

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

факультет інформаційних технологій

(факультет)

Кафедра інформаційних технологій та комп'ютерної інженерії

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи ступеня бакалавра

(бакалавра, спеціаліста, магістра)

Студента Губаренко Івана Дмитровича

(ПІБ)

академічної групи 126-19-1

(шифр)

спеціальності 126 «Інформаційні системи та технології»

(код і назва спеціальності)

за освітньо-професійною програмою

«Інформаційні системи та технології»

(офіційна назва)

на тему Розробка вебсайту підприємства з галузі рибного господарства

(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	проф. Коротенко Г.М.			
розділів:				

Рецензент	доц. Ширін А.Л.			
-----------	-----------------	--	--	--

Нормоконтролер	проф. Коротенко Г.М.			
----------------	----------------------	--	--	--

Дніпро
2023

ЗАТВЕРДЖЕНО:

завідувач кафедри

інформаційних технологій

та комп'ютерної інженерії

(повна назва)

Гнатушенко В.В.

(підпис)

(прізвище, ініціали)

« _____ » _____ 2023 року

ЗАВДАННЯ

на кваліфікаційну роботу

ступеня бакалавр

(бакалавра, спеціаліста, магістра)

студенту Губаренко І.Д. академічної групи 126-19-1

(прізвище та ініціали)

(шифр)

спеціальності 126 « Інформаційні системи та технології »

за освітньою-професійною програмою _____

«Інформаційні системи та технології»

на тему Розробка вебсайту підприємства з галузі рибного господарства

затверджену наказом ректора НТУ «Дніпровська політехніка» від 16.05.2023 р. № 350-с

Розділ	Зміст	Термін виконання
Розділ 1	Ознайомлення з матеріалами сайтів, літературою та виконання пошуку технологій і формування відповідних рішень	01.02.2023 – 30.03.2023
Розділ 2	Реалізація проектних рішень	01.04.2023 – 20.06.2023

Завдання видано

(підпис керівника)

Коротенко Г.М.

(прізвище, ініціали)

Дата видачі 01.02.2023 р.

Дата подання до екзаменаційної комісії 20.06.2023 р.

Прийнято до виконання _____ Губаренко І.Д.

(підпис студента)

(прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 111 с., 9 рис., 5 додатків, 13 джерел.

Об'єкт розробки: веб-сайт рибгосподарства «РИБ СТАВОК».

Мета роботи: надати компанії можливість бути представленою в мережі інтернет, залучати нових клієнтів через пошукові запити, надати повний спектр інформації по підприємству.

У вступі наведено інформацію про стан проблеми, здійснено аналіз програмного забезпечення та технічних аспектів розробки веб-сайту.

В першому розділі наведена ключова інформація по стану технічних та програмних засобів для вирішення поточної проблеми. Нами було проаналізовано які з технічних засобів краща за все підходять для розробки веб-сайту рибгосподарства.

У другому розділі нами був розроблений веб-сайт та повністю описаний його функціонал. Нами було надані рекомендації відносно подальшого розвитку та його оптимізації.

Практичне значення кваліфікаційної роботи полягає у тому що веб-сайт готовий для SEO-оптимізації та буде приносити клієнтів бізнесу.

Розроблене технологічне рішення може бути використано будь-яким рибгосподарством для того, щоб бути представленими в мережі Інтернет.

ВЕБ-САЙТ, PHP, ФРЕЙМВОРК, SEO-ОПТИМІЗАЦІЯ, РИБГОСПОДАРСТВО, ЗАЛУЧЕННЯ КЛІЄНТІВ, БЕЗПЕКА, ЗАХИСТ ВІД ХАКЕРІВ, АВТОМАТИЧНЕ ОНОВЛЕННЯ.

ABSTRACT

The work is featured by 110 pages, 9 figures, 5 appendices, 13 sources.

Object of development: the website of the fishery "RIB STAVOK".

The purpose of the work: to provide the company with the opportunity to be represented on the Internet, to engage new customers through search requests, to provide a full range of information about the company.

The introduction provides information on the state of the problem, analyzes the software and technical aspects of website development.

The first chapter provides key information on the state of technical and software tools for solving the current problem. We have analyzed which of the technical means are best suited for the development of a fish farming website.

In the second chapter, we developed the website and fully described its functionality. We were given recommendations for further development and optimization.

The practical significance of the qualification work is that the website is ready for SEO-optimization and will bring customers to the business.

The developed technological solution can be used by any fishery in order to be presented on the Internet.

WEBSITE, PHP, FRAMEWORK, SEO OPTIMIZATION, FISHERIES, ENGAGING CUSTOMERS, SECURITY, PROTECTION AGAINST HACKERS, AUTO UPDATE.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ СТАНУ ОБЛАСТІ РІШЕННЯ ЗАДАЧІ	8
1.1. Вибір засобу розробки для FrontEnd`у.....	8
1.1.1. Поняття FrontEnd.....	8
1.1.2. Компоненти для розробки FrontEnd`у	8
1.1.3 Поняття фреймворка.....	9
1.1.4. Вибір CSS фреймворку. Bootstrap, Foundation, Tailwind CSS	10
1.1.5. Фреймворк Bootstrap.....	11
1.1.6. Переваги фреймворка Bootstrap.....	12
1.2. Вибір засобу розробки для BackEnd`у	12
1.2.1. Поняття BackEnd	12
1.2.2. Поняття BackEnd-фреймворку.....	14
1.2.3. Опис фреймворка Django	14
1.3. Вибір СУБД	15
1.3.1. Поняття СУБД	15
1.3.2. Вибір СУБД: MYSQL, MongoDB, PostgreSQL	15
1.4. Вибір хостингу	18
1.5. Вибір системи управління версіями.....	19
1.5.1. Поняття Системи Управління Версіями GIT	19
1.5.2. Вибір Системи Управління Версіями GitHub	19
РОЗДІЛ 2 ПРОЕКТНІ РІШЕННЯ.....	21
2.1. Технічне завдання	21
2.1.1. Найменування і область застосування.....	21
2.1.2. Підстави для розробки.....	23
2.1.3. Призначення розробки.....	24

2.1.4. Вимоги до функціональних характеристик.....	24
2.1.5. Вимоги до надійності.....	25
2.1.6. Вимоги до складу і параметрів технічних засобів.....	27
2.1.7. Вимоги до інформаційної та програмної сумісності.....	29
2.1.8. Вимоги до програмної документації.....	30
2.2. Створення сховища в GitHub.....	31
2.2.1. Розподіл ролей виконавців.....	31
2.2.2. GitFlow.....	31
2.3. Створення таблиць в БД.....	32
2.3.1. Приклад створення таблиці в БД.....	33
2.3.2. Схема БД.....	35
2.3.3. Список таблиць.....	36
2.4. Архітектурна схема сайту.....	37
2.4.1. Алгоритм створення бази даних.....	38
2.4.2 Відображення сторінки.....	38
2.4.3. Опис розробленого сайту і його інтерфейсу.....	39
ВИСНОВКИ.....	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	41
Додаток А. Приклади коду програми.....	43
Додаток Б. Відгук керівника кваліфікаційної роботи.....	109
Додаток В. Рецензія.....	110
Додаток Г. Довідка.....	111

ВСТУП

Зв'язок задач, які вирішуються, з об'єктом діяльності фахівця спеціальності 126 «Інформаційні системи та технології». Метою роботи є розробка сайту який би допоміг рибогосподарству «РИБ СТАВОК» бути присутнім в інтернеті та залучати потенційних клієнтів завдяки пошуковим сайтам та зручному функціоналу.

Сучасний стан проблеми. Багато підприємств не пов'язаних з інтернет торгівлею вважають, що можна не мати власного сайту і бути успішним тому, що торгові представники в змозі «підняти» рівень продажів компанії на достатній для її функціонування рівень. Це критична прогалина в розумінні того як функціонує сучасна економіка та що потенційні клієнти хочуть бачити від постачальника.

Мета кваліфікаційної роботи. Мета полягає в розробці зручного та функціонального сайту для рибного господарства який би повністю покрити його потреби в присутності в інтернеті та в залучанні нових клієнтів. Сайт повинен мати високий рівень SEO-оптимізації та складатись з модулів які постійно оновлюються та надають високий рівень безпеки від взломів.

Обґрунтування актуальності теми. Дана тема актуальна через те, що будь-яка компанія повинна мати власний сайт в інтернеті не тільки, щоб надати загальні відомості про компанію як часи роботи, чи адресу але і залучати нових клієнтів чи постачальників до активної співпраці.

Конкретизацію постановки задач, які вирішуються у кваліфікаційній роботі бакалавра. Таким чином, необхідно створити сайт з використанням мови PHP за допомогою одного з фреймворків та SQL-based бази даних.

РОЗДІЛ 1

АНАЛІЗ СТАНУ ОБЛАСТІ РІШЕННЯ ЗАДАЧІ

1.1. Вибір засобу розробки для FrontEnd`у

1.1.1. Поняття FrontEnd

Згідно вікіпедії, інтерфейсна веб-розробка — це розробка графічного інтерфейсу користувача веб-сайту за допомогою HTML, CSS і JavaScript, щоб користувачі могли переглядати цей веб-сайт і взаємодіяти з ним.

Інструменти, що використовуються для фронтенд розробки. Існує кілька інструментів і платформ, таких як WordPress, Joomla та Drupal, які можна використовувати для розробки зовнішньої частини веб-сайту.

Мова розмітки гіпертексту (HTML) є основою будь-якого процесу розробки веб-сайту, без якої веб-сторінка не існує. Гіпертекст означає, що текст має вбудовані посилання, які називаються гіперпосиланнями. Коли користувач клацає слово чи фразу, яка містить гіперпосилання, відкривається інша веб-сторінка. Мова розмітки вказує, що текст можна перетворити на зображення, таблиці, посилання та інші представлення. Саме HTML-код забезпечує загальну структуру вигляду сайту. HTML був розроблений Тімом Бернерсом-Лі. Остання версія HTML називається HTML5 і була опублікована 28 жовтня 2014 року за рекомендацією W3C. Ця версія містить нові та ефективні способи обробки таких елементів, як відео та аудіофайли [7].

1.1.2. Компоненти для розробки FrontEnd`у

Все на сторінці, від логотипу до панелі пошуку, кнопок, загального макета та того, як користувач взаємодіє зі сторінкою, було створено розробником

інтерфейсу. Інтерфейсні розробники відповідають за зовнішній вигляд веб-сайту.

Інтерфейсні розробники також повинні переконатися, що веб-сайт добре виглядає на всіх пристроях (телефонах, планшетах і екранах комп'ютерів).

Для того щоб бути фронтенд розробником потрібно знати три основні мови, це HTML, CSS і JavaScript. Звідти ми можемо зосередитися на фреймворках, бібліотеках та інших корисних інструментах.

HTML розшифровується як HyperText Markup Language. HTML відображає вміст сторінки, наприклад кнопки, посилання, заголовки, абзаци та списки. HTML не використовується для стилізації. Саме для цього і потрібен CSS.

CSS розшифровується як каскадні таблиці стилів. CSS відповідає за стиль нашої веб-сторінки, включаючи кольори, макети та анімацію.

Адаптивний дизайн необхідний для створення веб-сайтів, які добре виглядатимуть на всіх пристроях.

Доступність — це практика забезпечення того, щоб кожен міг легко користуватися вашими веб-сайтами. Допоміжні технології дозволяють людям з вадами отримувати всю повноту інформації та бути лояльним користувачем.

JavaScript дозволяє користувачам взаємодіяти з веб-сторінкою. Приклади JavaScript можна знайти практично на будь-якій веб-сторінці.

Наприклад натискаючи кнопку «Меню» у верхній частині сторінки, відкривається спадний список параметрів. Щоразу, коли ми натискаємо цю кнопку, вона перемикається між відкриттям і закриттям меню.

JavaScript можна використовувати в онлайн-іграх, веб-сторінках, мобільних програмах тощо [7].

1.1.3 Поняття фреймворка

CSS Frameworks, бібліотеки та препроцесори. Опанувавши основи CSS, можна почати працювати з різними фреймворками та бібліотеками. Ці інструменти були створені, щоб допомогти прискорити процес розробки.

Коли ми розробляємо свою програму, у вашому коді будуть помилки, які потрібно виправити. Налагодження — це виявлення цих помилок і їх виправлення. Тестування — це ще одна важлива навичка, якій потрібно навчитися. Написання тестів для вашого коду — це спосіб переконатися, що ваш код виконує те, що він повинен робити [3].

1.1.4. Вибір CSS фреймворку. Bootstrap, Foundation, Tailwind CSS

Чи Tailwind CSS використовується ширше, ніж Bootstrap? Для початку давайте скористаємося даними GitHub, щоб визначити, яка з двох фреймворків CSS є більш популярною.

На GitHub Tailwind CSS має 3,1 тис. розгалужень і 61,1 тис. зірочок, тоді як Bootstrap має 77,3 тис. розгалужень і 160 тис. зірочок. З цього можна зробити загальне припущення, що Bootstrap є більш популярним, ніж Tailwind CSS. Ось ще один факт: Bootstrap, здається, більш популярний, ніж Tailwind CSS, згідно з останніми даними State of CSS на вкладці поінформованості.

Такі фреймворки, як Bootstrap, абстрагували створення компонентів до такої міри, що змушували розробників використовувати лише доступні надані шаблони. Те саме стосується інших фреймворків типу комплекту інтерфейсу користувача. Дехто може заперечити, що заміна фреймворку нашим власним CSS є варіантом, але якщо ми багато чого перевизначаємо, то чи є сенс використовувати фреймворк? Ми б завантажили бібліотеку й усе ще писали

власний код — це лише додаткові файли, про які потрібно турбуватися, і ми навіть не економимо час.

Ще одна проблема, яку я виявив із сайтами Bootstrap, полягає в тому, що вони майже завжди виглядають однаково, тому це перешкоджає нашій здатності включати творчість у середовище розробників. Це одна з переваг Tailwind CSS: його здатність легко створювати складні користувацькі інтерфейси, не заохочуючи будь-які два сайти виглядати однаково.

Ще одна ключова перевага використання Tailwind CSS над Bootstrap полягає в тому, що, оскільки програми та сайти складаються з попередньо розроблених віджетів, Tailwind CSS не нав'язує дизайнерські рішення, які важко змінити. Робота з Tailwind CSS означає використання набору службових класів, які дозволяють працювати саме з тим, що вам потрібно. На мій погляд, це гарний спосіб створити інтерфейс користувача, який є більш гнучким для творчості розробників.

Ще одна перевага, яку цінує будь-який розробник, це те, що ніколи не потрібно турбуватися про те, що зміни в одному елементі впливають на інший пов'язаний елемент. Більше не потрібно перемикатися між HTML і таблицями стилів у вашому редакторі, більше не повертатися, щоб перевірити та побачити, як ви назвали інший елемент [13].

1.1.5. Фреймворк Bootstrap

Bootstrap — це найпопулярніший фреймворк HTML, CSS і JavaScript для створення адаптивних, орієнтованих на мобільні пристрої проектів в Інтернеті. Tailwind CSS, з іншого боку, є найпопулярнішою утилітною структурою CSS для швидкої розробки інтерфейсу користувача.

Основна відмінність між TailwindCSS і Bootstrap полягає в тому, що Tailwind CSS не є комплектом інтерфейсу користувача. На відміну від наборів

інтерфейсу користувача, таких як Bootstrap, Bulma та Foundation, Tailwind CSS не має теми за замовчуванням або вбудованих компонентів інтерфейсу користувача. Натомість він постачається з попередньо розробленими віджетами, які можна використовувати для створення сайту з нуля.

Bootstrap відомий своєю швидкістю реагування, тоді як прихильники Tailwind CSS зазвичай цінують можливість налаштування фреймворка. Найкращий вибір для вас залежить від ваших пріоритетів і вимог до проекту, але давайте поговоримо про те, чому Tailwind CSS швидко набуває популярності та ширшого використання [2].

1.1.6. Переваги фреймворка Bootstrap

Bootstrap — це компонентна структура, що означає, що вона постачається з попередньо зібраними компонентами та містить інші утиліти для розшарування дисплеїв, інтервалів тощо.

Tailwind, з іншого боку, CSS — це фреймворк, який насамперед утилітний. Використання Tailwind CSS схоже на написання звичайного CSS. На відміну від Bootstrap, він не має попередньо зібраних компонентів.

Завдяки готовим компонентам Bootstrap ми можемо написати небагато CSS. У результаті незначні зміни в дизайні можуть призвести до збоїв. За допомогою Tailwind CSS ви стилізуєте свої елементи з нуля, використовуючи синтаксис стилю Tailwind. За допомогою Tailwind CSS легше вносити зміни, тому що вам потрібно лише видалити деякі класи CSS.

