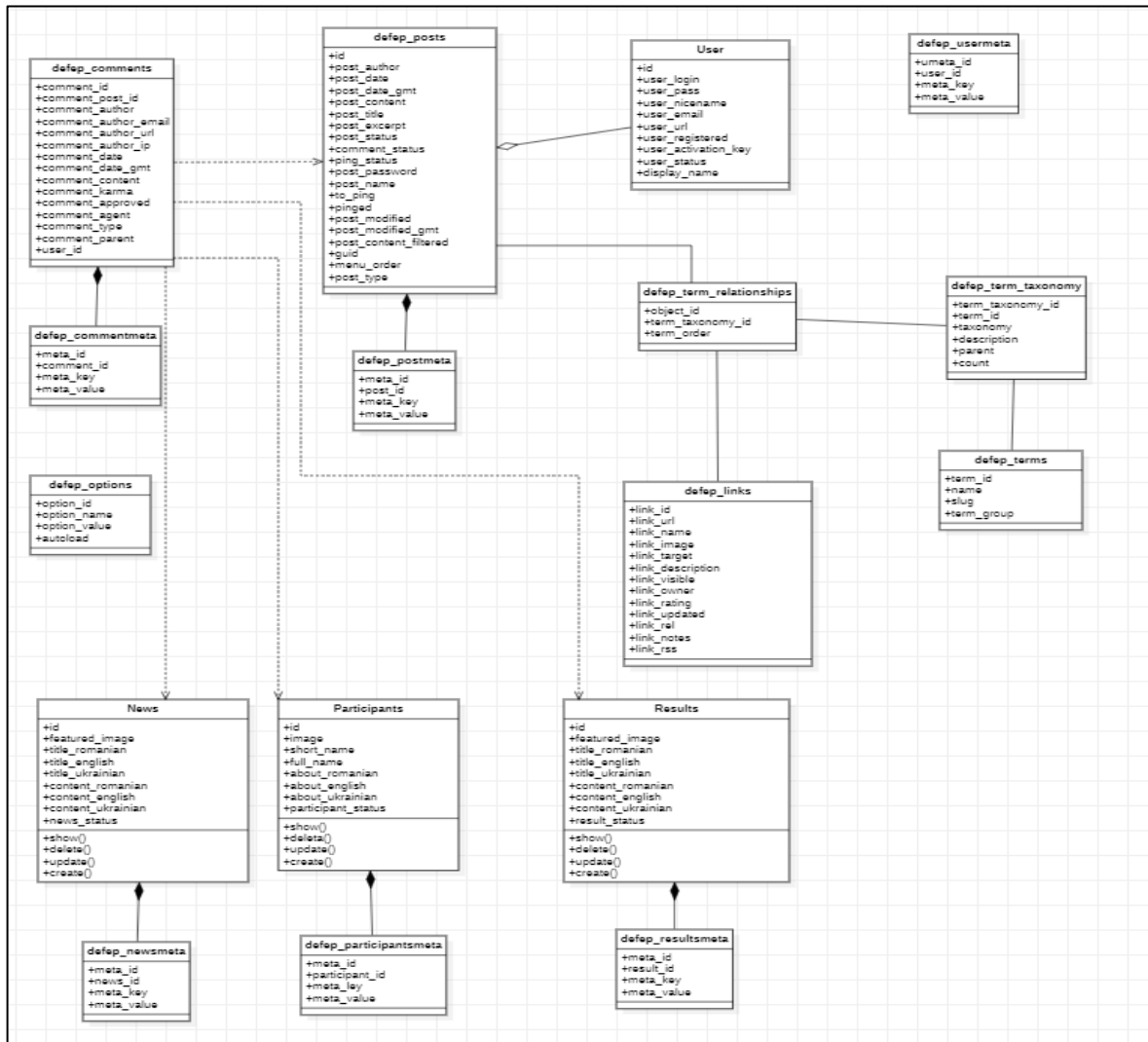


Рисунок 3.1 – Діаграма класів

Серед представлених класів методи мають лише ті, які є моделями для інформації сайту: новини, учасники та результати. Далі було створено діаграму компонентів

Діаграма компонентів відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись. Модуль програмного забезпечення може бути представлено як компоненту. Деякі компоненти існують під час компіляції, деякі — під час компонування, а деякі під час роботи програми [8].

Діаграма компонентів відображає лише структурні характеристики, для відображення окремих екземплярів компонент слід використовувати діаграму розгортання.

Діаграма компонентів серверної частини застосунку проєкту дистанційної освіти складається з 3 частин (рис. 3.2).

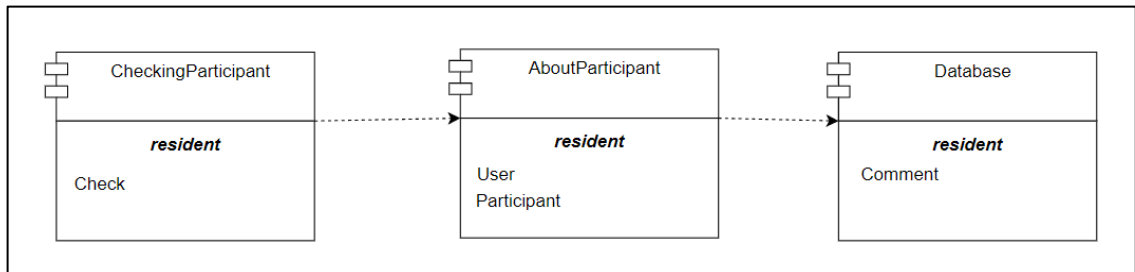


Рисунок 3.2 – Діаграма компонентів серверної частини застосунку

Діаграма розгортання (рис. 3.3) це діаграма, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах [9]. Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час роботи програми на таких діаграмах не відображаються; натомість, їх можна відобразити на діаграмах компонент. Діаграма розгортання відображає робочі екземпляри компонент, а діаграма компонент, натомість, відображає зв'язки між типами компонент.