Отже, що простіше, Tailwind CSS чи Bootstrap? Це значною мірою залежить від варіанту використання та кривої навчання. Якщо ми освоїли CSS, ми вже на шляху до освоєння Tailwind CSS. З цієї причини Tailwind CSS є чудовим вибором для більшості проектів [2].

1.2. Вибір засобу розробки для BackEnd`у

1.2.1. Поняття BackEnd

Серверна сторона веб-сайту. Вона зберігає та впорядковує дані, а також гарантує, що все на клієнтській стороні веб-сайту працює нормально. Це частина веб-сайту, яку ми не можемо бачити та взаємодіяти з нею. Це частина програмного забезпечення, яка не контактує безпосередньо з користувачами. Користувачі опосередковано отримують доступ до частин і характеристик, розроблених дизайнерами серверної частини, через зовнішню програму. Такі види діяльності, як написання API, створення бібліотек і робота з системними компонентами без інтерфейсів користувача або навіть систем наукового програмування, також включені в бекенд.

Внутрішня частина побудована за допомогою деяких мов, які обговорюються нижче:

PHP: PHP — це серверна мова сценаріїв, розроблена спеціально для веб-розробки. Оскільки код PHP виконується на стороні сервера, він називається мовою сценаріїв на стороні сервера.

C++: це мова програмування загального призначення, яка зараз широко використовується для конкурентного програмування. Він також використовується як серверна мова.

Java: Java є однією з найпопулярніших і широко використовуваних мов програмування та платформ. Він дуже масштабований. Компоненти Java легкодоступні та допоможуть нам зрозуміти належну структуру, концепції, функції тощо.

Python: Python — це мова програмування, яка дає змогу швидко працювати та ефективніше інтегрувати системи. Це також дуже важлива мова для серверної частини.

Node.js: Node.js — це кросплатформне середовище виконання з відкритим кодом для виконання коду JavaScript поза браузером. Більшість людей плутають і розуміють, що це фреймворк або мова програмування. Ми часто використовуємо Node.js для створення внутрішніх служб, таких як API, наприклад Web App або Mobile App. Він використовується у виробництві великими компаніями, такими як Paypal, Uber, Netflix, Walmart тощо [6].

1.2.2. Поняття BackEnd-фреймворку

Express – це фреймворк Nodejs, який використовується для розробки на стороні серверу. Він використовується для створення односторінкових, багатосторінкових і гібридних веб-додатків. З його допомогою Ми можемо обробляти кілька різних HTTP-запитів.

Ruby on Rails — це платформа на стороні сервера, що дотримується шаблону архітектури модель-подання-контролер. Він надає такі структури за замовчуванням, як веб-сервіси, веб-сторінки та бази даних.

Laravel є надійною платформою веб-додатків для PHP. Особливістю, яка робить його ідеальним, є повторне використання компонентів різних фреймворків для створення веб-додатку.

Spring – ця платформа на стороні сервера забезпечує підтримку інфраструктури для програм Java. Він діє як підтримка різних фреймворків, таких як Hibernate, Struts, EJB тощо. Він також має розширення, які допомагають швидко та легко розробляти програми Java [6].

1.2.3. Опис фреймворка Django

Django — це веб-фреймворк на Python, що дотримується шаблону «модель-шаблон-подання». Він використовується для створення великих і складних веб-додатків. Серед його особливостей — швидкість, безпека та можливість масштабування [4].

1.3. Вибір СУБД

1.3.1. Поняття СУБД

База даних — це організований набір даних, які зберігаються та доступні в електронному вигляді. Невеликі бази даних можна зберігати у файловій системі, тоді як великі бази даних розміщуються на комп'ютерних кластерах або хмарних сховищах. Розробка баз даних охоплює формальні методи та практичні міркування, включаючи моделювання даних, ефективне представлення та зберігання даних, мови запитів, безпеку та конфіденційність конфіденційних даних, а також питання розподілених обчислень, включаючи підтримку одночасного доступу та стійкість до відмов.

1.3.2. Вибір СУБД: MySQL, MongoDB, PostgreSQL

Різниця SQL проти NoSQL. «Системи управління реляційними базами даних» (RDBMS) з'явилися в 1970-х роках. Едгар Френк Кодд, британський комп'ютерний науковець, який працює в IBM, винайшов концепцію RDBMS. РСУБД часто називають «базами даних SQL», оскільки вони використовують SQL («Мова структурованих запитів»).

Для порівняння, бази даних NoSQL з'явилися наприкінці 2000-х. Термін «NoSQL» означає «не SQL» або не тільки SQL.

Відмінності між базами даних SQL і NoSQL такі:

Модель зберігання даних: бази даних SQL зберігають дані в таблицях із фіксованими рядками та стовпцями. Бази даних NoSQL зберігають такі дані, як документи, пари ключ-значення, широкі стовпці та графіки. Зауважте, що дані «широкого стовпця» стосуються таблиць із рядками та динамічними стовпцями.

Тип даних для зберігання: бази даних SQL дозволяють зберігати лише структуровані дані. Бази даних NoSQL також дозволяють зберігати неструктуровані дані.

Мета: бази даних SQL є базами даних загального призначення. Бази даних NoSQL можуть бути загального призначення, однак їх можна використовувати для зберігання документів, пар ключ-значення, даних із широкими стовпцями тощо.

Схема: бази даних SQL використовують жорстку схему, однак бази даних NoSQL гнучкі.

Масштабування: Бази даних SQL масштабуються за допомогою більших серверів, що є вертикальним масштабуванням. Сервери NoSQL масштабуються горизонтально, тобто з більшою кількістю серверів.

«Об'єднання»: бази даних SQL підтримують «Об'єднання», однак бази даних NoSQL їх не підтримують.

Відповідність ACID з кількома записами: бази даних SQL підтримують властивості ACID з кількома записами («Атомність», «Послідовність», «Ізоляція» та «Стійкість»), однак бази даних NoSQL часто їх не підтримують.

MySQL, PostgreSQL, SQLite, Oracle і Microsoft SQL Server є популярними прикладами баз даних SQL. MongoDB, Cassandra, Redis, Memcached і Amazon DynamoDB є популярними прикладами баз даних NoSQL.

Бази даних SQL. Шведська компанія під назвою MySQL AB створила MySQL. Девід Аксмарк, Аллан Ларссон і Майкл Віденіус спільно заснували цю компанію. Widenius і Axmark почали розробку MySQL у 1994 році, а вперше запустили її в 1995 році.

Пізніше Sun Microsystems придбала MySQL AB. Oracle придбала Sun Microsystems у 2010 році. Хоча MySQL є відкритим вихідним кодом, Oracle надає помітну підтримку для нього. Зверніть увагу, що MySQL також доступний за комерційними ліцензіями.

На момент написання цього останнього стабільного випуску MySQL є 8.0.19. Цей випуск було запущено в січні 2020 року.

MySQL займає чільне місце серед RDBMS з відкритим кодом. Багато відомих компаній використовують його, оскільки MySQL пропонує кілька ключових переваг.

Розробка PostgreSQL почалася в 1980-х роках, і ця RDBMS є наступником Ingres. Команда дослідників і розробників з Каліфорнійського університету в Берклі розробила Ingres. Майкл Стоунбрейкер, лідер цієї команди, розпочав розробку Postgres.

Стоунбрейкер і його команда помітили кілька обмежень RDBMS, і вони хотіли вирішити їх у рамках цього нового проекту. Новий проект вони назвали «Postgres». Після кількох років досліджень і розробок вони запустили PostgreSQL у 1996 році.

PostgreSQL користується значною підтримкою розробників. Ця група розробників відома як «Глобальна група розробки PostgreSQL», і вона продовжує вдосконалювати PostgreSQL.

На момент написання останнім стабільним випуском PostgreSQL є 15.3. Глобальна група розробки PostgreSQL запустила цей випуск у листопаді 2020 року.

Серед визнаних RDBMS з відкритим вихідним кодом PostgreSQL користується великою популярністю. Ця RDBMS також відома як «Postgres», і вона користується великою популярністю.

Щоб дізнатися більше про бази даних SQL (MySQL, PostgreSQL і SQLite), прочитайте цю вичерпну статтю – MySQL проти PostgreSQL проти SQLite: порівняння між 3 популярними RDBMS.

MongoDB — одна з найпопулярніших баз даних NoSQL. Це база даних загального призначення, яка базується на документах. Як база даних документів, MongoDB зберігає дані в JSON-подібних документах. 10gen, компанія-виробник програмного забезпечення, почала розробку MongoDB у 2007 році. Ця компанія перейшла на модель розробки з відкритим кодом у 2009 році, і того ж року запустила MongoDB. У 2013 році 10gen змінила назву на MongoDB Inc.

Творці MongoDB розробили його на C++. Вони також використовували Go, JavaScript і Python для кількох завдань. MongoDB використовує модель ліцензування «доступне джерело».

Ця база даних пройшла кілька ітерацій розробки. Його останній стабільний випуск вийшов у грудні 2020 року, і це випуск 6.0.5 [10].

1.4. Вибір хостингу




	.SHOP 198 ₴	.STORE 84 ₴	.ONLINE 120 ₴	.COM.UA 480 ₴	.SITE 84 ₴	.UNO 84 ₴	.IN.UA 486 ₴	.TECH 120 ₴	.WEBSITE 84 ₴	.COM 696 ₴	.SPACE 60 ₴	.FUN 60 ₴
	Швидкий Базовий план для невеликих сайтів			Кращий Оптимально для кількох сайтів				Якісний Готове рішення для веб студій і великих проектів				
📍 Місце на NVMe-диску	10 Gb			20 Gb				30 Gb				
🛡️ Безкоштовний SSL-сертифікат	✓			✓				✓				
🔧 Конструктор сайтів	✓			✓				✓				
🌐 Apache, OpenLiteSpeed	✓			✓				✓				
📄 Сайтів	1			5				необмежено*				
📊 Memory limit (RAM)	512 МБ			1024 МБ				1536 МБ				
🌐 Розташування серверів	🇺🇦 🇩🇪 🇫🇷 🇮🇹			🇺🇦 🇩🇪 🇫🇷 🇮🇹				🇺🇦 🇩🇪 🇫🇷 🇮🇹				
💰 30-денне повернення грошей	від 152 ₴/міс 231 ₴/міс			від 285 ₴/міс 435 ₴/міс				від 407 ₴/міс 621 ₴/міс				
	Замовити			Замовити				Замовити				
	5 днів на тест			5 днів на тест				5 днів на тест				
Детальніше про тарифи хостингу												
 Для більш надійного зберігання даних ми створюємо щотижневу копію хостингу та поштових скриньок на сервери Німеччини.				 Перебуваючи під постійним обстрілом російських ракет, зникає бажання обслуговувати винних у цьому. Підтримка хостингу та доменів окупантів – припинено.				 Наш дата-центр продовжує свою роботу. Техпідтримка знаходиться у різних містах, що дозволяє надавати послуги у повному обсязі.				

Рис. 1.1 Ціно-технічні пропозиції від українського хостинг-провайдеру
Український сегмент інтернету надає багато видів хостинг-сервісів та сервісів по реєстрації домену.

1.5. Вибір системи управління версіями

1.5.1. Поняття Системи Управління Версіями GIT

Git — це система контролю версій, яку ми завантажувемо на свій комп'ютер. Важливо, щоб ми використовували Git, якщо хочемо співпрацювати з іншими розробниками над проектом кодування або працювати над власним проектом.

Щоб перевірити, чи Git уже встановлено на нашому комп'ютері, ми можемо ввести команду `git--version` у терміналі.

Якщо у нас уже встановлено Git, ми побачите, яка у нас версія. Якщо у нас не встановлено Git, ви можете відвідати веб-сайт Git і легко виконати інструкції із завантаження, щоб установити правильну версію для нашої операційної системи.

1.5.2. Вибір Системи Управління Версіями GitHub

GitHub — це продукт, який дозволяє розміщувати наші проекти Git десь на віддаленому сервері (або іншими словами, у хмарі).

Важливо пам'ятати, що GitHub – це не Git. GitHub — це просто хостінг. Є інші компанії, які пропонують послуги хостингу, які роблять те саме, що GitHub, наприклад Bitbucket і GitLab.

Можна використовувати Git, вводячи команди в терміналі, або використовувати графічний інтерфейс користувача (GUI), наприклад Sourcetree або GitKraken.

Якщо обирати термінал, нам доведеться шукати, які команди Git нам знадобляться.

На щастя, нам не потрібно вчити їх напам'ять. За винятком кількох команд, які ми використовуватимете найчастіше, решту ми можете шукати, коли вони нам знадобляться (це те, що робить більшість розробників, навіть ті з десятиліттями досвіду). Git пропонує детальну документацію на своєму веб-сайті.

Якщо ми вирішите використовувати графічний інтерфейс користувача, то різні дії, які вам потрібно виконати, відобразатимуться більш наочно.

Незалежно від того, чи ми вирішите використовувати термінал чи графічний інтерфейс, нам потрібно буде зрозуміти основи роботи Git, щоб впевнено ним користуватися.

Щоб використовувати Git, нам потрібно мати проект, який ми хочемо контролювати версії. Це може бути або новий проект, або існуючий проект.

Якщо це новий проект, то нам потрібно створити нову папку проекту (ми можемо скористатися командою `mkdir`), а потім перейти до цієї папки проекту в терміналі [12].

РОЗДІЛ 2

ПРОЕКТНІ РІШЕННЯ

2.1. Технічне завдання

Кваліфікаційна робота має завдання по розробці інформаційного сайту рибного господарства наявними техніко-програмними засобами. Серед яких слід зазначити PHP, MySQL, HTML, CSS та інші.

2.1.1. Найменування і область застосування

Будь-яка компанія повинна мати свій сайт в мережі інтернет, щоб потенційні клієнти мали змогу дізнатись більше про фірму, переглянути перелік сервісів, які вона надає, використати контакт-форму для того, щоб зв'язатись з представниками компанії.

Присутність бізнесу в Інтернеті для рибного господарства, може мати значний вплив на його успіх. У наш час деякі компанії все ще не усвідомлюють, що більшість їхніх клієнтів відвідають їхні веб-сайти, перш ніж зробити покупку.

Наявність потужної онлайн-присутності, зокрема веб-сайту, може стати ключем для збільшення прибутку.

Однією з головних причин, чому рибгосподарство повинно мати веб-сайт для рибного господарства, є підвищення довіри до компанії. Скоріш за все, є кілька постачальників, які пропонують подібні послуги. Один із способів виділитися – мати веб-сайт, який добре виглядає та чітко передає якісну інформацію вашим споживачам.

Без веб-сайту люди можуть поставити під сумнів легітимність рибної компанії та її фактичне існування. Наявність веб-сайту - це можливість

справити чудове перше враження та переконати людей у тому, що ви — справжнє рибгосподарство.

Демонстрація бренду потенційним клієнтам - одна з найважливіших речей, які компанія можете зробити. Чітко визначивши які в вас цінності та яких результатів рибгосподарство намагається досягнути (рис. 2.1).



Рис. 2.1 Фото для заохочування клієнтів

Задля більшого ефекту, на сайті потрібно розміщувати ілюстрації з поясненнями та відео які свідчать про те що ваші виробничі процеси відповідають високим стандартам якості.

Це також те, що може виділити рибгосподарство поміж конкурентів. Без веб-сайту зробити це може бути неймовірно складно, оскільки люди не можуть легко знайти якісну та надійну інформацію про ваші ставки.

Можливо, одна з найцікавіших причин створення веб-сайту для рибгосподарства полягає в тому, що це може збільшити шанси на залучення потенційних клієнтів.

Коли люди знайдуть рибгосподарство в Інтернеті, цікавляться продуктом або послугою та захочуть дізнатися більше, завдяки інформації на веб-сайті вони знатимуть, як зможуть зв'язатися, що дає можливість збільшити продажі. Незважаючи на те, що веб-сайти мають вартість, за правильного використання вони мають позитивну рентабельність інвестицій.

Багато рибгосподарств отримують дзвінки від потенційних або існуючих клієнтів із простими запитаннями про місцезнаходження ставків, можливість доставки та години роботи. Якщо господарство пропустить дзвінок, клієнт скоріш за все залишиться незадоволеним. Зайві дзвінки також можуть відволікати персонал господарства від зосередження на управлінні господарством. Веб-сайт може зменшити кількість дзвінків і підвищити внутрішню продуктивність. У той же час це допомагає клієнтам знаходити корисну інформацію без необхідності телефонувати, що в кінцевому підсумку забезпечує загальну кращу взаємодію з користувачем.

2.1.2. Підстави для розробки

Розроблений сайт дозволить рибгосподарству бути представленою в інтернеті та дасть можливість розпочати діджитал-маркетинг. Зай ого допомогою також планується використовувати цифровий маркетинг для збільшення потенційних клієнтів і розвитку рибгосподарства спрямовуючи трафік на сайт. Щоб зробити це ефективно, слід використовувати органічний трафік, який спрямовувався на ваш сайт, щоб ви могли націлюватися на найбільш кваліфікованих клієнтів і отримати найкращу рентабельність інвестицій у рекламу. Це те, що не можна налаштувати заднім числом, тому

найкраще запуснути сайт на ранній стадії, навіть якщо ви зараз не плануєте інвестувати в рекламу.

2.1.3. Призначення розробки

Сайт буде допомагати отримувати органічний трафік з пошукової системи Гугл, де люди будуть цікавитись рибгосподарствами та ціною політикою на продукцію.

Наші зусилля в оптимізації сайту залежать від значного тестування, яке дасть розуміння що користувач робить на сайту. Дізнавшись, що приваблює відвідувачів, і проаналізувавши, як вони залучаються, ми можемо звукити свою увагу до чотирьох ключових областей.

Вміст: чи вирішує вміст нашого сайту проблему цільової аудиторії?

SEO: чи можуть люди знайти наш сайт за допомогою пошукових систем?

UX: Чи простий наш сайт у навігації та використанні?

Розміщення СТА: чи наші заклики до дії привертають увагу відвідувачів і чи збільшують кількість кліків?

Важливість оптимізації сайту. Оптимізація веб-сайту вимагає зусиль кількох команд та їхніх зацікавлених сторін. Для цього нам слід бути впевненим, що оптимізація сайту та зростання доходу нерозривно пов'язані.

Ми можемо створити спеціальні сторінки для різних видів риби. Ми зможемо проаналізувати за допомогою функціоналу сайту який з видів риби більш популярний та саме її поставити за замовчування на головну.

2.1.4. Вимоги до функціональних характеристик

Сайт рибгосподарства повинен бути розроблений на PHP та мати всі можливості для оптимізації при оновленні версії PHP в майбутньому. База повинна бути на MYSQL чи інших програмних продуктах, які б дозволили систематизувати отриманні данні.

Сайт повинен містити декілька сторінок які б включали інформацію по рибі та її характеристиках та контакту інформацію.

2.1.5. Вимоги до надійності

SSL (Secure Sockets Layer) - це протокол, який створює зашифроване з'єднання між веб-сервером і веб-браузером. Це означає, що будь-які дані, якими обмінюються відвідувач і веб-сайт, будуть безпечними.

Наявність сертифіката SSL для веб-сайту WordPress є обов'язковою щоб підвищити рівень довіри до нашого рибгосподарства. Сертифікат SSL допоможе захистити конфіденційну платіжну інформацію клієнтів, якщо вони вирішать купити щось за допомогою сайту.

На щастя, вам не потрібні технічні знання, щоб захистити ваш веб-сайт за допомогою SSL. Ми можете легко отримати сертифікат SSL від хостинг-провайдера, реєстратора домену або центру сертифікації (CA). Хоча ціни відрізняються, ми можете отримати сертифікат SSL безкоштовно.

Багато хостингових компаній надають їх безкоштовно у своїх пакетах хостингу. Наприклад, якщо ви розміщуєте свій веб-сайт у центрі CMS HubSpot, ви можете захистити свій вміст і дані потенційних клієнтів за допомогою безкоштовного стандартного SSL.

Якщо наш хостинг-провайдер цього не зробить, ми можете безкоштовно отримати сертифікат SSL від центрів сертифікації, як Let's Encrypt. Ми можемо знайти список безкоштовних або недорогих центрів сертифікації SSL тут.

Застаріле програмне забезпечення може поставити веб-сайт під загрозу вірусів, кібератак та інших проблем безпеки. Щоб уникнути цих проблем, ми будемо постійно оновлювати сайт, регулярно перевіряючи наявність оновлень або встановлюючи автоматичне оновлення. Ці оновлення зазвичай містять виправлення безпеки від розробників, тому дуже важливо зробити їх якнайшвидше.

Використання простих паролів, таких як загальні слова, номерні послідовності, ваше ім'я або назва сайту, може бути дуже небезпечним кроком. Такі паролі дозволяють легко зламати сайт, отримавши доступ до його інформації.

Використання надійних паролів – простий і безкоштовний спосіб захистити свій веб-сайт. Надійні паролі включають:

- Комбінація букв і цифр

- Великі, а також малі літери

- Спеціальні символи

Якщо нам важко придумати надійний пароль, наш браузер може його запропонувати.

Браузер пропонує пароль для безпеки сайту. Ми також можемо використовувати безкоштовний менеджер паролів, як Dashlane, щоб контролювати свої паролі на комп'ютері, ноутбуках і мобільних пристроях.

- Додавання двофакторної автентифікацію.

Двофакторна автентифікація (2FA) — це рівень безпеки, який посилює безпеку нашого сайту. Двофакторна автентифікація працює з двома різними типами структур, щоб запобігти хакеру отримати доступ до сайту.

Рівень безпеки пов'язує ваш пароль із текстовим кодом, розпізнаванням обличчя чи сітківки ока або з вашим відбитком пальця як двосторонню головоломку. Наш відбиток пальця або сітківку ока можна зафіксувати за допомогою сканера на мобільному пристрої.

Кожен, хто спробує зламати ваш веб-сайт, повинен буде вирішити обидві головоломки. Двофакторна автентифікація не є бездоганною системою, але вона покращить безпеку веб-сайту.

Ми можемо безкоштовно налаштувати 2FA на своєму сайті для обмеженої кількості користувачів із такими постачальниками, як DUO.

Створення регулярних резервних копій сайту рибгосподарства не є проактивним підходом до безпеки веб-сайту, але це важливо у випадках зловмисних атак, апаратних збоїв або стихійних лих. Наявність резервної копії сайту означає, що його можна швидко відновити. Без резервної копії ви ризикуєте втратити всі свої дані та налаштування.

Ми також можемо створити резервну копію основних файлів сайту рибгосподарства та вмісту наших баз даних. Резервне копіювання заощадить наш час, гроші та зусилля, необхідні для вирішення проблеми втрати даних. Ми можете створювати резервні копії вручну в разі потреби, чи робити це автоматично.

2.1.6. Вимоги до складу і параметрів технічних засобів

Всі складові вебсайту повинні взаємодіяти з WordPress та бути повністю сумісними між собою та з хостингом. WordPress — це спосіб створення веб-сайтів у модульній екосистемі з відкритим кодом. Вона відома як система керування вмістом (CMS), що означає, що ви отримуєте інтуїтивно зрозумілий спосіб створювати та керувати своїми публікаціями та сторінками вашого веб-сайту.

Він поставляється з повним набором функцій, але його також можна розширити. Є два способи зробити це:

Теми – вони визначають вигляд вашого сайту, хоча деякі також надають додаткові функції.

Плагіни – так ви розширюєте функціональність свого сайту, а екосистема плагінів величезна. Ми зможемо знайти плагін для будь-якого практичного застосування та, можливо, знадобиться.

Оскільки WordPress є платформою з відкритим вихідним кодом, ви також можете розвивати свій сайт, використовуючи будь-які навички кодування, які у вас є. Іншими словами, його повний код доступний для перегляду та завантаження безкоштовно, а співавтори з усього світу продовжують вдосконалювати пропозицію.

Кожен аспект WordPress простий у використанні. Так, основи WordPress легко освоїти. Однак є багато глибших аспектів WordPress, які потребують роздумів. Це не складна платформа для вивчення та використання сама по собі, але вона потужна з відповідною кодовою базою.

WordPress.com і WordPress.org (тобто саморозміщений WordPress) — це не одне й те саме. Перше є комерційним підприємством Automattic. Хоча існує багато перехресних рішень, останнє є рішенням з відкритим вихідним кодом із широкою спільнотою, яка допомагає розвивати платформу.

Переваги використання WordPress. У використанні платформи WordPress для публікації веб-сайту є багато позитивних моментів. WordPress має значні переваги в використанні та розробці за допомогою його:

- Публікація сайту рибгосподарства не займе багато зусиль.
- WordPress є безпечним і продовжує впроваджувати сучасні технології та стандарти для публікації в Інтернеті.
- Екосистема величезна, з тисячами тем і корисних плагінів, доступних для завантаження.

- Спільнота також величезна, і є багато розробників, які допоможуть майже з усіма аспектами використання платформи.
- Ми маємо майже необмежені можливості для використання WordPress і створення нашого сайту. Він повністю масштабований для проекту будь-якого розміру та використовується деякими з найбільших підприємств у світі.

2.1.7. Вимоги до інформаційної та програмної сумісності

Якщо ми будемо використовувати WordPress, багато хостинг-провайдерів пропонують керований хостинг, який буде автоматично оновлюватись.

Для ядра WordPress, тем і плагінів ви можете вибрати вбудовану опцію WordPress, яка вмикає автоматичне оновлення.

Як увімкнути автоматичне оновлення на WordPress

Щоб увімкнути автоматичне оновлення основного програмного забезпечення WordPress, виконайте наведені нижче дії.

Крок 1. Перейдіть до Інформаційна панель > Оновлення.

Крок 2. Натисніть посилання Увімкнути автоматичні оновлення для всіх нових версій WordPress.

Увімкніть автоматичні оновлення для всіх нових версій WordPress за посиланням на інформаційній панелі

Щоб увімкнути автоматичне оновлення плагінів WordPress, виконайте наведені нижче дії.

Крок 1. Перейдіть до Інформаційна панель > Плагіни > Установлені плагіни.

Крок 2. Там ви побачите всі плагіни, які ви наразі встановили.

Крок 3. Клацніть посилання Увімкнути автоматичні оновлення, розташоване поруч із кожним плагіном.

Так само, щоб увімкнути автоматичні оновлення для вашої теми WordPress, виконайте наведені нижче дії.

Крок 1. Перейдіть до Інформаційна панель > Вигляд > Теми.

Крок 2. Виберіть тему (якщо на вашому веб-сайті встановлено декілька тем, оновіть їх усі).

Крок 3. Потім клацніть посилання, яке відображається як Увімкнути автоматичні оновлення.

Увімкнути посилання на автоматичне оновлення на сторінці теми

Ми також повинні бути обережними, перш ніж встановлювати чи інтегрувати сторонні розширення чи служби. Перш ніж це робити, ми перевіримо відгуки або авторизацію розробника щоб уникнути ризиків, що плагін може стати бекдором для хакерів.

2.1.8. Вимоги до програмної документації

Програмна документація може складати наступну інформацію. Функціональні вимоги розбивають системні функції та функції, які дозволяють системі працювати належним чином.

Слід використовувати огляд як довідковий матеріал, щоб переконатися, що наші вимоги відповідають основним потребам користувача під час роботи з модулями. Існують тисячі функціональних вимог, які слід включити залежно від вашого продукту. Деякі з найпоширеніших:

- Якщо/тоді поведінка
- Логіка обробки даних
- Системні робочі процеси
- Обробка транзакцій
- Адміністративні функції
- Регуляторні потреби та вимоги відповідності

- Вимоги до продуктивності
- Деталі операцій, які виконуються для кожного екрана

Чим більше деталей ми можемо включити у свій документ SRS, тим ефективніше буде наша робота та тим менше буде доопрацювань вже під час використання сервісу.

2.2. Створення сховища в GitHub

На GitHub.com ми повинні перейти на головну сторінку сховища. Над списком файлів потрібно обрати спадне меню «Додати файл» і натиснути «Завантажити файли». Крім того, ми можемо перетягнути файли у наш браузер. Щоб вибрати файли, які потрібно завантажити, перетягніть файл або папку або клацніть «Вибрати файли».

2.2.1. Розподіл ролей виконавців

На нашому проєкті сайту рибгосподарства буде тільки один девелопер та дизайнер. Тому в нас немає потреби в багатьох ролях, для розробки сайту.

2.2.2. GitFlow

Gitflow — це альтернативна модель розгалуження Git, яка передбачає використання гілок функцій і кількох основних гілок. Його вперше опублікував і зробив популярним Вінсент Дріссен у nvie. Порівняно з магістральною розробкою, Gitflow має численні, довготривалі гілки та більші коміти. За цією моделлю розробники створюють гілку функції та відкладають її об'єднання з гілкою основного магістралю, доки функція не буде завершена. Ці довгоживучі

гілки функції вимагають більшої співпраці для злиття та мають вищий ризик відхилення від магістральної гілки. Вони також можуть вводити суперечливі оновлення.

Gitflow можна використовувати для проектів із запланованим циклом випуску та для найкращих практик безперервної доставки DevOps. Цей робочий процес не додає жодних нових концепцій або команд, окрім того, що вимагається для робочого процесу Feature Branch. Натомість він призначає дуже конкретні ролі різним галузям і визначає, як і коли вони повинні взаємодіяти. Окрім гілок функцій, він використовує окремі гілки для підготовки, підтримки та запису випусків. Звичайно, ви також можете використовувати всі переваги робочого процесу Feature Branch: запити на отримання, ізольовані експерименти та ефективнішу співпрацю.

Gitflow — це лише абстрактна ідея робочого процесу Git. Це означає, що він визначає, які гілки створювати та як їх об'єднувати. Ми торкнемося цілей гілок нижче. Набір інструментів git-flow — це фактичний інструмент командного рядка, який має процес встановлення. Процес встановлення git-flow простий. Пакети для git-flow доступні в кількох операційних системах. У системах OSX ви можете виконати `brew install git-flow`. У Windows вам потрібно буде завантажити та встановити git-flow. Після встановлення git-flow ви можете використовувати його у своєму проекті, виконавши `git flow init`. Git-flow — це оболонка Git. Команда `git flow init` є розширенням типової команди `git init` і не змінює нічого у вашому сховищі, окрім створення для вас гілок.

2.3. Створення таблиць в БД

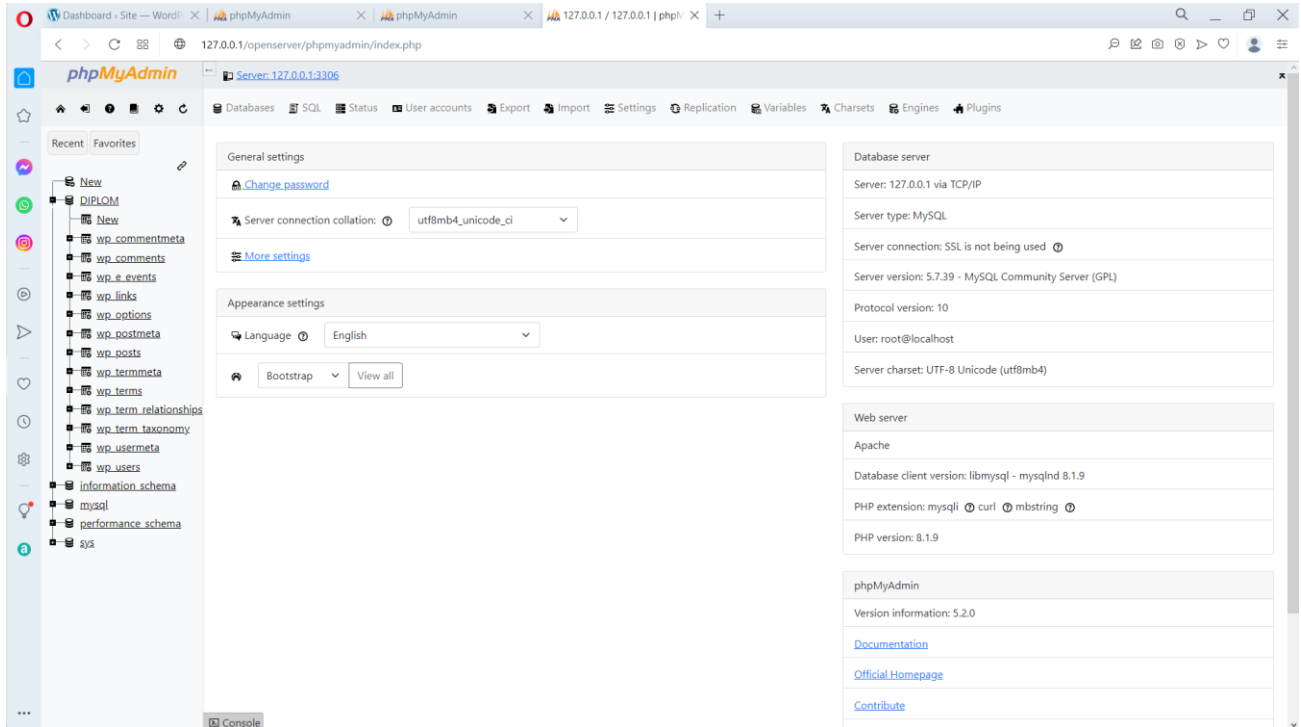


Рис. 2.2 Створення таблиць в базі даних

Ми можемо створити будь-які таблиці в нашій базі даних, які допоможуть нам сортувати матеріал на сайті. Це дозволить надати найкращий сервіс нашим користувачам.