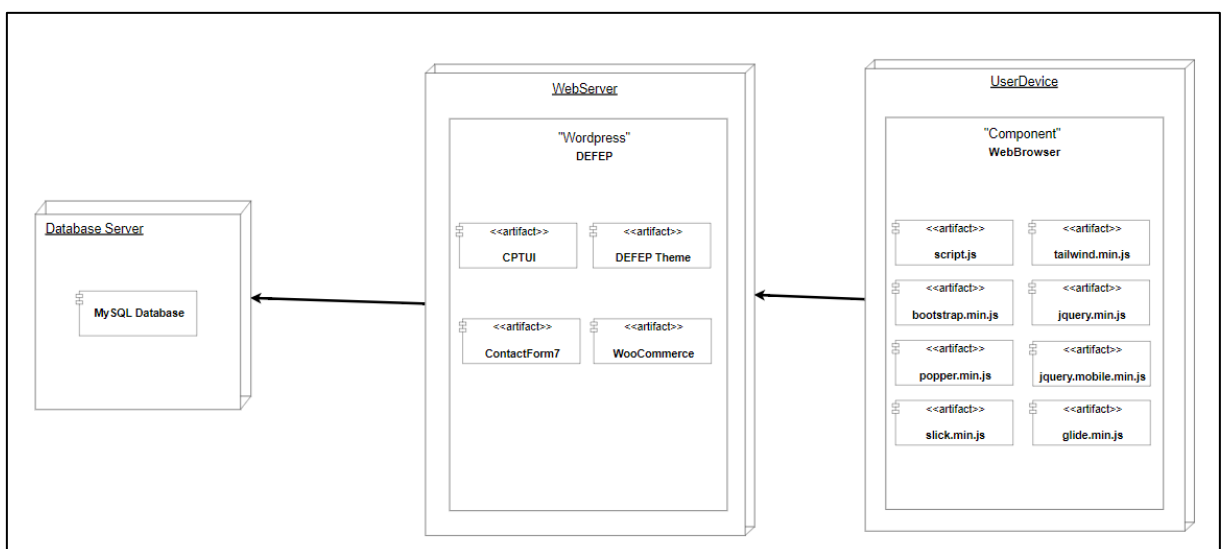


Рисунок 3.3 – Діаграма розгортання

В даному випадку діаграма розгортання має три елементи – сервер бази даних, вебсервер на якому зберігається та хоститься застосунок та пристрій користувача із застосунком для перегляду вебсайту, в даному випадку – браузер. Кожен з цих елементів має певні компоненти, наприклад на сервері бази даних зберігається сама база даних з усією інформацією, присутньою на сайті і у разі необхідності – застосунок вебсайту звертається до бази для отримання необхідної інформації, зазвичай це відбувається після запиту зі сторони користувача.

Було створено діаграму пакетів. Діаграми пакетів відображають залежності між пакетами, які складають модель [10].

Діаграми пакетів можуть використовувати пакети, які містять прецеденти для ілюстрації функціональності програмного забезпечення системи.

Діаграма пакетів застосунку проєкту дистанційної освіти складається з 3 основних пакетів (рис. 3.4).

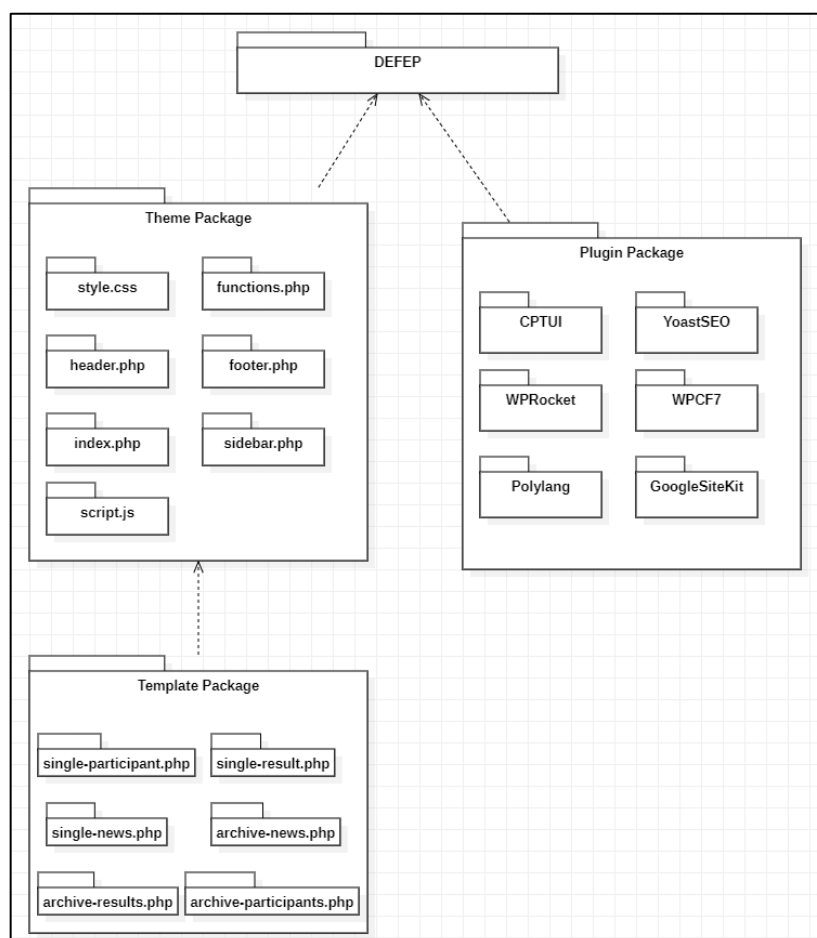


Рисунок 3.4 – Діаграма пакетів

Серед пакетів застосунку – пакет з темою, пакет з шаблонами та пакет з плагінами.

Пакет з темою відповідає за наявність файлів, які впливають на зовнішній вигляд та роботу вебсайту, включаючи в себе файл стилів, файл з функціями, хедер, футер тощо. Від нього походить пакет із шаблонами, який впливає на дизайн та структуру сторінок самого застосунку.

Також окремо виділяється пакет з плагінами. Він включає в себе розширення та їх файли, які збільшують функціонал вебсайту, а також надають спеціалізовані інструменти для розробників для розширення можливостей в розробці самого застосунку.

3.2 Огляд технологій

Для розробки вебзастосунку проєкту розвитку дистанційної освіти майбутнього було використано наступні технології:

- HTML;
- JS;
- CSS;
- Bootstrap;
- jQuery;
- Wordpress;
- MySQL.

Обрання технологій для розробки застосунку є есенціальною складовою успішної побудови проєкту, оскільки цей вибір має великий вплив на подальшу структуру та зручність застосунку.

Обрані технології надають необхідні інструменти та можливості для вирішення поставлених задач перед розробником, тому ці технології підбираються в залежності від потреб, розмірів, а також необхідної функціональності проєкту.

Наприклад якщо проєкт не є занадто великим, то можна обрати стек технологій, який може в подальшому бути повільним в роботі, але через невеликий масштабу застосунку – ця повільність буде нівельована. І навпаки,

якщо проєкт масштабний – то буде краще обрати технології, які хоч і важкі в своїй будові, але дають найкращу швидкість роботи та дозволяють в подальшому комфортно впроваджувати новий функціонал.

Якщо ж брати архітектуру застосунку, то тут можна діяти за таким же принципом. Використовувати легкі фреймворки або мови програмування коли проєкт не є таким масштабним, оскільки архітектура буде простою. Та навпаки, використовувати тяжчі технології, якщо архітектура застосунку планується бути масштабною.

І останнім чинником, який впливає на обрання стеку технологій – це час розробки проєкту. Якщо часу видається мало, то тоді можна використати технології, які буквально працюють в два кліки і розробник їх буде розуміти на сто відсотків, хоч і жертвуючи певним функціоналом та, можливо, швидкодією. І якщо часу вдосталь – то краще обрати комплексні рішення, які хоч і не будуть такими ж простими в засвоєнні та використанні, але нададуть розробнику максимум можливих інструментів для розробки.

Тому звідси видно, що обрання певного стеку технологій є есенціальним при розробці вебсайту, оскільки це впливає на можливості розробника, швидкодію вебсайту та можливість в подальшому розширювати та додавати нові інструменти для роботи з застосунком.