2.3.1. Приклад створення таблиці в БД

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	link_id	bigint(20)		UNSIGNED	No	None	AUTO_INCREMENT		More
2	link_url	varchar(255)	utf8mb4_unicode_520_ci		No				More
3	link_name	varchar(255)	utf8mb4_unicode_520_ci		No				More
4	link_image	varchar(255)	utf8mb4_unicode_520_ci		No				More
5	link_target	varchar(25)	utf8mb4_unicode_520_ci		No				More
6	link_description	varchar(255)	utf8mb4_unicode_520_ci		No				More
7	link_visible	varchar(20)	utf8mb4_unicode_520_ci		No	Y			More
8	link_owner	bigint(20)		UNSIGNED	No	1			More
9	link_rating	int(11)			No	0			More
10	link_updated	datetime			No	0000-00-00 00:00:00			More
11	link_rel	varchar(255)	utf8mb4_unicode_520_ci		No				More
12	link_notes	mediumtext	utf8mb4_unicode_520_ci		No	None			More
13	link_rss	varchar(255)	utf8mb4_unicode_520_ci		No				More

Рис. 2.3 Приклад створення таблиці «Посилання» на нашому сайті

Таблиця «Посилання» має наступні стовпчики: Name, Type, Collation, Attributes, Null, Default, Comments, Extra, та Action. Ми створили наступні рядки:

- link_id bigint(20). Цей параметр дозволяє нам мати унікальний ідентифікатор для кожного запису.
- link_url varchar(255). Це параметр надає нам можливість мати унікальне посилання на цей запис.
- link_name varchar(255). Цей параметр дозволяє нам створити унікальне ім'я для запису.
- link_image varchar(255). Цей параметр дозволяє мати зображення до цього запису.
- link_target varchar(25). Дозволяє вибрати як цей запис буде взаємодіяти з іншими посиланнями.
- link_description varchar(255). Ми зможемо додати опис до цього запису, який нам допоможе пригадати для чого ми його створили.

- link_visible Index varchar(20). Ми зможемо вирішити чи буде цей запис доступний для всіх, чи він буде мати обмежену доступність.
- link_owner bigint(20). Прив'язує цей запис до користувача, який її створив.
- link_rating int(11). Це внутрішній службовий параметр.
- link_updated. Цей параметр записує коли в останній раз запис був оновлений.
- link_rel varchar(255). Це внутрішній службовий параметр.
- link_notes. Ми зможемо додавати нотатки до цього запису.
- link_rss. Дозволить нам використовувати RSS параметри для розповсюдження запису.

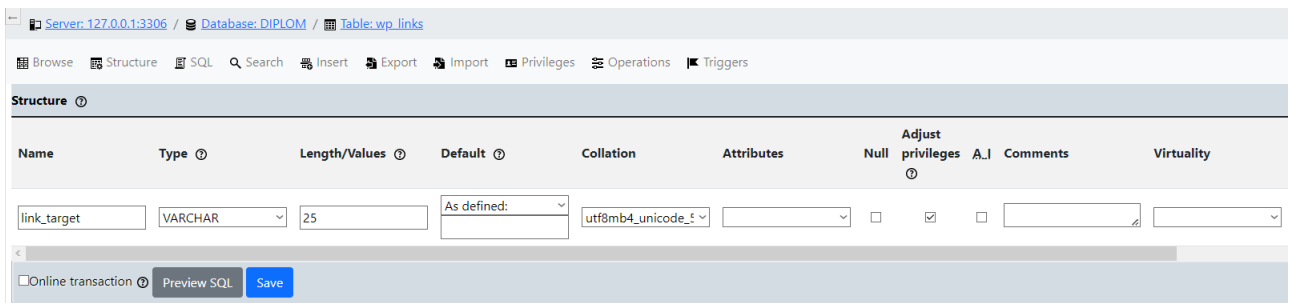


Рис. 2.4 Параметри рядку таблиці «Посилання»

Ми можемо змінити деякі параметри чи додати коменти для подальшого використання.

2.3.2. Схема БД

The screenshot shows the phpMyAdmin interface with the 'wp_postmeta' table selected. The table structure is displayed as follows:

meta_id	post_id	meta_key	meta_value
1	2	_wp_page_template	default
2	3	_wp_page_template	default
3	6	_elementor_edit_mode	builder
4	6	_elementor_template_type	kit
5	7	_wp_attached_file	2023/02/log_file_2023-02-02_09-53-55.txt
6	700	_wp_attached_file	2012/01/desert-road.jpg
7	700	_wp_attachment_metadata	a:6:{s:5:"width";i:980;s:6:"height";i:659;s:4:"fil...
8	700	_filesize	182747
9	700	_filesmime_type	image/jpeg
10	700	_jetpack_related_posts_cache	a:1:{s:32:"445a927db38e88276a96c16f8c13e94";a:2{...
11	700	_tiny_compress_images	a:1:{i:0;a:4:{s:5:"start";i:1454922231;s:5:"input"...
12	700	_vp_views_count	120
14	962	_wp_attached_file	2011/11/photodune-1107867-city-bike-m.jpg
15	962	_wp_attachment_metadata	a:6:{s:5:"width";i:1732;s:6:"height";i:1155;s:4:"f...
16	962	_tiny_compress_images	a:1:{i:0;a:4:{s:5:"start";i:1454921995;s:5:"input"...
17	962	_filesize	347399
18	962	_filesmime_type	image/jpeg
20	2765	_wp_attached_file	2014/01/unsplash_523ae1f5502d6_1-scaled-1.jpg
21	2765	_wp_attachment_metadata	a:6:{s:5:"width";i:2560;s:6:"height";i:1707;s:4:"f...
22	2765	_wp_attachment_backup_sizes	a:2:{s:14:"resized-100x75";a:4:{s:4:"file";s:35:"u...
23	2765	_tiny_compress_images	a:15:{i:0;a:4:{s:5:"start";i:1454921534;s:5:"input"...

Рис. 2.5 Схема бази даних сайт рибгосподарства

Наша база даних підтримує стандарт SQL data base таким чином вся інформація структурована по відповідних стовпчиках та має ключі доступу. Це дозволяє проводити пошук в ній швидко.

2.3.3. Список таблиць

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> wp_commentmeta	☆ 🗑️ 🔍 📄 📁 📂	2	InnoDB	utf8mb4_unicode_520_ci	48.0 KiB	-
<input type="checkbox"/> wp_comments	☆ 🗑️ 🔍 📄 📁 📂	7	InnoDB	utf8mb4_unicode_520_ci	96.0 KiB	-
<input type="checkbox"/> wp_e_events	☆ 🗑️ 🔍 📄 📁 📂	2	InnoDB	utf8mb4_unicode_520_ci	32.0 KiB	-
<input type="checkbox"/> wp_links	☆ 🗑️ 🔍 📄 📁 📂	0	InnoDB	utf8mb4_unicode_520_ci	32.0 KiB	-
<input type="checkbox"/> wp_options	☆ 🗑️ 🔍 📄 📁 📂	173	InnoDB	utf8mb4_unicode_520_ci	2.1 MiB	-
<input type="checkbox"/> wp_postmeta	☆ 🗑️ 🔍 📄 📁 📂	1,076	InnoDB	utf8mb4_unicode_520_ci	2.7 MiB	-
<input type="checkbox"/> wp_posts	☆ 🗑️ 🔍 📄 📁 📂	151	InnoDB	utf8mb4_unicode_520_ci	272.0 KiB	-
<input type="checkbox"/> wp_termmeta	☆ 🗑️ 🔍 📄 📁 📂	0	InnoDB	utf8mb4_unicode_520_ci	48.0 KiB	-
<input type="checkbox"/> wp_terms	☆ 🗑️ 🔍 📄 📁 📂	38	InnoDB	utf8mb4_unicode_520_ci	48.0 KiB	-
<input type="checkbox"/> wp_term_relationships	☆ 🗑️ 🔍 📄 📁 📂	45	InnoDB	utf8mb4_unicode_520_ci	32.0 KiB	-
<input type="checkbox"/> wp_term_taxonomy	☆ 🗑️ 🔍 📄 📁 📂	38	InnoDB	utf8mb4_unicode_520_ci	48.0 KiB	-
<input type="checkbox"/> wp_usermeta	☆ 🗑️ 🔍 📄 📁 📂	20	InnoDB	utf8mb4_unicode_520_ci	48.0 KiB	-
<input type="checkbox"/> wp_users	☆ 🗑️ 🔍 📄 📁 📂	1	InnoDB	utf8mb4_unicode_520_ci	64.0 KiB	-
13 tables	Sum	1,553	InnoDB	utf8_general_ci	5.5 MiB	0 B

Рис. 2.5 Список таблиць сайту

База даних складається з таблиць які використовуються для функціонування модулів коментарії, події, посилання, параметри, постмета та пости, умови та умовимета, взаємодія поміж умовами, умови таксономі, інформація користувачів, та перелік самих користувачів. До кожної з таблиць ми можемо застосувати дії які перелічені в стовпчику Action.

2.4. Архітектурна схема сайту

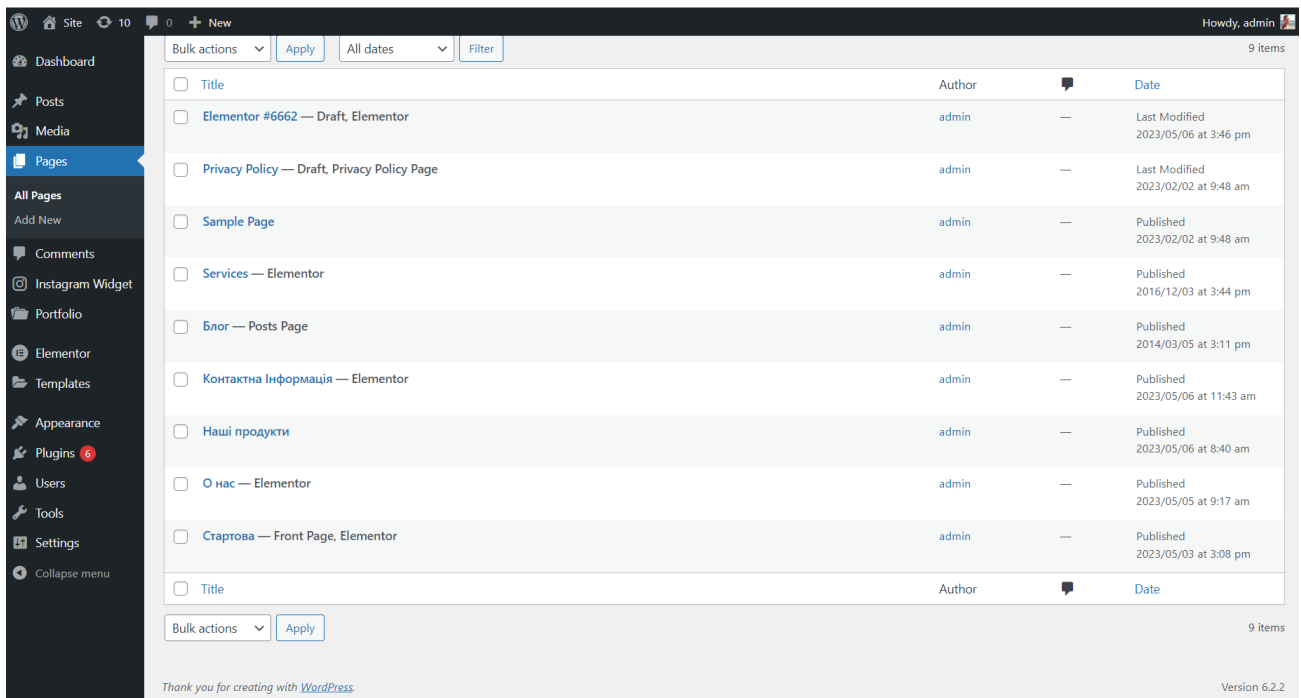


Рис. 2.6. Структура сайту

Сайт включає в себе головну сторінку на яку попадає будь-який користувач. Сторінка «Наші продукти» допоможуть потенційним замовникам розібратись в сортах риби яку ми вирощуємо. В розділі «Блог» ми публікуємо статті які допоможуть сайту бути вище на пошукових сайтах як Гугл. «Контактна інформація» допоможе потенційним клієнтам розібратись де ми знаходимось та який наш графік роботи. Розділ «О нас» розповість клієнтам про то як розвивався наш рибний бізнес та які ключові показники ми змогли досягти.

2.4.1. Алгоритм створення бази даних

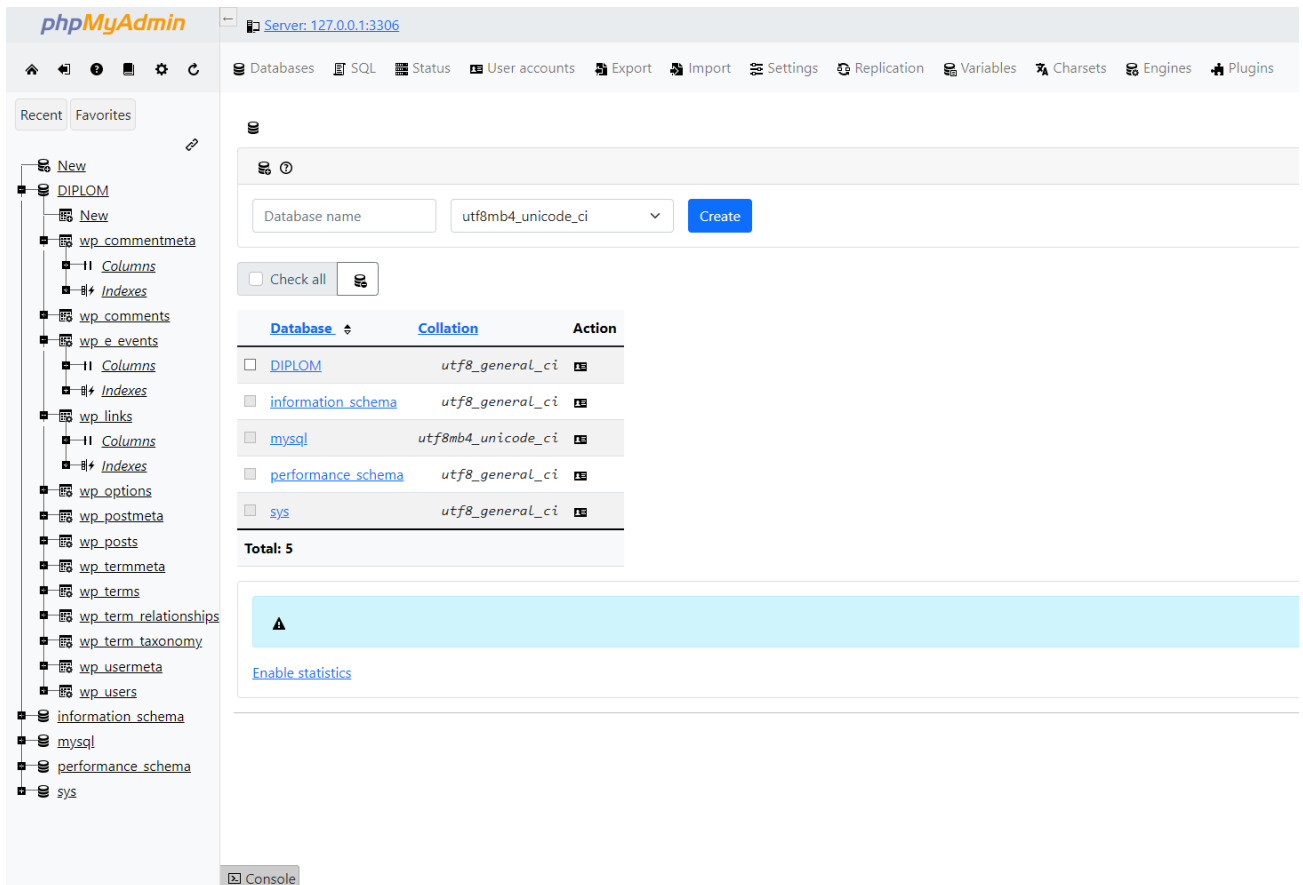
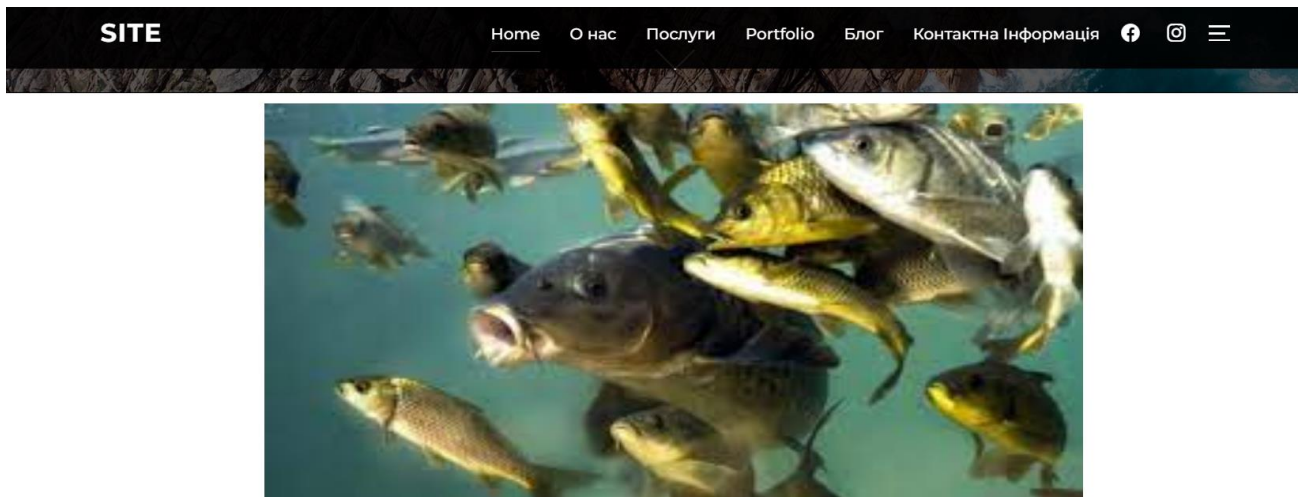


Рис. 2.7 Бази даних

Ми можемо створити іншу базу даних якщо в нас в цьому буде потреба. В даному випадку ми плануємо робити бекап бази даних кожний місяць для того, зоб зберегти інформацію на сайті.

2.4.2 Відображення сторінки



КОМПАНІЯ "РИБ СТАВОК" ВАШ НАДІЙНИЙ ПОСТАЧАЛЬНИК СВІЖОЇ РИБИ

Ми створили наше товариство на базі ставків Дніпропетровської області в 2003 році. Це дозволило нам вийти в трійцю постачальників свіжої риби в Дніпропетровській, Запорізькій, Харківській, та

Рис. 2.8 Стартова сторінка сайту рибгосподарства «РИБ СТАВОК»

Поточна структура сайту надає максимум інформації о продукції господарства, її якості та умов роботи з клієнтами.

2.4.3. Опис розробленого сайту і його інтерфейсу

Сайт «Риб Ставок» був розроблений за допомогою засобів РНР та надає користувачу можливість дізнатись більше про компанію, а саме про сорти риби, яка вирощується, контактну інформацію, публікації, та інше. Користувачі мають змогу робити репост публікацій та залишати коментарі під ними.

ВИСНОВКИ

Підводячи підсумки ми би хотіли акцентувати увагу на наступних результатах, яких вдалось досягнути:

- Ми змогли дослідити технічне підґрунтя нашої задачі та обрати найкращі програмні продукти для його вирішення.
- Нами було розроблено технічне завдання на розробку сайту.
- Ми провели розробку сайту засобами PHP.
- Ми змогли налаштувати всі модулі сайту для стабільного його функціонування.
- Нами були підготовлені всі сторінки сайту для подальшої оптимізації.
- Сайт було максимально налагоджено під вимоги рибгосподарства.
- Сайт буде автоматично оновлюватись кожного разу коли нова версія ключових модулів буде з'являтися в інтернеті.
- Нами були розроблені по політиці бекапів на випадок коли зловмисники можуть отримати доступ до сайту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Backend Languages: The First Step for Becoming a Backend Web Developer [Електрон. ресурс]. – Спосіб доступу: URL: <https://www.scaler.com/topics/software-engineering/backend-languages/>
2. Build fast, responsive sites with Bootstrap [Електрон. ресурс]. – Спосіб доступу: URL: <https://getbootstrap.com/>
3. CSS. [Електрон. ресурс]. – Спосіб доступу: URL: <https://en.wikipedia.org/wiki/CSS>
4. Django introduction [Електрон. ресурс]. – Спосіб доступу: URL: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>
5. Front End Web Development [Електрон. ресурс]. – Спосіб доступу: URL: https://en.wikipedia.org/wiki/Front-end_web_development
6. Front-End vs. Back-End: What's the Difference? [Електрон. ресурс]. – Спосіб доступу: URL: <https://www.computerscience.org/bootcamps/resources/frontend-vs-backend/>
7. HTML: HyperText Markup Language [Електрон. ресурс]. – Спосіб доступу: URL: <https://developer.mozilla.org/en-US/docs/Web/HTML>
8. JavaScript [Електрон. ресурс]. – Спосіб доступу: URL: <https://en.wikipedia.org/wiki/JavaScript>
9. MongoDB [Електрон. ресурс]. – Спосіб доступу: URL: <https://www.techtarget.com/searchdatamanagement/definition/MongoDB>
10. SQL vs NoSQL databases [Електрон. ресурс]. – Спосіб доступу: URL: <https://circleci.com/blog/sql-vs-nosql-databases/>
11. Top 20 Best CSS Frameworks for Front-End Developers in 2023 [Електрон. ресурс]. – Спосіб доступу: URL: <https://hackr.io/blog/best-css-frameworks>

12. What Is GitHub? A Beginner's Introduction to GitHub [Електрон. ресурс]. – Спосіб доступу: URL: <https://kinsta.com/knowledgebase/what-is-github/>
13. What is Tailwind CSS? A Beginner's Guide [Електрон. ресурс]. – Спосіб доступу: URL: <https://www.freecodecamp.org/news/what-is-tailwind-css-a-beginners-guide/>

Додаток А. Приклади коду програми

```

<?php
/**
 * Confirms that the activation key that is sent in an email after a user signs
 * up for a new site matches the key for that user and then displays confirmation.
 *
 * @package WordPress
 */

define( 'WP_INSTALLING', true );

/** Sets up the WordPress Environment. */
require __DIR__ . '/wp-load.php';

require __DIR__ . '/wp-blog-header.php';

if ( ! is_multisite() ) {
    wp_redirect( wp_registration_url() );
    die();
}

$valid_error_codes = array( 'already_active', 'blog_taken' );

list( $activate_path ) = explode( '?', wp_unslash( $_SERVER['REQUEST_URI'] ) );
$activate_cookie      = 'wp-activate-' . COOKIEHASH;

$key = "";
$result = null;

if ( isset( $_GET['key'] ) && isset( $_POST['key'] ) && $_GET['key'] !== $_POST['key'] ) {

```

```

        wp_die( __( 'A key value mismatch has been detected. Please follow the link provided in
your activation email.' ), __( 'An error occurred during the activation' ), 400 );
    } elseif ( ! empty( $_GET['key'] ) ) {
        $key = $_GET['key'];
    } elseif ( ! empty( $_POST['key'] ) ) {
        $key = $_POST['key'];
    }

    if ( $key ) {
        $redirect_url = remove_query_arg( 'key' );

        if ( remove_query_arg( false ) !== $redirect_url ) {
            setcookie( $activate_cookie, $key, 0, $activate_path, COOKIE_DOMAIN, is_ssl(),
true );

            wp_safe_redirect( $redirect_url );
            exit;
        } else {
            $result = wpmu_activate_signup( $key );
        }
    }

    if ( null === $result && isset( $_COOKIE[ $activate_cookie ] ) ) {
        $key = $_COOKIE[ $activate_cookie ];
        $result = wpmu_activate_signup( $key );
        setcookie( $activate_cookie, ' ', time() - YEAR_IN_SECONDS, $activate_path,
COOKIE_DOMAIN, is_ssl(), true );
    }

    if ( null === $result || ( is_wp_error( $result ) && 'invalid_key' === $result->get_error_code() ) ) {
        status_header( 404 );
    } elseif ( is_wp_error( $result ) ) {
        $error_code = $result->get_error_code();
    }

```

```

        if ( ! in_array( $error_code, $valid_error_codes, true ) ) {
            status_header( 400 );
        }
    }
}

```

```
nocache_headers();
```

```

if ( is_object( $wp_object_cache ) ) {
    $wp_object_cache->cache_enabled = false;
}

```

```
// Fix for page title.
```

```
$wp_query->is_404 = false;
```

```
/**
```

```
 * Fires before the Site Activation page is loaded.
```

```
 *
```

```
 * @since 3.0.0
```

```
 */
```

```
do_action( 'activate_header' );
```

```
/**
```

```
 * Adds an action hook specific to this page.
```

```
 *
```

```
 * Fires on { @see 'wp_head' }.
```

```
 *
```

```
 * @since MU (3.0.0)
```

```
 */
```

```
function do_activate_header() {
```

```
    /**
```

```
     * Fires before the Site Activation page is loaded.
```

```

*
* Fires on the {@see 'wp_head'} action.
*
* @since 3.0.0
*/
do_action( 'activate_wp_head' );
}
add_action( 'wp_head', 'do_activate_header' );

/**
* Loads styles specific to this page.
*
* @since MU (3.0.0)
*/
function wpmu_activate_stylesheet() {
    ?>
    <style type="text/css">
        .wp-activate-container { width: 90%; margin: 0 auto; }
        .wp-activate-container form { margin-top: 2em; }
        #submit, #key { width: 100%; font-size: 24px; box-sizing: border-box; }
        #language { margin-top: 0.5em; }
        .wp-activate-container .error { background: #f66; color: #333; }
        span.h3 { padding: 0 8px; font-size: 1.3em; font-weight: 600; }
    </style>
    <?php
}
add_action( 'wp_head', 'wpmu_activate_stylesheet' );
add_action( 'wp_head', 'wp_strict_cross_origin_referrer' );
add_filter( 'wp_robots', 'wp_robots_sensitive_page' );

get_header( 'wp-activate' );

```



```
$blog_details = get_blog_details();
```

```
?>
```

```
<div id="signup-content" class="widecolumn">
```

```
    <div class="wp-activate-container">
```

```
        <?php if ( ! $key ) { ?>
```

```
            <h2><?php _e( 'Activation Key Required' ); ?></h2>
```

```
            <form name="activateform" id="activateform" method="post" action="<?php echo  
network_site_url( $blog_details->path . 'wp-activate.php' ); ?>">
```

```
                <p>
```

```
                    <label for="key"><?php _e( 'Activation Key:' ); ?></label>
```

```
                    <br /><input type="text" name="key" id="key" value="" size="50"
```

```
autofocus="autofocus" />
```

```
                </p>
```

```
                <p class="submit">
```

```
                    <input id="submit" type="submit" name="Submit" class="submit"
```

```
value="<?php esc_attr_e( 'Activate' ); ?>" />
```

```
                </p>
```

```
            </form>
```

```
        <?php
```

```
        } else {
```

```
            if ( is_wp_error( $result ) && in_array( $result->get_error_code(),  
$valid_error_codes, true ) ) {
```

```
                $signup = $result->get_error_data();
```

```
                ?>
```

```
                <h2><?php _e( 'Your account is now active!' ); ?></h2>
```

```
                <?php
```

```
                echo '<p class="lead-in">';
```

```
                if ( " === $signup->domain . $signup->path ) {
```

```
                    printf(
```

```

/* translators: 1: Login URL, 2: Username, 3: User email
address, 4: Lost password URL. */
__( 'Your account has been activated. You may now <a
href="%1$s">log in</a> to the site using your chosen username of &#8220;%2$s&#8221;. Please
check your email inbox at %3$s for your password and login instructions. If you do not receive an
email, please check your junk or spam folder. If you still do not receive an email within an hour, you
can <a href="%4$s">reset your password</a>.' ),
network_site_url( $blog_details->path . 'wp-login.php', 'login'
),
$signup->user_login,
$signup->user_email,
wp_lostpassword_url()
);
} else {
printf(
/* translators: 1: Site URL, 2: Username, 3: User email
address, 4: Lost password URL. */
__( 'Your site at %1$s is active. You may now log in to your
site using your chosen username of &#8220;%2$s&#8221;. Please check your email inbox at %3$s
for your password and login instructions. If you do not receive an email, please check your junk or
spam folder. If you still do not receive an email within an hour, you can <a href="%4$s">reset your
password</a>.' ),
sprintf( '<a href="http://%1$s%2$s">%1$s%2$s</a>',
$signup->domain, $blog_details->path ),
$signup->user_login,
$signup->user_email,
wp_lostpassword_url()
);
}
echo '</p>';
} elseif ( null === $result || is_wp_error( $result ) ) {
?>

```

```

<h2><?php _e( 'An error occurred during the activation' ); ?></h2>
<?php if ( is_wp_error( $result ) ) : ?>
    <p><?php echo $result->get_error_message(); ?></p>
<?php endif; ?>
<?php
} else {
    $url = isset( $result['blog_id'] ) ? get_home_url( (int) $result['blog_id'] ) : '';
    $user = get_userdata( (int) $result['user_id'] );
    ?>
<h2><?php _e( 'Your account is now active!' ); ?></h2>

<div id="signup-welcome">
<p><span class="h3"><?php _e( 'Username:' ); ?></span> <?php echo $user-
>user_login; ?></p>
<p><span class="h3"><?php _e( 'Password:' ); ?></span> <?php echo
$result['password']; ?></p>
</div>

<?php
if ( $url && network_home_url( '', 'http' ) !== $url ) :
    switch_to_blog( (int) $result['blog_id'] );
    $login_url = wp_login_url();
    restore_current_blog();
    ?>
<p class="view">
<?php
        /* translators: 1: Site URL, 2: Login URL. */
        printf( __( 'Your account is now activated. <a
href="%1$s">View your site</a> or <a href="%2$s">Log in</a>' ), $url, esc_url( $login_url ) );
    ?>
</p>
<?php else : ?>

```

```

        <p class="view">
        <?php
            printf(
                /* translators: 1: Login URL, 2: Network home URL.
*/
                __( 'Your account is now activated. <a
href="%1$s">Log in</a> or go back to the <a href="%2$s">homepage</a>.' ),
                network_site_url( $blog_details->path . 'wp-login.php',
'login' ),
                network_home_url( $blog_details->path )
            );
        ?>
    </p>
    <?php
    endif;
    }
}
?>
</div>
</div>
<?php
get_footer( 'wp-activate' );
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the web site, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 */

```

```
* * Database settings
* * Secret keys
* * Database table prefix
* * ABSPATH
*
* @link https://wordpress.org/support/article/editing-wp-config-php/
*
* @package WordPress
*/

// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'DIPLOM' );

/** Database username */
define( 'DB_USER', 'root' );

/** Database password */
define( 'DB_PASSWORD', '' );

/** Database hostname */
define( 'DB_HOST', '127.0.0.1' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8mb4' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication unique keys and salts.
 *

```

```

* Change these to different unique phrases! You can generate these using
* the { @link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service }.
*
* You can change these at any point in time to invalidate all existing cookies.
* This will force all users to have to log in again.
*
* @since 2.6.0
*/

define( 'AUTH_KEY',
'Y=tdgHQINM*mWgj3j^f?DX!=.t{sXTSSLGKdU#qsC%JaY:#L$w/9G4|6^J3cp&K' );
define( 'SECURE_AUTH_KEY',
'[AdSLH,SI1o}LB(1K(g98St=)*@uYf9`JXWz!URWWc7Ib_60fxz&24o9G8qvP}pV' );
define( 'LOGGED_IN_KEY',
'A-;[BTtYpW0#xZx[pqCr-
tQ`I_&Kiqd`aQ4Q4B^76JJcba}n5:`gky_(hpJ#G%K;' );
define( 'NONCE_KEY',
'5r!32OK;.7k-MvNCP8ssd8CR
e)9c.9,S9tyO&|eiaMJ8.%HSLISKF$^6DTSVr(Q' );
define( 'AUTH_SALT',
'~m=^)*N,^Foi5.[Kib,jlbp-
Bd;)hL9$:^^[hclv*$j%u6S?:cpNIW~q5DQH p7x' );
define( 'SECURE_AUTH_SALT',
'4=%8g{I8*OI$>(mc2XA1%%raIK&)0CV^E` (Ph/6g0,5SQys@!{8&y)1n(xElv7D3' );
define( 'LOGGED_IN_SALT',
'Kgo2O2I5HNj[<Va}|,L%,C6`C_`_kdl4(Rlo>3,}Z=F@n-[
dvN#=i,hiO5JzV:e' );
define( 'NONCE_SALT',
'_K3^b1(R@vyGNy>u(jRU@rX&?+eaQ6(uUEyjTQr2akBhsp-
Pmi5k<&x:HUi,g1*G' );

/**#@-*/

/**
 * WordPress database table prefix.
 *
 * You can have multiple installations in one database if you give each
 * a unique prefix. Only numbers, letters, and underscores please!