Серед основних мов програмування було вирішено обрати PHP та JavaScript.

PHP (Hypertext Preprocessor) було обрано в якості основної мови програмування для бекенду. Головною перевагою цієї мови є її простота, а також функціонал, який поступово розширюється з існуючою програмою підтримки даної мови та виводом її в open source (рис. 3.5). Це дозволяє майже кожен рік представляти розробникам нові інструменти для роботи, а також новий функціонал.



Рисунок 3.5 – Логотип PHP

Великою перевагою даної мови також є велика кількість існуючих для неї фреймворків та бібліотек, що в десятки разів збільшує її можливості. Більшість фреймворків PHP мають гарно прописану документацію, що дозволяє розробникам швидко опанувати нові технології та вводити їх в роботу. Перевагою мови є також і те, що вона одна з найпоширеніших мов програмування у світі. За статистикою близько 70% вебсайтів існуючих в мережі Інтернет написані на цій мові, а це означає, що якщо не на всіх, то на більшості хостинг-платформах є її підтримка.

Але певно все ж найбільшою її перевагою, є великий обсяг інструментів для роботи з різноманітними базами даних, наприклад MySQL, що дозволяє розробнику ефективно працювати з даними (рис. 3.6).

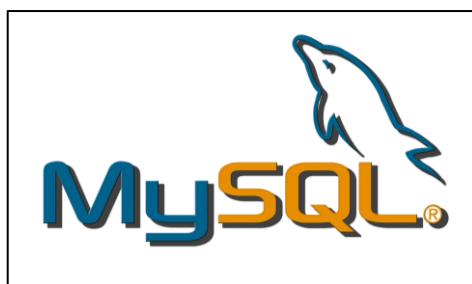


Рисунок 3.6 – Логотип MySQL

Також дана мова була обрана специфічно через CMS Wordpress. Це система контентного менеджменту, яку можна розглядати окремо як фреймворк, оскільки вона надає величезну кількість інструментів для розробки проєктів.

Для фронтенду основною мовою програмування було вирішено обрати Javascript, оскільки вона є однією з найпопулярніших мов програмування у сегменті веброзробки (рис. 3.7).



Рисунок 3.7 – Логотип JavaScript

Ключовою перевагою JavaScript є гнучкість її використання, що дозволяє створювати вебсторінки, мобільні додатки та в деяких випадках навіть десктопні застосунки. Це універсальний інструмент розробки, який підтримується багатьма браузерами та дозволяє працювати з ним прямо в браузерах, використовуючи інструменти розробників.

Дана мова так само, як і PHP має величезно кількість різноманітних фреймворкі та бібліотек, кількість яких з кожним днем збільшується, що дозволяє використовувати JS в різних програмних рішеннях.

Основною ціллю використання даної мови є взаємодія з користувачем на клієнтському рівні. Мова дозволяє створювати інтерактивні події, маніпулювати різними елементами сторінок вебзастосунку, а також відправляти дані на сервер для роботи.

В цілому JavaScript це велика в своїх сферах застосування мова, яка дозволяє пов'язати роботу клієнтської та серверної частини, що робить її прекрасним інструментом для розробки вебзастосунків.

В якості основної технології для розробки застосунку було обрано CMS систему WordPress (рис. 3.8).



Рисунок 3.8 – Логотип WordPress

WordPress є сьогодні однією з найпопулярніших технологій для створення вебсайтів і управління контентом і існує на ринку технологій розробки програмного забезпечення вже 20 років.

Дана технологія має велику кількість переваг. Серед них можна виділити, наприклад, простоту використання, оскільки WP має інтуїтивно зрозумілий і дружній для користувача інтерфейс, а встановлення та налаштування сайту на його основі займає мінімальну кількість часу і зусиль, оскільки все робиться лише в декілька кліків мишею.

Серед переваг також можна виділити досить велику гнучкість технології, оскільки у розробника є величезний вибір з плагінів, які розроблені для цієї системи і які встановлюються в декілька натискань. Ще до більшості плагінів розроблена досить хороша документація, яка при бажанні розробника дозволяє розширити його інструменти до великих масштабів.

Також не варто забувати, що окрім існування плагінів, розробник також має можливість власного налаштування цієї системи, використовуючи код, написаний на мові PHP, який дозволяє розширювати можливості Wordpress до лімітів, які можливі на мові PHP.

У даної технології є велика спільнота користувачів, як рядових, так і розробників з величезним досвідом. Існують спеціалізовані форуми, на яких у разі необхідності можна задати питання і отримати як пряму відповідь, так і безліч порад від досвідчених розробників. Також через існування такої великої спільноти, окрім плагінів існує велика кількість готових тем для вебсайтів,

кожна з яких має свій неповторний дизайн та можливості, як наприклад покращений редактор сторінок та вбудовану адаптивність для мобільної версії вебсайту.

І найголовніша перевага даної CMS – це те, що вона безкоштовна і для неї регулярно виходять оновлення, які приносять новий функціонал. Але якщо все ж виникає бажання заплатити за роботу розробників цієї технології – завжди можна зробити благодійне пожертвування на подальший розвиток системи.

Серед допоміжних технологій можна також виділити Bootstrap – CSS фреймворк, який надає велику кількість інструментів роботи з HTML елементами та який спрощує розробку динамічних вебсайтів та вебзастосунків (рис. 3.9).



Рисунок 3.9 – Логотип Bootstrap

Серед фреймворків для JS можна виділити jQuery. Це одна з найпопулярніших бібліотек JavaScript, яка спрощує роботу з маніпулюванням HTML-документів, обробкою подій, анімацією, взаємодією з сервером та іншими завданнями. Через свої особливості та інструментал це робить її досить привабливим серед розробників (рис. 3.10).



Рисунок 3.10 – Логотип jQuery

Найбільшою перевагою даної бібліотеки є його простота використання.

Вона надає зручний синтаксис, який дозволяє легко обирати елементи сторінки, змінювати їх властивості, оброблювати події тощо. Самі ж команди є короткими та зрозумілими, що набагато полегшує розробку та кардинально зберігає час. Перевагою також є і хороша кросс-браузерна підтримка. Бібліотека надає можливість створювати вебзастосунки, які працюють без проблем на різних браузерах, без потреб вирішувати проблеми сумісності, які можуть раптом виникнути.

Висновки до розділу 3

У трьому розділі кваліфікаційної роботи бакалавра було спроектовано вебзастосунок розвитку дистанційної освіти майбутнього, використовуючи UML-діаграми, серед яких: діаграма класів, діаграма компонентів, діаграма розгортання, діаграма пакетів. Це дозволяє візуалізувати роботу та архітектуру системи, що розробляється, що в подальшому дозволяє поліпшити розробку програмного продукту, а також імплементацію нового функціоналу в систему.

Цей етап є одним із найважливіших під час розробки вебзастосунку, через те, що вся необхідна інформація візуалізовується у зручному вигляді та до неї можна в будь-який час в майбутньому звернутись за підказкою в подальшій розробці, допомагаючи розробникам як в часі, так і в розумінні архітектури.