```

```
*/  
$table_prefix = 'wp_';  
  
/**  
 * For developers: WordPress debugging mode.  
 *  
 * Change this to true to enable the display of notices during development.  
 * It is strongly recommended that plugin and theme developers use WP_DEBUG  
 * in their development environments.  
 *  
 * For information on other constants that can be used for debugging,  
 * visit the documentation.  
 *  
 * @link https://wordpress.org/support/article/debugging-in-wordpress/  
 */  
define( 'WP_DEBUG', false );  
define( 'FS_METHOD', 'direct' );  
  
/* Add any custom values between this line and the "stop editing" line. */  
  
/* That's all, stop editing! Happy publishing. */  
  
/** Absolute path to the WordPress directory. */  
if ( ! defined( 'ABSPATH' ) ) {  
    define( 'ABSPATH', __DIR__ . '/' );  
}  
  
/** Sets up WordPress vars and included files. */  
require_once ABSPATH . 'wp-settings.php';
```

```

<?php
/**
 * WordPress User Page
 *
 * Handles authentication, registering, resetting passwords, forgot password,
 * and other user handling.
 *
 * @package WordPress
 */

/** Make sure that the WordPress bootstrap has run before continuing. */
require __DIR__ . '/wp-load.php';

// Redirect to HTTPS login if forced to use SSL.
if ( force_ssl_admin() && ! is_ssl() ) {
    if ( 0 === strpos( $_SERVER['REQUEST_URI'], 'http' ) ) {
        wp_safe_redirect( set_url_scheme( $_SERVER['REQUEST_URI'], 'https' ) );
        exit;
    } else {
        wp_safe_redirect( 'https://' . $_SERVER['HTTP_HOST'] . $_SERVER['REQUEST_URI'] );
        exit;
    }
}

/**
 * Output the login page header.
 *
 * @since 2.1.0
 *
 * @global string $error Login error message set by deprecated pluggable wp_login()
function

```



```

*                               or plugins replacing it.
* @global bool|string $interim_login Whether interim login modal is being displayed. String
'success'
*                               upon successful login.
* @global string   $action   The action that brought the visitor to the login page.
*
* @param string $title Optional. WordPress login Page title to display in the `<title>` element.
*                               Default 'Log In'.
* @param string $message Optional. Message to display in header. Default empty.
* @param WP_Error $wp_error Optional. The error to pass. Default is a WP_Error instance.
*/
function login_header( $title = 'Log In', $message = "", $wp_error = null ) {
    global $error, $interim_login, $action;

    // Don't index any of these forms.
    add_filter( 'wp_robots', 'wp_robots_sensitive_page' );
    add_action( 'login_head', 'wp_strict_cross_origin_referrer' );

    add_action( 'login_head', 'wp_login_viewport_meta' );

    if ( ! is_wp_error( $wp_error ) ) {
        $wp_error = new WP_Error();
    }

    // Shake it!
    $shake_error_codes = array( 'empty_password', 'empty_email', 'invalid_email',
'invalidcombo', 'empty_username', 'invalid_username', 'incorrect_password',
'retrieve_password_email_failure' );
    /**
     * Filters the error codes array for shaking the login form.
     *
     * @since 3.0.0

```

```

*
* @param string[] $shake_error_codes Error codes that shake the login form.
*/
$shake_error_codes = apply_filters( 'shake_error_codes', $shake_error_codes );

if ( ( $shake_error_codes && $wp_error->has_errors() && in_array( $wp_error-
>get_error_code(), $shake_error_codes, true ) ) {
    add_action( 'login_footer', 'wp_shake_js', 12 );
}

$login_title = get_bloginfo( 'name', 'display' );

/* translators: Login screen title. 1: Login screen name, 2: Network or site name. */
$login_title = sprintf( __( '%1$s &lsquo; %2$s &#8212; WordPress' ), $title, $login_title );

if ( wp_is_recovery_mode() ) {
    /* translators: %s: Login screen title. */
    $login_title = sprintf( __( 'Recovery Mode &#8212; %s' ), $login_title );
}

/**
 * Filters the title tag content for login page.
 *
 * @since 4.9.0
 *
 * @param string $login_title The page title, with extra context added.
 * @param string $title      The original page title.
 */
$login_title = apply_filters( 'login_title', $login_title, $title );

?><!DOCTYPE html>
<html <?php language_attributes(); ?>>

```

```

<head>
  <meta http-equiv="Content-Type" content="<?php bloginfo( 'html_type' ); ?>;
charset=<?php bloginfo( 'charset' ); ?>" />
  <title><?php echo $login_title; ?></title>
  <?php

  wp_enqueue_style( 'login' );

  /*
  * Remove all stored post data on logging out.
  * This could be added by add_action('login_head'...) like wp_shake_js(),
  * but maybe better if it's not removable by plugins.
  */
  if ( 'loggedout' === $wp_error->get_error_code() ) {
    ?>
    <script>if("sessionStorage" in window){try{for(var key in
sessionStorage){if(key.indexOf("wp-autosave-")!=-
1){sessionStorage.removeItem(key)}}}catch(e){}};</script>
    <?php
  }

  /**
  * Enqueue scripts and styles for the login page.
  *
  * @since 3.1.0
  */
  do_action( 'login_enqueue_scripts' );

  /**
  * Fires in the login page header after scripts are enqueued.
  *
  * @since 2.1.0

```

```

*/
do_action( 'login_head' );

$login_header_url = __( 'https://wordpress.org/' );

/**
 * Filters link URL of the header logo above login form.
 *
 * @since 2.1.0
 *
 * @param string $login_header_url Login header logo URL.
 */
$login_header_url = apply_filters( 'login_headerurl', $login_header_url );

$login_header_title = "";

/**
 * Filters the title attribute of the header logo above login form.
 *
 * @since 2.1.0
 * @deprecated 5.2.0 Use { @see 'login_headertext' } instead.
 *
 * @param string $login_header_title Login header logo title attribute.
 */
$login_header_title = apply_filters_deprecated(
    'login_headertitle',
    array( $login_header_title ),
    '5.2.0',
    'login_headertext',
    __( 'Usage of the title attribute on the login logo is not recommended for accessibility
reasons. Use the link text instead.' )
);

```

```
$login_header_text = empty( $login_header_title ) ? __( 'Powered by WordPress' ) :
$login_header_title;
```

```
/**
 * Filters the link text of the header logo above the login form.
 *
 * @since 5.2.0
 *
 * @param string $login_header_text The login header logo link text.
 */
$login_header_text = apply_filters( 'login_headertext', $login_header_text );

$classes = array( 'login-action-' . $action, 'wp-core-ui' );

if ( is_rtl() ) {
    $classes[] = 'rtl';
}

if ( $interim_login ) {
    $classes[] = 'interim-login';

    ?>
    <style type="text/css">html{background-color: transparent;}</style>
    <?php

    if ( 'success' === $interim_login ) {
        $classes[] = 'interim-login-success';
    }
}

$classes[] = ' locale-' . sanitize_html_class( strtolower( str_replace( '_', '-', get_locale() ) ) );
```

```

/**
 * Filters the login page body classes.
 *
 * @since 3.5.0
 *
 * @param string[] $classes An array of body classes.
 * @param string $action The action that brought the visitor to the login page.
 */
$classes = apply_filters( 'login_body_class', $classes, $action );

?>
</head>
<body class="login no-js <?php echo esc_attr( implode( ' ', $classes ) ); ?>">
<script type="text/javascript">
    document.body.className = document.body.className.replace('no-js','js');
</script>
<?php
/**
 * Fires in the login page header after the body tag is opened.
 *
 * @since 4.6.0
 */
do_action( 'login_header' );

?>
<div id="login">
    <h1><a href="<?php echo esc_url( $login_header_url ); ?>"><?php echo
$login_header_text; ?></a></h1>
<?php
/**
 * Filters the message to display above the login form.

```

```

*
* @since 2.1.0
*
* @param string $message Login message text.
*/
$message = apply_filters( 'login_message', $message );

if ( ! empty( $message ) ) {
    echo $message . "\n";
}

// In case a plugin uses $error rather than the $wp_errors object.
if ( ! empty( $error ) ) {
    $wp_error->add( 'error', $error );
    unset( $error );
}

if ( $wp_error->has_errors() ) {
    $errors = "";
    $messages = "";

    foreach ( $wp_error->get_error_codes() as $code ) {
        $severity = $wp_error->get_error_data( $code );
        foreach ( $wp_error->get_error_messages( $code ) as $error_message ) {
            if ( 'message' === $severity ) {
                $messages .= ' ' . $error_message . "<br />\n";
            } else {
                $errors .= ' ' . $error_message . "<br />\n";
            }
        }
    }
}

```

```

if ( ! empty( $errors ) ) {
    /**
     * Filters the error messages displayed above the login form.
     *
     * @since 2.1.0
     *
     * @param string $errors Login error message.
     */
    echo '<div id="login_error">' . apply_filters( 'login_errors', $errors ) .
"</div>\n";
}

if ( ! empty( $messages ) ) {
    /**
     * Filters instructional messages displayed above the login form.
     *
     * @since 2.5.0
     *
     * @param string $messages Login messages.
     */
    echo '<p class="message" id="login-message">' . apply_filters(
'login_messages', $messages ) . "</p>\n";
}
}

} // End of login_header().

/**
 * Outputs the footer for the login page.
 *
 * @since 3.1.0
 *

```



```

* @global bool|string $interim_login Whether interim login modal is being displayed. String
'success'
*
        upon successful login.
*
* @param string $input_id Which input to auto-focus.
*/
function login_footer( $input_id = " ) {
    global $interim_login;

    // Don't allow interim logins to navigate away from the page.
    if ( ! $interim_login ) {
        ?>
        <p id="backtoblog">
            <?php
                $html_link = sprintf(
                    '<a href="%s">%s</a>',
                    esc_url( home_url( '/' ) ),
                    sprintf(
                        /* translators: %s: Site title. */
                        _x( '&larr; Go to %s', 'site' ),
                        get_bloginfo( 'title', 'display' )
                    )
                )
            );
        /**
         * Filter the "Go to site" link displayed in the login page footer.
         *
         * @since 5.7.0
         *
         * @param string $link HTML link to the home URL of the current site.
         */
        echo apply_filters( 'login_site_html_link', $html_link );
        ?>

```

```

</p>
<?php

    the_privacy_policy_link( '<div class="privacy-policy-page-link">', '</div>' );
}

?>
</div><?php // End of <div id="login">. ?>

<?php
if (
    ! $interim_login &&
    /**
     * Filters the Languages select input activation on the login screen.
     *
     * @since 5.9.0
     *
     * @param bool Whether to display the Languages select input on the login screen.
     */
    apply_filters( 'login_display_language_dropdown', true )
) {
    $languages = get_available_languages();

    if ( ! empty( $languages ) ) {
        ?>
        <div class="language-switcher">
            <form id="language-switcher" action="" method="get">

                <label for="language-switcher-locales">
                    <span class="dashicons dashicons-translation" aria-
hidden="true"></span>

                    <span class="screen-reader-text">

```

```

        <?php
            /* translators: Hidden accessibility text. */
            _e( 'Language' );
        ?>
    </span>
</label>

<?php
$args = array(
    'id'          => 'language-switcher-locales',
    'name'        => 'wp_lang',
    'selected'    => determine_locale(),
    'show_available_translations' => false,
    'explicit_option_en_us' => true,
    'languages'  => $languages,
);

/**
 * Filters default arguments for the Languages select input on
the login screen.
 *
 * The arguments get passed to the wp_dropdown_languages()
function.
 *
 * @since 5.9.0
 *
 * @param array $args Arguments for the Languages select
input on the login screen.
 */
wp_dropdown_languages(
    'login_language_dropdown_args', $args );
    apply_filters(
?>

```

```

        <?php if ( $interim_login ) { ?>
            <input type="hidden" name="interim-login" value="1"
/>

        <?php } ?>

        <?php if ( isset( $_GET['redirect_to'] ) && " !=="
$_GET['redirect_to'] ) { ?>
            <input type="hidden" name="redirect_to"
value="<?php echo sanitize_url( $_GET['redirect_to'] ); ?>" />
            <?php } ?>

        <?php if ( isset( $_GET['action'] ) && " !==" $_GET['action'] )
{ ?>
            <input type="hidden" name="action" value="<?php
echo esc_attr( $_GET['action'] ); ?>" />
            <?php } ?>

            <input type="submit" class="button" value="<?php
esc_attr_e( 'Change' ); ?>">

        </form>
    </div>

    <?php } ?>
<?php } ?>
<?php

if ( ! empty( $input_id ) ) {
    ?>
    <script type="text/javascript">
    try{ document.getElementById('<?php echo $input_id; ?>').focus();}catch(e){ }
    if(typeof wpOnload==='function')wpOnload();

```

```
        </script>
        <?php
    }

    /**
     * Fires in the login page footer.
     *
     * @since 3.1.0
     */
    do_action( 'login_footer' );

    ?>
    <div class="clear"></div>
    </body>
    </html>
    <?php
}

/**
 * Outputs the JavaScript to handle the form shaking on the login page.
 *
 * @since 3.0.0
 */
function wp_shake_js() {
    ?>
    <script type="text/javascript">
    document.querySelector('form').classList.add('shake');
    </script>
    <?php
}

/**
```

```
* Outputs the viewport meta tag for the login page.
*
* @since 3.7.0
*/
function wp_login_viewport_meta() {
    ?>
    <meta name="viewport" content="width=device-width" />
    <?php
}

/*
* Main part.
*
* Check the request and redirect or display a form based on the current action.
*/

$action = isset( $_REQUEST['action'] ) ? $_REQUEST['action'] : 'login';
$errors = new WP_Error();

if ( isset( $_GET['key'] ) ) {
    $action = 'resetpass';
}

if ( isset( $_GET['checkemail'] ) ) {
    $action = 'checkemail';
}

$default_actions = array(
    'confirm_admin_email',
    'postpass',
    'logout',
    'lostpassword',
```

```

        'retrievepassword',
        'resetpass',
        'rp',
        'register',
        'checkemail',
        'confirmation',
        'login',
        WP_Recovery_Mode_Link_Service::LOGIN_ACTION_ENTERED,
    );

    // Validate action so as to default to the login screen.
    if ( ! in_array( $action, $default_actions, true ) && false === has_filter( 'login_form_' . $action ) ) {
        $action = 'login';
    }

    nocache_headers();

    header( 'Content-Type: ' . get_bloginfo( 'html_type' ) . '; charset=' . get_bloginfo( 'charset' ) );

    if ( defined( 'RELOCATE' ) && RELOCATE ) { // Move flag is set.
        if ( isset( $_SERVER['PATH_INFO'] ) && ( $_SERVER['PATH_INFO'] !==
$_SERVER['PHP_SELF'] ) ) {
            $_SERVER['PHP_SELF'] = str_replace( $_SERVER['PATH_INFO'], '',
$_SERVER['PHP_SELF'] );
        }

        $url = dirname( set_url_scheme( 'http://' . $_SERVER['HTTP_HOST'] .
$_SERVER['PHP_SELF'] ) );

        if ( get_option( 'siteurl' ) !== $url ) {
            update_option( 'siteurl', $url );
        }
    }

```

```

}

// Set a cookie now to see if they are supported by the browser.
$secure = ( 'https' === parse_url( wp_login_url(), PHP_URL_SCHEME ) );
setcookie( TEST_COOKIE, 'WP Cookie check', 0, COOKIEPATH, COOKIE_DOMAIN, $secure );

if ( SITECOOKIEPATH !== COOKIEPATH ) {
    setcookie( TEST_COOKIE, 'WP Cookie check', 0, SITECOOKIEPATH,
COOKIE_DOMAIN, $secure );
}

if ( isset( $_GET['wp_lang'] ) ) {
    setcookie( 'wp_lang', sanitize_text_field( $_GET['wp_lang'] ), 0, COOKIEPATH,
COOKIE_DOMAIN, $secure );
}

/**
 * Fires when the login form is initialized.
 *
 * @since 3.2.0
 */
do_action( 'login_init' );

/**
 * Fires before a specified login form action.
 *
 * The dynamic portion of the hook name, `$action`, refers to the action
 * that brought the visitor to the login form.
 *
 * Possible hook names include:
 *
 * - `login_form_checkemail`

```



```

* - `login_form_confirm_admin_email`
* - `login_form_confirmation`
* - `login_form_entered_recovery_mode`
* - `login_form_login`
* - `login_form_logout`
* - `login_form_lostpassword`
* - `login_form_postpass`
* - `login_form_register`
* - `login_form_resetpass`
* - `login_form_retrievepassword`
* - `login_form_rp`
*
* @since 2.8.0
*/
do_action( "login_form_{ $action }" );

$http_post = ( 'POST' === $_SERVER['REQUEST_METHOD'] );
$interim_login = isset( $_REQUEST['interim-login'] );

/**
 * Filters the separator used between login form navigation links.
 *
 * @since 4.9.0
 *
 * @param string $login_link_separator The separator used between login form navigation links.
 */
$login_link_separator = apply_filters( 'login_link_separator', '|' );

switch ( $action ) {

    case 'confirm_admin_email':
        /*