Перед реалізацією вебзастосунку також було проведено порівняння деяких мов програмування і на основі їх переваг і недоліків було обрано необхідний стек технологій для роботи з проєктом. Таке порівняння дозволяє обрати найраці інструменти для реалізації програмного продукту, спираючись на їх функціонал, існуючі фреймворки для можливого розширення системи, а також обмеження, які має враховувати розробник, що в подальшому дозволяє зробити розробку більш швидкою та ефективною.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ ТА ПРОВЕДЕННЯ ТЕСТУВАННЯ

Вебзастосунок проєкту розвитку дистанційної освіти майбутнього містить в собі. У вебзастосунку є 3 типи користувачів: користувач, редактор та адміністратор. За необхідності адміністратор може додати нові типи користувачів, залежно від їх ролі в системі.

4.1 Огляд дизайну вебзастосунку

Для створення дизайну або “mock-ups” застосунку було використано спеціальний програмний застосунок «Figma».

Figma – це популярний інструмент для дизайну та прототипування, який дозволяє командам спільно працювати над вебдизайном і макетами. Він пропонує низку особливостей, які роблять його потужним інструментом для створення вебпродуктів.

Завдяки використанню хмарної бази даних та можливості спільної роботи в реальному часі, користувачі мають можливість змінювати, коментувати та спілкуватись один з одним під час спільної роботи. Особливо це корисно у випадку віддаленої роботи, коли розробники та дизайнери знаходяться в різних куточках світу і не мають можливості зустрічі в реальному світі.

На початку було реалізовано зовнішній вигляд головної сторінки (рис. 4.1).



Рисунок 4.1 – Дизайн головної сторінки

Оскільки лише адміністратор застосунку має права на створення новий користувачів, то реєстрації в застосунку не передбачено. Для логіну в системі використовується спеціальне URL-посилання (рис. 4.2). Під час його використання юзера переносить на стандартну сторінку логіну Wordpress.

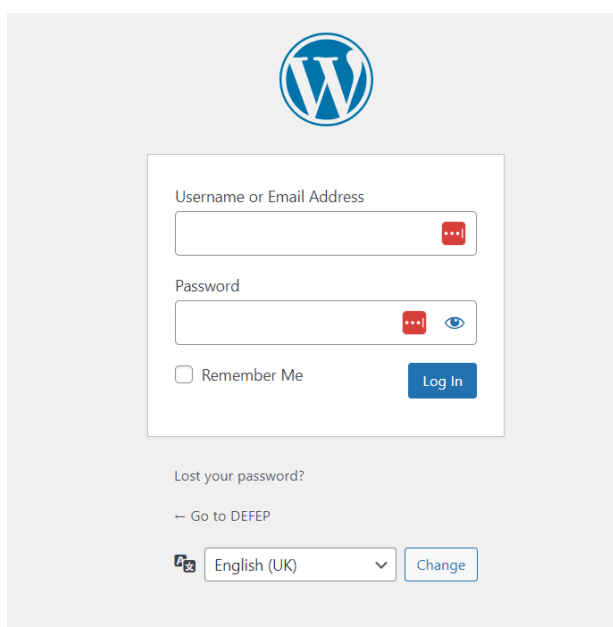


Рисунок 4.2 – Сторінка логіну

Також було створено дизайн подвійного хедеру застосунку, який використовується для навігації по сайту (рис 4.3). Хедер було вирішено зробити подвійним оскільки в ньому повинно знаходитись багато інформації.



Рисунок 4.3 – Дизайн хедеру

Було створено дизайн футеру вебсайту, в якому знаходяться посилання на соціальні мережі проєкту, інформація про керівника проєкту, а також інформація про авторське право (рис. 4.4). Для зручності користування сайтом також було додано кнопку “To the top”, яка дозволяє користувачу миттєво проскролити на початок сторінки вебзастосунку. Футер так само як і хедер є подвійним для вміщення всієї необхідної інформації.

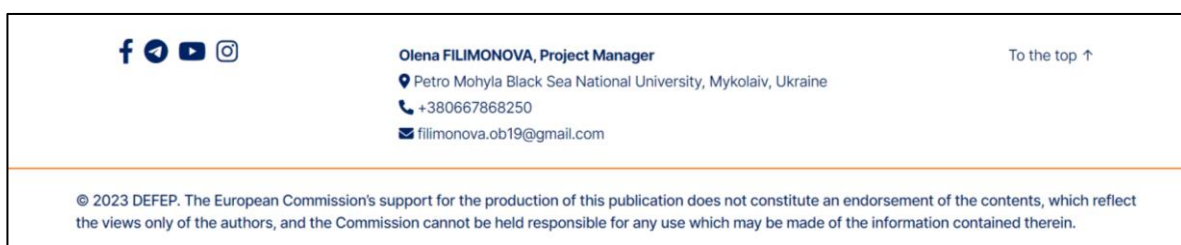


Рисунок 4.4 – Дизайн футеру

Після цього було створено та в подальшому реалізовано “mock-up” сторінки новин, результатів та активностей. На сторінці новин знаходяться невеликі блоки, кожен з яких уособлює додану на вебсайт новину (рис. 4.5).



Рисунок 4.5 – Дизайн сторінки новин

Для сторінки звітів було вирішено зробити аналогічний дизайн, оскільки вона має схожий зміст зі сторінкою новин (рис. 4.6).

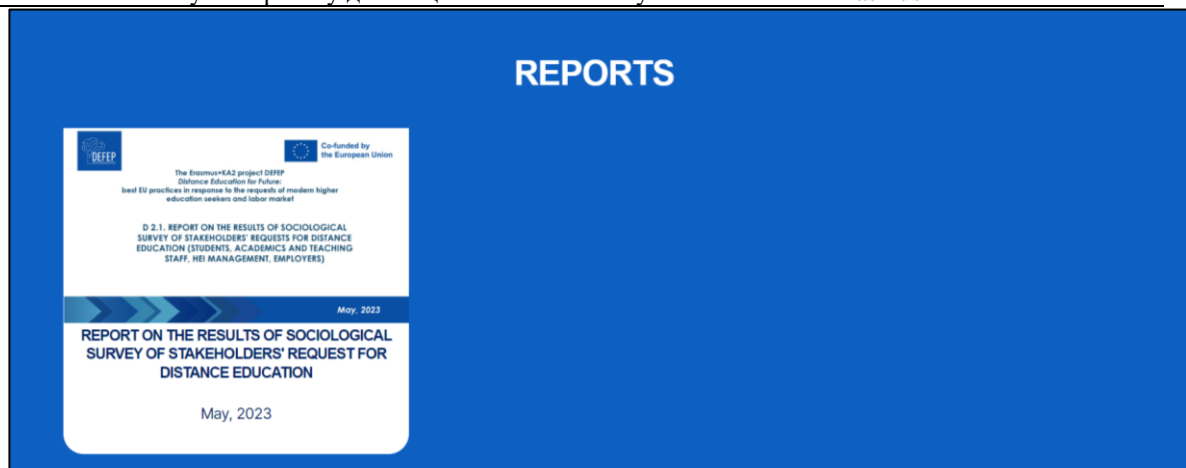


Рисунок 4.6 – Дизайн сторінки звітів

Також було реалізовано адаптивний інтерфейс вебзастосунку. Адаптивний підхід до розробки застосунку дозволяє забезпечити коректне відображення застосунку на різних пристроях та роздільних здатностях екранів, що забезпечує комфортність використання застосунку незалежно від пристрою.

Реалізація адаптивного інтерфейсу є важливою складовою розробки застосунку, на те є декілька причин. Перша причина – це велика різноманітність пристроїв для перегляду вебконтенту, серед них смартфони, комп'ютерні монітори, планшети, телевізори тощо. Через це потрібно, щоб верстка застосунку була пристосована до різної роздільної здатності, автоматично використовуючи ті чи інші правила для вірного розміщення елементів вебзастосунку.

Друга ж причина для цього – це SEO-оптимізація. Це вид оптимізації, який використовується для покращення рейтингу вебсайту в різних пошукових системах. Важливою вона є через те, що в таких рейтингах часто враховується адаптивність для використання на мобільних пристроях. Це дозволяє вебзастосунку з'являтися на вищих позиціях під час пошуку в пошукових системах.

Для реалізації адаптивного інтерфейсу було вирішено частину хедеру лишити в верхній частині екрану, а частину винести в спеціально створене бокове меню вебзастосунку. На рис. 4.7 представлено стандартний вигляд сайту на версії для персональних комп'ютерів.



Рисунок 4.7 – Стандартний вигляд сайту

Після зменшення роздільної здатності екрану, хедер кардинально змінюється, з'являється кнопка для відкриття бокового меню. Серед елементів на місці лишаються лише логотипи, а також номер програми Erasmus+KA2, за якою зроблений вебзастосунок (рис. 4.8).



Рисунок 4.8 – Вигляд на мобільних пристроях

Для того щоб відкрити бокове меню, достатньо натиснути на кнопку, що з'явилась при зменшенні роздільної здатності екрану. В меню знаходяться всі важливі посилання, які знаходяться в хедері комп'ютерної версії, а також перемикач мов вебсайту (рис. 4.9).

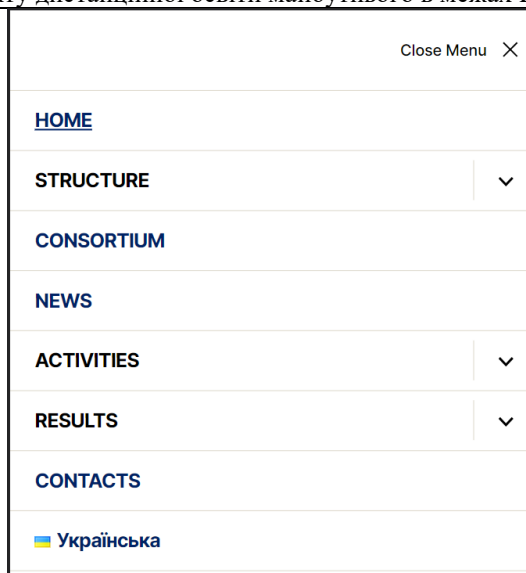


Рисунок 4.9 – Бокове меню

Отже як можна побачити, реалізація адаптивності дозволяє вплинути на user experience користувача позитивно.

4.2 Розробка функціоналу адмін-панелі

Великою перевагою CMS системи Wordpress є можливість повного кастомного налаштування адмін-панелі, що дозволяє в подальшому навіть користувачам, які не знайомі зі сферою розробки програмного забезпечення користуватись нею та працювати з контентом. Така кастомізація передбачена як роботою із кодом напряму в системі, так і можливістю встановлення різноманітних плагінів, які часом можуть розширити функціонал системи не гірше, ніж різноманітні фреймворки.

Оскільки в вебзастосунку використовуються різні типи постів (новини, активності, результати), було вирішено створено кастомні типи постів, використовуючи мову програмування PHP та функціонал самої системи. Нижче наведено уривки коду створення кожного з таких типів постів (рис. 4.10, 4.11, 4.12).


```
function cptui_register_my_cpts() {  
  
    /**  
     * Post Type: Results.  
     */  
  
    $labels = [  
        "name" => esc_html__( "Results", "twentytwenty" ),  
        "singular_name" => esc_html__( "Result", "twentytwenty" ),  
        "menu_name" => esc_html__( "Project Results", "twentytwenty" ),  
        "all_items" => esc_html__( "All Results", "twentytwenty" ),  
        "add_new" => esc_html__( "Add new", "twentytwenty" ),  
        "add_new_item" => esc_html__( "Add new Result", "twentytwenty" ),  
        "edit_item" => esc_html__( "Edit Result", "twentytwenty" ),  
        "new_item" => esc_html__( "New Result", "twentytwenty" ),  
        "view_item" => esc_html__( "View Result", "twentytwenty" ),  
        "view_items" => esc_html__( "View Results", "twentytwenty" ),  
        "search_items" => esc_html__( "Search Results", "twentytwenty" ),  
        "not_found" => esc_html__( "No Results found", "twentytwenty" ),  
        "not_found_in_trash" => esc_html__( "No Results found in trash", "twentytwenty" ),  
        "parent" => esc_html__( "Parent Result:", "twentytwenty" ),  
    ];  
}
```

Рисунок 4.10 – Код створення типу постів «Результат»

```
$args = [  
    "label" => esc_html__( "Activities", "twentytwenty" ),  
    "labels" => $labels,  
    "description" => "",  
    "public" => true,  
    "publicly_queryable" => true,  
    "show_ui" => true,  
    "show_in_rest" => true,  
    "rest_base" => "",  
    "rest_controller_class" => "WP_REST_Posts_Controller",  
    "rest_namespace" => "wp/v2",  
    "has_archive" => false,  
    "show_in_menu" => true,  
];
```

Рисунок 4.11 – Код створення типу постів «Активності»

```
    "rest_controller_class" => "WP_REST_Posts_Controller",  
    "rest_namespace" => "wp/v2",  
    "has_archive" => false,  
    "show_in_menu" => true,  
    "show_in_nav_menus" => true,  
    "delete_with_user" => false,  
    "exclude_from_search" => false,  
    "capability_type" => "post",  
    "map_meta_cap" => true,  
    "hierarchical" => false,  
    "can_export" => false,  
    "rewrite" => [ "slug" => "news", "with_front" => true ],  
    "query_var" => true,  
    "menu_icon" => "dashicons-media-text",  
    "supports" => [ "title", "editor", "thumbnail" ],  
    "show_in_graphql" => false,  
];  
  
register_post_type( "news", $args );  
  
add_action( 'init', 'cptui_register_my_cpts' );
```

Рисунок 4.12 – Код створення типу постів «Новини»

Після написання даного коду в адмін-панелі з'явилися нові типи постів (рис. 4.13).