```

```

* Note that `is_user_logged_in()` will return false immediately after logging in
* as the current user is not set, see wp-includes/pluggable.php.
* However this action runs on a redirect after logging in.
*/
if ( ! is_user_logged_in() ) {
    wp_safe_redirect( wp_login_url() );
    exit;
}

if ( ! empty( $_REQUEST['redirect_to'] ) ) {
    $redirect_to = $_REQUEST['redirect_to'];
} else {
    $redirect_to = admin_url();
}

if ( current_user_can( 'manage_options' ) ) {
    $admin_email = get_option( 'admin_email' );
} else {
    wp_safe_redirect( $redirect_to );
    exit;
}

/**
 * Filters the interval for dismissing the admin email confirmation screen.
 *
 * If `0` (zero) is returned, the "Remind me later" link will not be displayed.
 *
 * @since 5.3.1
 *
 * @param int $interval Interval time (in seconds). Default is 3 days.
 */

```

```

    $remind_interval = (int) apply_filters( 'admin_email_remind_interval', 3 *
DAY_IN_SECONDS );

    if ( ! empty( $_GET['remind_me_later'] ) ) {
        if ( ! wp_verify_nonce( $_GET['remind_me_later'], 'remind_me_later_nonce'
)) {

            wp_safe_redirect( wp_login_url() );
            exit;
        }

        if ( $remind_interval > 0 ) {
            update_option( 'admin_email_lifespan', time() + $remind_interval );
        }

        $redirect_to = add_query_arg( 'admin_email_remind_later', 1, $redirect_to );
        wp_safe_redirect( $redirect_to );
        exit;
    }

    if ( ! empty( $_POST['correct-admin-email'] ) ) {
        if ( ! check_admin_referer( 'confirm_admin_email',
'confirm_admin_email_nonce' ) ) {
            wp_safe_redirect( wp_login_url() );
            exit;
        }

        /**
         * Filters the interval for redirecting the user to the admin email confirmation
screen.
         *
         * If `0` (zero) is returned, the user will not be redirected.
         */
    }

```

```

* @since 5.3.0
*
* @param int $interval Interval time (in seconds). Default is 6 months.
*/
$admin_email_check_interval = (int) apply_filters(
'admin_email_check_interval', 6 * MONTH_IN_SECONDS );

if ( $admin_email_check_interval > 0 ) {
    update_option( 'admin_email_lifespan', time() +
$admin_email_check_interval );
}

wp_safe_redirect( $redirect_to );
exit;
}

login_header( __( 'Confirm your administration email' ), '', $errors );

/**
 * Fires before the admin email confirm form.
 *
 * @since 5.3.0
 *
 * @param WP_Error $errors A `WP_Error` object containing any errors generated
by using invalid
 *
    credentials. Note that the error object may not contain any errors.
 */
do_action( 'admin_email_confirm', $errors );

?>

```

```

        <form class="admin-email-confirm-form" name="admin-email-confirm-form"
action="<?php echo esc_url( site_url( 'wp-login.php?action=confirm_admin_email', 'login_post' ) );
?>" method="post">

        <?php
        /**
         * Fires inside the admin-email-confirm-form form tags, before the hidden
fields.

         *
         * @since 5.3.0
         */
do_action( 'admin_email_confirm_form' );

wp_nonce_field( 'confirm_admin_email', 'confirm_admin_email_nonce' );

?>
<input type="hidden" name="redirect_to" value="<?php echo esc_attr(
$redirect_to ); ?>" />

<h1 class="admin-email__heading">
        <?php _e( 'Administration email verification' ); ?>
</h1>
<p class="admin-email__details">
        <?php _e( 'Please verify that the <strong>administration
email</strong> for this website is still correct.' ); ?>
        <?php

        /* translators: URL to the WordPress help section about admin email.
*/

        $admin_email_help_url = __(
'https://wordpress.org/documentation/article/settings-general-screen/#email-address' );

        $accessibility_text = sprintf(

```

```

        '<span class="screen-reader-text"> %s</span>',
        /* translators: Hidden accessibility text. */
        __( '(opens in a new tab)' )
    );

    printf(
        '<a href="%s" rel="noopener" target="_blank">%s</a>',
        esc_url( $admin_email_help_url ),
        __( 'Why is this important?' ),
        $accessibility_text
    );

?>
</p>
<p class="admin-email__details">
    <?php

        printf(
            /* translators: %s: Admin email address. */
            __( 'Current administration email: %s' ),
            '<strong>' . esc_html( $admin_email ) . '</strong>'
        );

?>
</p>
<p class="admin-email__details">
    <?php _e( 'This email may be different from your personal email
address.' ); ?>
</p>

<div class="admin-email__actions">
    <div class="admin-email__actions-primary">

```

```

<?php

$change_link = admin_url( 'options-general.php' );
$change_link = add_query_arg( 'highlight',
'confirm_admin_email', $change_link );

?>

<a class="button button-large" href="<?php echo esc_url(
$change_link ); ?>"><?php _e( 'Update' ); ?></a>

<input type="submit" name="correct-admin-email"
id="correct-admin-email" class="button button-primary button-large" value="<?php esc_attr_e(
'The email is correct' ); ?>" />

</div>

<?php if ( $remind_interval > 0 ) : ?>

<div class="admin-email__actions-secondary">

<?php

$remind_me_link = wp_login_url( $redirect_to );
$remind_me_link = add_query_arg(
    array(
        'action' => 'confirm_admin_email',
        'remind_me_later' => wp_create_nonce(
'remind_me_later_nonce' ),
    ),
    $remind_me_link
);

?>

<a href="<?php echo esc_url( $remind_me_link );
?>"><?php _e( 'Remind me later' ); ?></a>

</div>

<?php endif; ?>

```

```

        </div>
    </form>

    <?php

    login_footer();
    break;

case 'postpass':
    if ( ! array_key_exists( 'post_password', $_POST ) ) {
        wp_safe_redirect( wp_get_referer() );
        exit;
    }

    require_once ABSPATH . WPINC . '/class-phpass.php';
    $shasher = new PasswordHash( 8, true );

    /**
     * Filters the life span of the post password cookie.
     *
     * By default, the cookie expires 10 days from creation. To turn this
     * into a session cookie, return 0.
     *
     * @since 3.7.0
     *
     * @param int $expires The expiry time, as passed to setcookie().
     */
    $expire = apply_filters( 'post_password_expires', time() + 10 *
DAY_IN_SECONDS );
    $referer = wp_get_referer();

    if ( $referer ) {

```



```

        $secure = ( 'https' === parse_url( $referer, PHP_URL_SCHEME ) );
    } else {
        $secure = false;
    }

    setcookie( 'wp-postpass_' . COOKIEHASH, $hasher->HashPassword( wp_unslash(
$_POST['post_password'] ) ), $expire, COOKIEPATH, COOKIE_DOMAIN, $secure );

    wp_safe_redirect( wp_get_referer() );
    exit;

case 'logout':
    check_admin_referer( 'log-out' );

    $user = wp_get_current_user();

    wp_logout();

    if ( ! empty( $_REQUEST['redirect_to'] ) ) {
        $redirect_to = $_REQUEST['redirect_to'];
        $requested_redirect_to = $redirect_to;
    } else {
        $redirect_to = add_query_arg(
            array(
                'loggedout' => 'true',
                'wp_lang' => get_user_locale( $user ),
            ),
            wp_login_url()
        );

        $requested_redirect_to = "";
    }

```

```

/**
 * Filters the log out redirect URL.
 *
 * @since 4.2.0
 *
 * @param string $redirect_to      The redirect destination URL.
 * @param string $requested_redirect_to The requested redirect destination URL
passed as a parameter.
 * @param WP_User $user           The WP_User object for the user that's logging
out.
 */
$redirect_to = apply_filters( 'logout_redirect', $redirect_to, $requested_redirect_to,
$user );

wp_safe_redirect( $redirect_to );
exit;

case 'lostpassword':
case 'retrievepassword':
    if ( $http_post ) {
        $errors = retrieve_password();

        if ( ! is_wp_error( $errors ) ) {
            $redirect_to = ! empty( $_REQUEST['redirect_to'] ) ?
$_REQUEST['redirect_to'] : 'wp-login.php?checkemail=confirm';
            wp_safe_redirect( $redirect_to );
            exit;
        }
    }

    if ( isset( $_GET['error'] ) ) {

```

```

        if ( 'invalidkey' === $_GET['error'] ) {
            $errors->add( 'invalidkey', __( '<strong>Error:</strong> Your
password reset link appears to be invalid. Please request a new link below.' ) );
        } elseif ( 'expiredkey' === $_GET['error'] ) {
            $errors->add( 'expiredkey', __( '<strong>Error:</strong> Your
password reset link has expired. Please request a new link below.' ) );
        }
    }
}

```

```

$lostpassword_redirect = ! empty( $_REQUEST['redirect_to'] ) ?
$_REQUEST['redirect_to'] : "";

```

```

/**
 * Filters the URL redirected to after submitting the lostpassword/retrievepassword
form.

```

```

*
```

```

* @since 3.0.0

```

```

*
```

```

* @param string $lostpassword_redirect The redirect destination URL.

```

```

*/

```

```

$redirect_to = apply_filters( 'lostpassword_redirect', $lostpassword_redirect );

```

```

/**

```

```

* Fires before the lost password form.

```

```

*
```

```

* @since 1.5.1

```

```

* @since 5.1.0 Added the ` $errors ` parameter.

```

```

*
```

```

* @param WP_Error $errors A `WP_Error` object containing any errors generated
by using invalid

```

```

*          credentials. Note that the error object may not contain any errors.

```

```

*/

```

```

do_action( 'lost_password', $errors );

```

```
login_header( __( 'Lost Password' ), '<p class="message">' . __( 'Please enter your
username or email address. You will receive an email message with instructions on how to reset
your password.' ) . '</p>', $errors );
```

```
$user_login = "";
```

```
if ( isset( $_POST['user_login'] ) && is_string( $_POST['user_login'] ) ) {
    $user_login = wp_unslash( $_POST['user_login'] );
}
```

```
?>
```

```
<form name="lostpasswordform" id="lostpasswordform" action="<?php echo
esc_url( network_site_url( 'wp-login.php?action=lostpassword', 'login_post' ) ); ?>" method="post">
```

```
<p>
```

```
<label for="user_login"><?php _e( 'Username or Email Address' );
```

```
?></label>
```

```
<input type="text" name="user_login" id="user_login" class="input"
value="<?php echo esc_attr( $user_login ); ?>" size="20" autocapitalize="off"
autocomplete="username" />
```

```
</p>
```

```
<?php
```

```
/**
```

```
* Fires inside the lostpassword form tags, before the hidden fields.
```

```
*
```

```
* @since 2.1.0
```

```
*/
```

```
do_action( 'lostpassword_form' );
```

```
?>
```

```

        <input type="hidden" name="redirect_to" value="<?php echo esc_attr(
$redirect_to ); ?>" />
        <p class="submit">
            <input type="submit" name="wp-submit" id="wp-submit"
class="button button-primary button-large" value="<?php esc_attr_e( 'Get New Password' ); ?>" />
        </p>
    </form>

    <p id="nav">
        <a href="<?php echo esc_url( wp_login_url() ); ?>"><?php _e( 'Log in' );
?></a>

        <?php

        if ( get_option( 'users_can_register' ) ) {
            $registration_url = sprintf( '<a href="%s">%s</a>', esc_url(
wp_registration_url() ), __( 'Register' ) );

            echo esc_html( $login_link_separator );

            /** This filter is documented in wp-includes/general-template.php */
            echo apply_filters( 'register', $registration_url );
        }

        ?>
    </p>
    <?php

    login_footer( 'user_login' );
    break;

case 'resetpass':
case 'rp':

```

```

list( $rp_path ) = explode( '?', wp_unslash( $_SERVER['REQUEST_URI'] ) );
$rp_cookie      = 'wp-resetpass-' . COOKIEHASH;

if ( isset( $_GET['key'] ) && isset( $_GET['login'] ) ) {
    $value = sprintf( '%s:%s', wp_unslash( $_GET['login'] ), wp_unslash(
$_GET['key'] ) );
    setcookie( $rp_cookie, $value, 0, $rp_path, COOKIE_DOMAIN, is_ssl(),
true );

    wp_safe_redirect( remove_query_arg( array( 'key', 'login' ) ) );
    exit;
}

if ( isset( $_COOKIE[ $rp_cookie ] ) && 0 < strpos( $_COOKIE[ $rp_cookie ], ':' ) )
{
    list( $rp_login, $rp_key ) = explode( ':', wp_unslash( $_COOKIE[ $rp_cookie
] ), 2 );

    $user = check_password_reset_key( $rp_key, $rp_login );

    if ( isset( $_POST['pass1'] ) && ! hash_equals( $rp_key, $_POST['rp_key'] ) )
) {
        $user = false;
    }
} else {
    $user = false;
}

if ( ! $user || is_wp_error( $user ) ) {
    setcookie( $rp_cookie, ' ', time() - YEAR_IN_SECONDS, $rp_path,
COOKIE_DOMAIN, is_ssl(), true );
}

```

```

        if ( $user && $user->get_error_code() === 'expired_key' ) {
            wp_redirect( site_url( 'wp-
login.php?action=lostpassword&error=expiredkey' ) );
        } else {
            wp_redirect( site_url( 'wp-
login.php?action=lostpassword&error=invalidkey' ) );
        }

        exit;
    }

    $errors = new WP_Error();

    // Check if password is one or all empty spaces.
    if ( ! empty( $_POST['pass1'] ) ) {
        $_POST['pass1'] = trim( $_POST['pass1'] );

        if ( empty( $_POST['pass1'] ) ) {
            $errors->add( 'password_reset_empty_space', __( 'The password
cannot be a space or all spaces.' ) );
        }
    }

    // Check if password fields do not match.
    if ( ! empty( $_POST['pass1'] ) && trim( $_POST['pass2'] ) !== $_POST['pass1'] ) {
        $errors->add( 'password_reset_mismatch', __( '<strong>Error:</strong> The
passwords do not match.' ) );
    }

    /**
     * Fires before the password reset procedure is validated.
     */

```

```

* @since 3.5.0
*
* @param WP_Error $errors WP Error object.
* @param WP_User|WP_Error $user WP_User object if the login and reset key
match. WP_Error object otherwise.
*/
do_action( 'validate_password_reset', $errors, $user );

if ( ( ! $errors->has_errors() ) && isset( $_POST['pass1'] ) && ! empty(
$_POST['pass1'] ) ) {
    reset_password( $user, $_POST['pass1'] );
    setcookie( $rp_cookie, ' ', time() - YEAR_IN_SECONDS, $rp_path,
COOKIE_DOMAIN, is_ssl(), true );
    login_header( __( 'Password Reset' ), '<p class="message reset-pass">' . __(
'Your password has been reset.' ) . ' <a href="' . esc_url( wp_login_url() ) . '">' . __( 'Log in' ) .
'</a></p>' );
    login_footer();
    exit;
}

wp_enqueue_script( 'utils' );
wp_enqueue_script( 'user-profile' );

login_header( __( 'Reset Password' ), '<p class="message reset-pass">' . __( 'Enter
your new password below or generate one.' ) . '</p>', $errors );

?>
<form name="resetpassform" id="resetpassform" action="<?php echo esc_url(
network_site_url( 'wp-login.php?action=resetpass', 'login_post' ) ); ?>" method="post"
autocomplete="off">
    <input type="hidden" id="user_login" value="<?php echo esc_attr( $rp_login
); ?>" autocomplete="off" />

```



```

<div class="user-pass1-wrap">
  <p>
    <label for="pass1"><?php _e( 'New password' ); ?></label>
  </p>

  <div class="wp-pwd">
    <input type="password" name="pass1" id="pass1"
class="input password-input" size="24" value="" autocomplete="new-password" spellcheck="false"
data-reveal="1" data-pw="<?php echo esc_attr( wp_generate_password( 16 ) ); ?>" aria-
describedby="pass-strength-result" />

    <button type="button" class="button button-secondary wp-
hide-pw hide-if-no-js" data-toggle="0" aria-label="<?php esc_attr_e( 'Hide password' ); ?>">
      <span class="dashicons dashicons-hidden" aria-
hidden="true"></span>
    </button>
    <div id="pass-strength-result" class="hide-if-no-js" aria-
live="polite"><?php _e( 'Strength indicator' ); ?></div>
  </div>
  <div class="pw-weak">
    <input type="checkbox" name="pw_weak" id="pw-weak"
class="pw-checkbox" />
    <label for="pw-weak"><?php _e( 'Confirm use of weak
password' ); ?></label>
  </div>
</div>

<p class="user-pass2-wrap">
  <label for="pass2"><?php _e( 'Confirm new password' ); ?></label>
  <input type="password" name="pass2" id="pass2" class="input"
size="20" value="" autocomplete="new-password" spellcheck="false" />