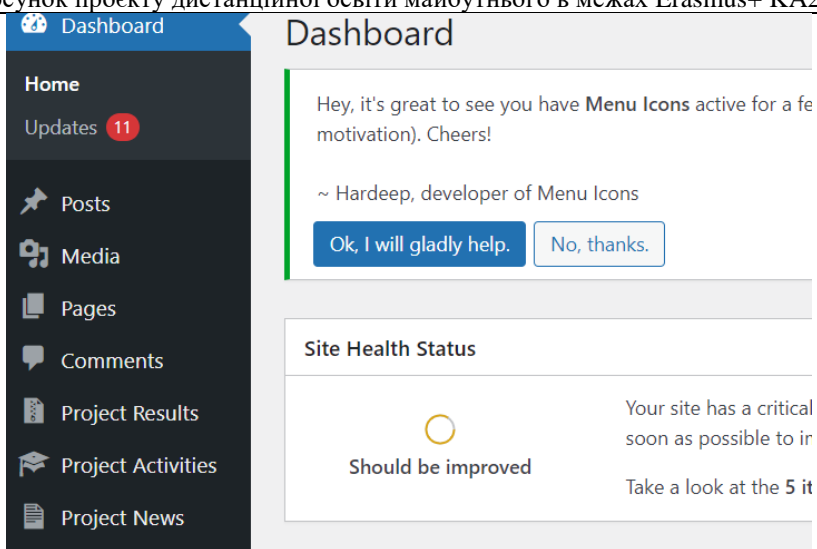


Рисунок 4.13 – Нові типи постів в адмін-панелі

Оскільки деякі типи постів (результати та активності) мають свої підкатегорії, були створені кастомні таксономії для них (рис. 4.14, 4.15).

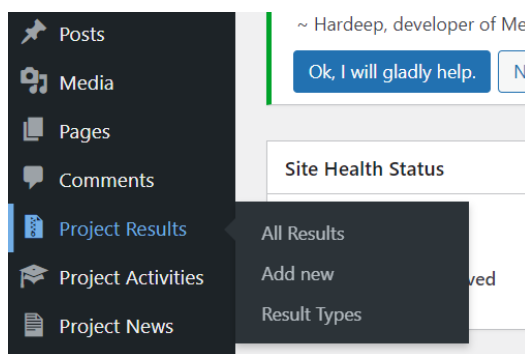


Рисунок 4.14 – Нові категорії типу посту «Результати»

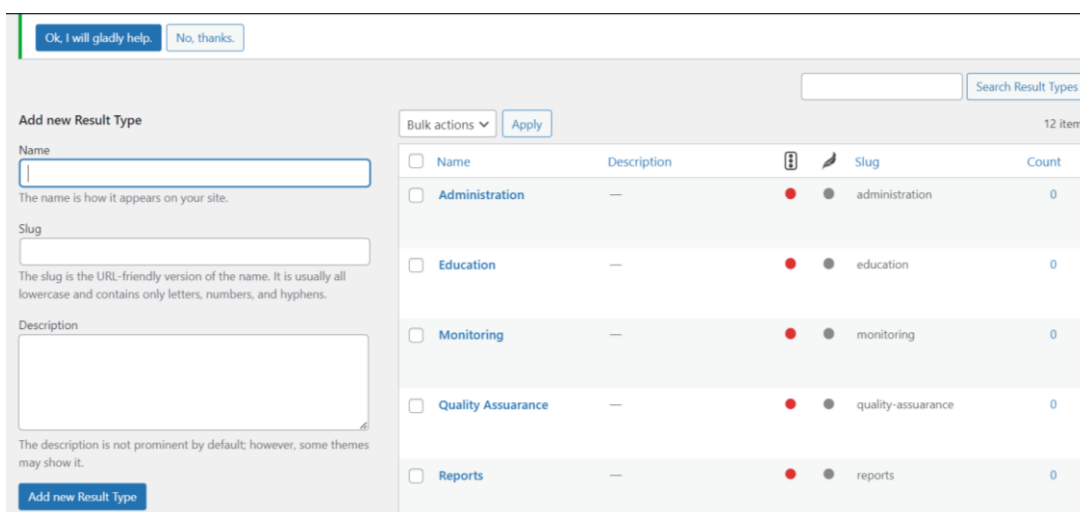


Рисунок 4.15 – Сторінка додавання категорій

Після написання всього необхідного коду, через адмін-панель з'явилась можливість додавати пости необхідних типів, а також з'явилась можливість прив'язки цих постів до певних категорій.

4.3 Додавання мультимовності на сайт

Одним з основних завдань роботи, було створення можливості перемикатись між декількома версіями сайту, які розроблені для тої чи іншої мови.

Для реалізації такої можливості для користувачів було обрано плагін під назвою «Polylang». Даний плагін дозволяє створювати різні версії для сторінок, в залежності від необхідної мови.

Алгоритм його роботи наступний – спочатку створюється звичайна сторінка через адмін-панель вебсайту, далі адміністратор чи редактор переходить в перегляд всіх створених сторінок, які присутні на вебсайті. В полях створених сторінок знаходяться кнопки додавання перекладів для різних мов (рис. 4.16).

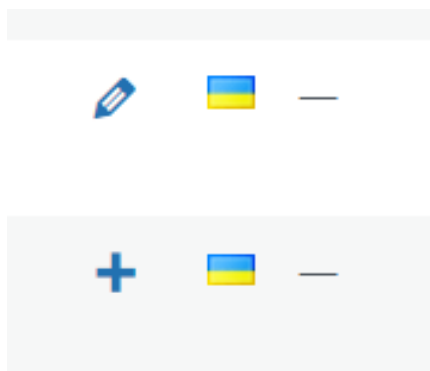


Рисунок 4.16 – Кнопки додавання перекладу сторінки та редагування перекладу

Після натискання на додавання нового перекладу, адміністратор опиняється в стандартному редакторі сторінок Wordpress. Там він додає нові блоки, на основі перекладу та в кінці зберігає і публікує переклад.

Після опублікування перекладу, через редактор меню можна додати спеціальний пункт для перемикання мов (рис. 4.17).

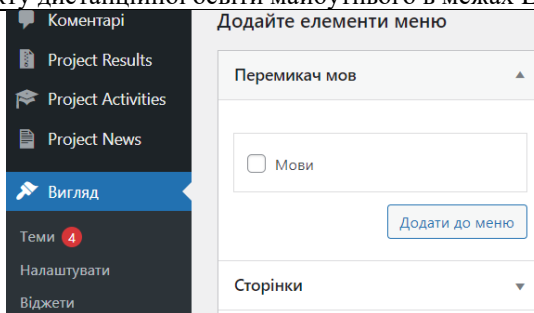


Рисунок 4.17 – Перемикач мов

Після додавання перемикача мов, в хедері вебсайту з'явиться новий елемент. Його зовнішній вигляд можна змінювати завдяки налаштування в самому плагіні. В даному випадку перемикач має вигляд прапорів країн, в якій використовуються мови (рис. 4.18).



Рисунок 4.18 – Перемикач мов біля вкладки «Contacts»

Використання плагіну в даному випадку набагато спростило розробку, надаючи в розробці необхідні інструменти для перекладу сторінок вебсайту. Завдяки цьому не довелося розробляти код для роботи таких перекладів – достатньо було скористатись вже готовими програмними рішеннями, що призвело до покращення умов та швидкості розробки ПЗ.

4.4 SEO-оптимізація

SEO-оптимізація для вебзастосунків є дуже важливою складовою, оскільки організує появи сторінок вебсайту в пошукових системах, а також покращує їх позицію в результатах пошуку.

Для даної оптимізації було обрано плагін під назвою «YoastSEO». Цей плагін надає спеціальні інструменти для покращення видимості та позицій сторінок вебсайту та самого сайту в пошукових системах. Він дозволяє додавати ключові слова та фрази для кожної сторінки або посту, після чого автоматично здійснює аналіз контенту, доданих ключових слів та видає звіт з порадами для покращення оптимізації.

Для початку було налаштовано плагін в меню його налаштувань (рис. 4.19)

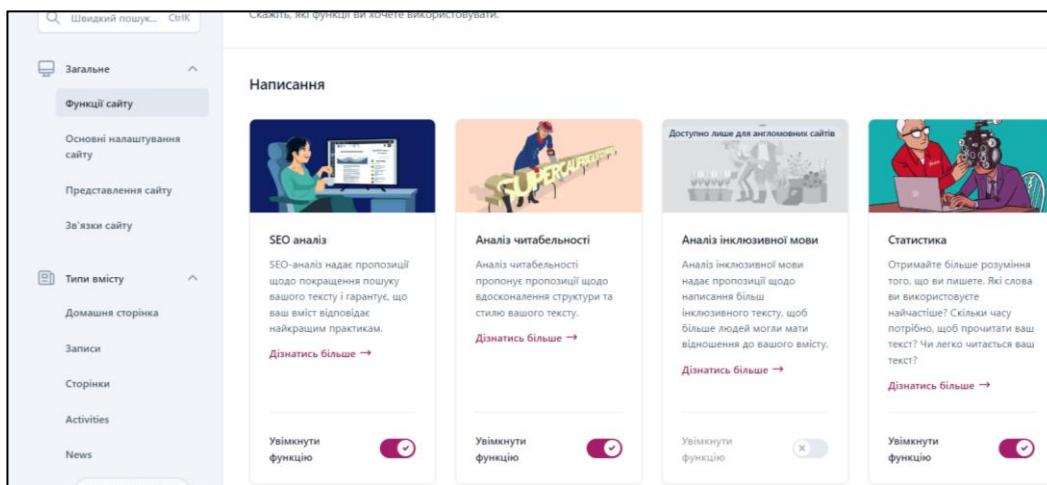


Рисунок 4.19 – Меню налаштування плагіну

В налаштуваннях було увімкнено більшість можливостей плагіну, окрім аналізу інклюзивної мови, оскільки в даному випадку вона не є важливою.

Також окремо було увімкнено появу різних типів постів в пошукових результатах (рис. 4.20).

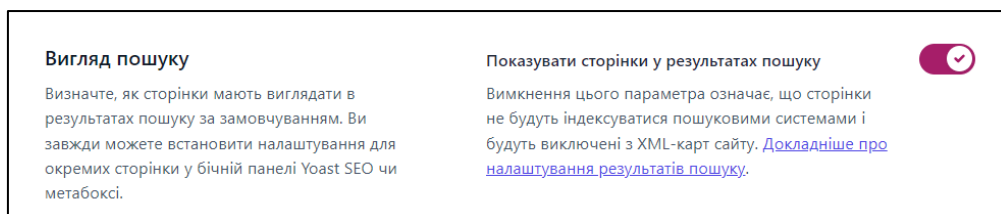


Рисунок 4.20 – Опція для появи сторінок у результатах пошуку

Аналогічно сторінкам, така опція була активована і для всіх інших типів постів на сайті (активності, результати, новини тощо).

Після цього було налаштовано ключові слова сторінок вебсайту, використовуючи спеціальний блок, який з'явився в редакторі сторінок після встановлення плагіну (рис. 4.21).

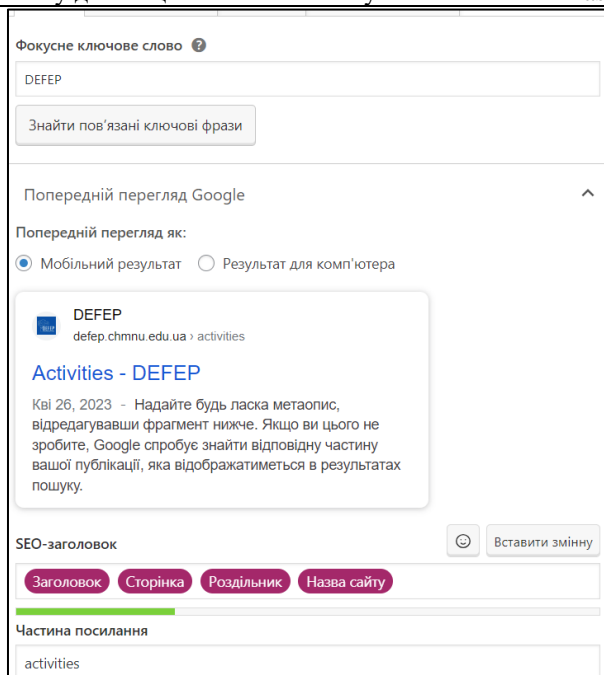


Рисунок 4.21 – Блок редагування ключових слів

Після пророблених дій зі сторінками, вони будуть з'являтися у пошуковій системі частіше, а також будуть займати більш високі позиції. Це дозволить збільшити охоплення аудиторії, які будуть зацікавлені в даному проєкті.

4.4 Тестування

Після всіх пройдених етапів розробки програмного проєкту залишається найважливіший – тестування. Найважливіший він, бо дозволяє віднайти всі помилки, баги в дизайні та проблеми з застосунком. Серед таких проблем можуть бути як недоліки у функціональності застосунку, можливі проблеми при роботі не тільки адміністратора, а й користувача з застосунком. Наприклад адміністратор не може додати нового користувача в систему та переназначити його ролі або редактор не може додати чи видалити різні пости в системі. Якщо ж говорити за користувача, то це можуть бути різні проблеми з коректністю відображення застосунку на різних платформах та девайсах.

Тестування також дозволяє розробникам знайти нові можливості для поліпшення швидкодії застосунку та оптимізації певних частин коду. В наступних таблицях наведені проведення тестування за різних сценаріїв роботи застосунку.

Таблиця 4.1 – Створення нового користувача

Діючі особи	Адміністратор
Мета	Створення нового користувача у системі
Передумова	Користувача не існує в системі
<p>Успішні сценарії:</p> <ul style="list-style-type: none"> – Адміністратор переходить до вкладки користувачів на натискає на кнопку «Додати користувача». – Адміністратор заповнює всі необхідні поля нового користувача (юзернейм, електронну пошту, пароль, ім'я користувача тощо). – Адміністратор натискає на кнопку «Додати нового користувача». – Система оброблює новий обліковий запис. – Система зберігає обліковий запис. – Система повертає адміністратора на сторінку користувачів з написом про усіх операції. – Адміністратор використовуючи логін і пароль намагається увійти в створений обліковий запис. 	
<p>Сценарій успішний. Збережено новий обліковий запис. Адміністратор успішно заходить в створений обліковий запис користувача.</p>	
<p>Розширенні сценарії:</p> <ul style="list-style-type: none"> – Адміністратор вводить не всі необхідні дані. Результат: система не створює акаунт, система виводить повідомлення про помилку. – Адміністратор вводить неправильну електронну пошту. Результат: система не створює акаунт, система виводить повідомлення про помилку – Адміністратор намагається створити вже існуючого користувача в системі. Результат: система не створює акаунт, система виводить повідомлення про помилку. 	
<p>Усі сценарії розширення успішно виконані.</p>	

Таблиця 4.2 – Вхід в обліковий запис

Діючі особи	Користувач
Мета	Вхід в обліковий запис
Передумова	Користувач не авторизований у системі
<p>Успішні сценарії:</p> <ul style="list-style-type: none"> – Користувач переходить до форми логіну за спеціальним посиланням. – Користувач заповнює необхідні дані (логін, пароль). – Користувач натискає кнопку «Увійти». – Система обробляє дані, надіслані користувачем. – Система переводить користувача в адмін-панель. 	
<p>Розширені сценарії:</p> <ul style="list-style-type: none"> – Користувач вводить не всі необхідні дані. Результат: Авторизація не відбулась, система видає повідомлення про помилку. – Користувач вводить неправильний логін або пароль. Результат: Авторизація ну відбулась, система виводить повідомлення про помилку. 	

Таблиця 4.3 – Створення новини

Діючі особи	Адміністратор, редактор
Мета	Створення новини
Передумова	Новини не існує в системі, адміністратор або редактор авторизовані в системі.
<p>Успішні сценарії:</p> <ul style="list-style-type: none"> – Адміністратор переходить в адмін-панель. – Адміністратор переходить у розділ новин в адмін-панелі. – Адміністратор натискає на кнопку «Додати новину». – Адміністратор переходить на сторінку додавання новини, де вводить необхідні інформацію про новину. – Адміністратор натискає на кнопку «Опублікувати новину». – Система оброблює запит. – Система видає повідомлення про успішність додавання новини. 	

Кінець таблиці 4.3

Сценарій успішний. Новина додана.
Розширенні сценарії: <ul style="list-style-type: none">– Адміністратор не вводить назву новини. Результат: Новина не додана, система виводить повідомлення про помилку.– Адміністратор задає URL-адресу вже існуючої новини. Результат: Новина не додана, система виводить повідомлення про помилку.

Висновки до розділу 4

У четвертому розділі кваліфікаційної роботи бакалавра було розглянути елементи системи, яка розроблялась. Було створено дизайни сторінок та елементів вебзастосунку, які в подальшому використовувались при верстанні та розробці вебзастосунку. Було створено дизайни для головної сторінки, хедеру, футеру, а також сторінок з новинами, результатами та активностями.

Було додано нові кастомні типи постів, які необхідні в подальшій роботі з даними для їх фільтрації та сортування. До типів постів було додано таксономії для можливості додавання категорій постів для зручності користувача, а також редакторів вебсайту.

Після розробки вебзастосунку було проведено його тестування, в яких використовувались різні можливі сценарії його роботи. Тестування дозволило віднайти всі помилки в системі, а також перевірити її функціональність. Всі тестування було успішно пройдено, що свідчить про правильність структури розробленого застосунку.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра було створено застосунок проекту розвитку дистанційної освіти майбутнього, з можливістю додавати контент через адмін-панель.

Для досягнення поставленої мети роботи було виконання наступні завдання:

- аналіз предметної сфери;
- обговорення деталей проекту;
- дизайн та розробка логотипу проекту;
- дизайн зовнішнього вигляду сайту проекту;
- верстка сайту проекту;
- розробка та встановлення необхідного функціоналу адмін-панелі;
- тестування та фікс багів;
- відправлення проекту на production server та його запуск в мережу Інтернет.

Для виконання цих завдань було створено UML-діаграми, для зрозумілості архітектури застосунку. Серед цих діаграм:

- діаграма use case;
- діаграма станів;
- діаграма взаємодії;
- діаграми компонентів та розгортання;
- діаграма класів.

Після завершення роботи над проектом було проведено його тестування, що дозволило віднайти всі його помилки та недоліки, які в подальшому було виправлено. Також це дало змогу переглянути розумілість функціоналу та можливостей застосунку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Munster Technological University. URL: <https://www.mtu.ie/> (Last accessed: 02.05.2023).
2. Coursera. URL: <https://www.coursera.org/> (Last accessed: 03.05.2023).
3. Stanford Online. URL: <https://online.stanford.edu/> (Last accessed: 03.05.2023).
4. Zandstra M. PHP 8 Objects, Patterns, and Practice: Mastering OO Enhancements, Design Patterns, and Essential Development Tools. 6th ed. Apress, 2021. 858 p.
5. Prettyman S. Learn PHP 8: Using MySQL, JavaScript, CSS3, and HTML5. 2nd ed. Apress, 2020. 583 p.
6. MacDonald M. WordPress: The Missing Manual: The Book That Should Have Been in the Box. O'Reilly Media, Incorporated, 2020. 558 p.
7. Messenlehner B., Coleman J. Building Web Apps with WordPress: WordPress as an Application Framework. 2nd ed. O'Reilly Media, 2019. 546 p.
8. Duckett J. PHP & MySQL: Server-side Web Development. Wiley, 2022. 672 p.
9. McFedries P. Web Coding & Development All-in-One For Dummies (For Dummies (Computer/Tech)). For Dummies, 2018. 848 p.
10. Russ M. Learning UML 2.0: A Pragmatic Introduction to UML, O'Reilly Media, 2006. 296 p.

ДОДАТОК А

Акт впровадження



Erasmus+KA2 project «Distance Education for Future: Best EU Practices in Response to the Requests of Modern Higher Education Seekers and Labor Market»



Co-funded by the European Union



АКТ

про впровадження результатів розробки Колеснікова Микити Олександровича

Цей акт засвідчує те, що для проєкту Erasmus+KA2 DEFEP «Дистанційна освіта майбутнього: кращі практики ЄС у відповідь на запити сучасних здобувачів вищої освіти і представників ринку праці» № 101083143-DEFEP-ERASMUS-EDU-2022-CBHE було використано програмне забезпечення, створене в результаті виконання кваліфікаційної роботи бакалавра **Колеснікова М. О.** на тему «Вебзастосунок проєкту дистанційної освіти майбутнього в межах програми Erasmus+ KA2».

Розроблене програмне забезпечення дозволяє акцентувати увагу більш широкої аудиторії до завдань та результатів виконання проєкту. Доступ до результатів розробки можна отримати за посиланням <https://defep.chmnu.edu.ua>.

Координаторка проєкту
програми ЕРАЗМУС+КА2,
канд. економ. наук, доцент

Олена ФІЛІМОНОВА

Рисунок 1.1 – Акт впровадження