```

```

</p>

<p class="description indicator-hint"><?php echo wp_get_password_hint();
?></p>

<br class="clear" />

<?php

/**
 * Fires following the 'Strength indicator' meter in the user password reset
form.
 *
 * @since 3.9.0
 *
 * @param WP_User $user User object of the user whose password is being
reset.

 */
do_action( 'resetpass_form', $user );

?>
<input type="hidden" name="rp_key" value="<?php echo esc_attr( $rp_key
); ?>" />

<p class="submit reset-pass-submit">
    <button type="button" class="button wp-generate-pw hide-if-no-js
skip-aria-expanded"><?php _e( 'Generate Password' ); ?></button>
    <input type="submit" name="wp-submit" id="wp-submit"
class="button button-primary button-large" value="<?php esc_attr_e( 'Save Password' ); ?>" />
</p>
</form>

<p id="nav">

```

```

<a href="<?php echo esc_url( wp_login_url() ); ?>"><?php _e( 'Log in' );
?></a>

<?php

if ( get_option( 'users_can_register' ) ) {
    $registration_url = sprintf( '<a href="%s">%s</a>', esc_url(
wp_registration_url() ), __( 'Register' ) );

    echo esc_html( $login_link_separator );

    /** This filter is documented in wp-includes/general-template.php */
    echo apply_filters( 'register', $registration_url );
}

?>
</p>
<?php

login_footer( 'pass1' );
break;

case 'register':
    if ( is_multisite() ) {
        /**
         * Filters the Multisite sign up URL.
         *
         * @since 3.0.0
         *
         * @param string $sign_up_url The sign up URL.
         */
        wp_redirect( apply_filters( 'wp_signup_location', network_site_url( 'wp-
signup.php' ) ) );

```

```

        exit;
    }

    if ( ! get_option( 'users_can_register' ) ) {
        wp_redirect( site_url( 'wp-login.php?registration=disabled' ) );
        exit;
    }

    $user_login = "";
    $user_email = "";

    if ( $http_post ) {
        if ( isset( $_POST['user_login'] ) && is_string( $_POST['user_login'] ) ) {
            $user_login = wp_unslash( $_POST['user_login'] );
        }

        if ( isset( $_POST['user_email'] ) && is_string( $_POST['user_email'] ) ) {
            $user_email = wp_unslash( $_POST['user_email'] );
        }

        $errors = register_new_user( $user_login, $user_email );

        if ( ! is_wp_error( $errors ) ) {
            $redirect_to = ! empty( $_POST['redirect_to'] ) ?
$_POST['redirect_to'] : 'wp-login.php?checkemail=registered';
            wp_safe_redirect( $redirect_to );
            exit;
        }
    }

    $registration_redirect = ! empty( $_REQUEST['redirect_to'] ) ?
$_REQUEST['redirect_to'] : "";

```

```

/**
 * Filters the registration redirect URL.
 *
 * @since 3.0.0
 * @since 5.9.0 Added the `$errors` parameter.
 *
 * @param string $registration_redirect The redirect destination URL.
 * @param int|WP_Error $errors User id if registration was successful,
 *                               WP_Error object otherwise.
 */
$redirect_to = apply_filters( 'registration_redirect', $registration_redirect, $errors );

login_header( __( 'Registration Form' ), '<p class="message register">' . __( 'Register
For This Site' ) . '</p>', $errors );

?>
<form name="registerform" id="registerform" action="<?php echo esc_url( site_url(
'wp-login.php?action=register', 'login_post' ) ); ?>" method="post" novalidate="novalidate">
    <p>
        <label for="user_login"><?php _e( 'Username' ); ?></label>
        <input type="text" name="user_login" id="user_login" class="input"
value="<?php echo esc_attr( wp_unslash( $user_login ) ); ?>" size="20" autocapitalize="off"
autocomplete="username" />
    </p>
    <p>
        <label for="user_email"><?php _e( 'Email' ); ?></label>
        <input type="email" name="user_email" id="user_email"
class="input" value="<?php echo esc_attr( wp_unslash( $user_email ) ); ?>" size="25"
autocomplete="email" />
    </p>
    <?php

```

```

/**
 * Fires following the 'Email' field in the user registration form.
 *
 * @since 2.1.0
 */
do_action( 'register_form' );

?>
<p id="reg_passmail">
    <?php _e( 'Registration confirmation will be emailed to you.' ); ?>
</p>
<br class="clear" />
<input type="hidden" name="redirect_to" value="<?php echo esc_attr(
$redirect_to ); ?>" />
<p class="submit">
    <input type="submit" name="wp-submit" id="wp-submit"
class="button button-primary button-large" value="<?php esc_attr_e( 'Register' ); ?>" />
</p>
</form>

<p id="nav">
    <a href="<?php echo esc_url( wp_login_url() ); ?>"><?php _e( 'Log in' );
?></a>

<?php

echo esc_html( $login_link_separator );

$html_link = sprintf( '<a href="%s">%s</a>', esc_url( wp_lostpassword_url()
), __( 'Lost your password?' ) );

/** This filter is documented in wp-login.php */

```

```

        echo apply_filters( 'lost_password_html_link', $html_link );

        ?>
    </p>
    <?php

    login_footer( 'user_login' );
    break;

case 'checkemail':
    $redirect_to = admin_url();
    $errors      = new WP_Error();

    if ( 'confirm' === $_GET['checkemail'] ) {
        $errors->add(
            'confirm',
            sprintf(
                /* translators: %s: Link to the login page. */
                __( 'Check your email for the confirmation link, then visit the
<a href="%s">login page</a>.' ),
                wp_login_url()
            ),
            'message'
        );
    } elseif ( 'registered' === $_GET['checkemail'] ) {
        $errors->add(
            'registered',
            sprintf(
                /* translators: %s: Link to the login page. */
                __( 'Registration complete. Please check your email, then visit
the <a href="%s">login page</a>.' ),
                wp_login_url()
            )
        );
    }
}

```

```

        ),
        'message'
    );
}

/** This action is documented in wp-login.php */
$errors = apply_filters( 'wp_login_errors', $errors, $redirect_to );

login_header( __( 'Check your email' ), "", $errors );
login_footer();
break;

case 'confirmation':
    if ( ! isset( $_GET['request_id'] ) ) {
        wp_die( __( 'Missing request ID.' ) );
    }

    if ( ! isset( $_GET['confirm_key'] ) ) {
        wp_die( __( 'Missing confirm key.' ) );
    }

    $request_id = (int) $_GET['request_id'];
    $key        = sanitize_text_field( wp_unslash( $_GET['confirm_key'] ) );
    $result     = wp_validate_user_request_key( $request_id, $key );

    if ( is_wp_error( $result ) ) {
        wp_die( $result );
    }

/**
 * Fires an action hook when the account action has been confirmed by the user.
 */

```



```

* Using this you can assume the user has agreed to perform the action by
* clicking on the link in the confirmation email.
*
* After firing this action hook the page will redirect to wp-login a callback
* redirects or exits first.
*
* @since 4.9.6
*
* @param int $request_id Request ID.
*/
do_action( 'user_request_action_confirmed', $request_id );

$message = _wp_privacy_account_request_confirmed_message( $request_id );

login_header( __( 'User action confirmed.' ), $message );
login_footer();
exit;

case 'login':
default:
    $secure_cookie = "";
    $customize_login = isset( $_REQUEST['customize-login'] );

    if ( $customize_login ) {
        wp_enqueue_script( 'customize-base' );
    }

    // If the user wants SSL but the session is not SSL, force a secure cookie.
    if ( ! empty( $_POST['log'] ) && ! force_ssl_admin() ) {
        $user_name = sanitize_user( wp_unslash( $_POST['log'] ) );
        $user = get_user_by( 'login', $user_name );

```

```

if ( ! $user && strpos( $user_name, '@' ) ) {
    $user = get_user_by( 'email', $user_name );
}

if ( $user ) {
    if ( get_user_option( 'use_ssl', $user->ID ) ) {
        $secure_cookie = true;
        force_ssl_admin( true );
    }
}

}

if ( isset( $_REQUEST['redirect_to'] ) ) {
    $redirect_to = $_REQUEST['redirect_to'];
    // Redirect to HTTPS if user wants SSL.
    if ( $secure_cookie && false !== strpos( $redirect_to, 'wp-admin' ) ) {
        $redirect_to = preg_replace( '^http://', 'https://', $redirect_to );
    }
} else {
    $redirect_to = admin_url();
}

$reauth = empty( $_REQUEST['reauth'] ) ? false : true;

$user = wp_signon( array(), $secure_cookie );

if ( empty( $_COOKIE[ LOGGED_IN_COOKIE ] ) ) {
    if ( headers_sent() ) {
        $user = new WP_Error(
            'test_cookie',
            sprintf(

```

```

                /* translators: 1: Browser cookie documentation URL,
2: Support forums URL. */
                __( '<strong>Error:</strong> Cookies are blocked due
to unexpected output. For help, please see <a href="%1$s">this documentation</a> or try the <a
href="%2$s">support forums</a>.' ),
                __(
'https://wordpress.org/documentation/article/cookies/' ),
                __( 'https://wordpress.org/support/forums/' )
                )
            );
        } elseif ( isset( $_POST['testcookie'] ) && empty( $_COOKIE[
TEST_COOKIE ] ) ) {
            // If cookies are disabled, the user can't log in even with a valid
username and password.
            $user = new WP_Error(
                'test_cookie',
                sprintf(
                    /* translators: %s: Browser cookie documentation
URL. */
                    __( '<strong>Error:</strong> Cookies are blocked or
not supported by your browser. You must <a href="%s">enable cookies</a> to use WordPress.' ),
                    __(
'https://wordpress.org/documentation/article/cookies/#enable-cookies-in-your-browser' )
                )
            );
        }
    }

    $requested_redirect_to = isset( $_REQUEST['redirect_to'] ) ?
$_REQUEST['redirect_to'] : '';
    /**
     * Filters the login redirect URL.

```

```

*
* @since 3.0.0
*
* @param string $redirect_to The redirect destination URL.
* @param string $requested_redirect_to The requested redirect destination
URL passed as a parameter.
* @param WP_User|WP_Error $user WP_User object if login was
successful, WP_Error object otherwise.
*/
$redirect_to = apply_filters( 'login_redirect', $redirect_to, $requested_redirect_to,
$user );

if ( ! is_wp_error( $user ) && ! $reauth ) {
    if ( $interim_login ) {
        $message = '<p class="message">'. __( 'You have logged in
successfully.' ) . '</p>';

        $interim_login = 'success';
        login_header( "", $message );

        ?>
    </div>
    <?php

    /** This action is documented in wp-login.php */
    do_action( 'login_footer' );

    if ( $customize_login ) {
        ?>
        <script type="text/javascript">setTimeout( function(){ new
wp.customize.Messenger({ url: '<?php echo wp_customize_url(); ?>', channel: 'login'
}).send('login') }, 1000 );</script>
        <?php

```

```

    }

    ?>
</body></html>
<?php

    exit;
}

// Check if it is time to add a redirect to the admin email confirmation screen.
if ( is_a( $user, 'WP_User' ) && $user->exists() && $user->has_cap(
'manage_options' ) ) {
    $admin_email_lifespan = (int) get_option( 'admin_email_lifespan' );

    /*
     * If `0` (or anything "falsey" as it is cast to int) is returned, the user
will not be redirected
     * to the admin email confirmation screen.
    */
    /** This filter is documented in wp-login.php */
    $admin_email_check_interval = (int) apply_filters(
'manage_options', 6 * MONTH_IN_SECONDS );

    if ( $admin_email_check_interval > 0 && time() >
$admin_email_lifespan ) {
        $redirect_to = add_query_arg(
            array(
                'action' => 'confirm_admin_email',
                'wp_lang' => get_user_locale( $user ),
            ),
            wp_login_url( $redirect_to )
        );
    }
}

```

```

    }
}

if ( ( empty( $redirect_to ) || 'wp-admin/' === $redirect_to || admin_url() ===
$redirect_to ) ) {
    // If the user doesn't belong to a blog, send them to user admin. If the
user can't edit posts, send them to their profile.
    if ( is_multisite() && ! get_active_blog_for_user( $user->ID ) && !
is_super_admin( $user->ID ) ) {
        $redirect_to = user_admin_url();
    } elseif ( is_multisite() && ! $user->has_cap( 'read' ) ) {
        $redirect_to = get_dashboard_url( $user->ID );
    } elseif ( ! $user->has_cap( 'edit_posts' ) ) {
        $redirect_to = $user->has_cap( 'read' ) ? admin_url(
'profile.php' ) : home_url();
    }

    wp_redirect( $redirect_to );
    exit;
}

wp_safe_redirect( $redirect_to );
exit;
}

$errors = $user;
// Clear errors if loggedout is set.
if ( ! empty( $_GET['loggedout'] ) || $reauth ) {
    $errors = new WP_Error();
}

```

```

if ( empty( $_POST ) && $errors->get_error_codes() === array( 'empty_username',
'empty_password' ) ) {
    $errors = new WP_Error( "", "" );
}

if ( $interim_login ) {
    if ( ! $errors->has_errors() ) {
        $errors->add( 'expired', __( 'Your session has expired. Please log in to
continue where you left off.' ), 'message' );
    }
} else {
    // Some parts of this script use the main login form to display a message.
    if ( isset( $_GET['loggedout'] ) && $_GET['loggedout'] ) {
        $errors->add( 'loggedout', __( 'You are now logged out.' ), 'message' );
    } elseif ( isset( $_GET['registration'] ) && 'disabled' ===
$_GET['registration'] ) {
        $errors->add( 'registerdisabled', __( '<strong>Error:</strong> User
registration is currently not allowed.' ) );
    } elseif ( strpos( $redirect_to, 'about.php?updated' ) ) {
        $errors->add( 'updated', __( '<strong>You have successfully updated
WordPress!</strong> Please log back in to see what&#8217;s new.' ), 'message' );
    } elseif ( WP_Recovery_Mode_Link_Service::LOGIN_ACTION_ENTERED
=== $action ) {
        $errors->add( 'enter_recovery_mode', __( 'Recovery Mode Initialized.
Please log in to continue.' ), 'message' );
    } elseif ( isset( $_GET['redirect_to'] ) && false !== strpos(
$_GET['redirect_to'], 'wp-admin/authorize-application.php' ) ) {
        $query_component = wp_parse_url( $_GET['redirect_to'],
PHP_URL_QUERY );
        $query = array();
        if ( $query_component ) {
            parse_str( $query_component, $query );

```

```

    }

    if ( ! empty( $query['app_name'] ) ) {
        /* translators: 1: Website name, 2: Application name. */
        $message = sprintf( 'Please log in to %1$s to authorize %2$s
to connect to your account.', get_bloginfo( 'name', 'display' ), '<strong>' . esc_html(
$query['app_name'] ) . '</strong>' );
    } else {
        /* translators: %s: Website name. */
        $message = sprintf( 'Please log in to %s to proceed with
authorization.', get_bloginfo( 'name', 'display' ) );
    }

    $errors->add( 'authorize_application', $message, 'message' );
}

}

/**
 * Filters the login page errors.
 *
 * @since 3.6.0
 *
 * @param WP_Error $errors WP Error object.
 * @param string   $redirect_to Redirect destination URL.
 */
$errors = apply_filters( 'wp_login_errors', $errors, $redirect_to );

// Clear any stale cookies.
if ( $reauth ) {
    wp_clear_auth_cookie();
}

```



```

login_header( __( 'Log In' ), "", $errors );

if ( isset( $_POST['log'] ) ) {
    $user_login = ( 'incorrect_password' === $errors->get_error_code() ||
'empty_password' === $errors->get_error_code() ) ? esc_attr( wp_unslash( $_POST['log'] ) ) : "";
}

$rememberme = ! empty( $_POST['rememberme'] );

$aria_describedby = "";
$has_errors = $errors->has_errors();

if ( $has_errors ) {
    $aria_describedby = ' aria-describedby="login_error"';
}

if ( $has_errors && 'message' === $errors->get_error_data() ) {
    $aria_describedby = ' aria-describedby="login-message"';
}

wp_enqueue_script( 'user-profile' );
?>

<form name="loginform" id="loginform" action="<?php echo esc_url( site_url( 'wp-
login.php', 'login_post' ) ); ?>" method="post">
    <p>
        <label for="user_login"><?php _e( 'Username or Email Address' );
?></label>
        <input type="text" name="log" id="user_login"<?php echo
$aria_describedby; ?> class="input" value="<?php echo esc_attr( $user_login ); ?>" size="20"
autocapitalize="off" autocomplete="username" />
    </p>

```

```

<div class="user-pass-wrap">
    <label for="user_pass"><?php _e( 'Password' ); ?></label>
    <div class="wp-pwd">
        <input type="password" name="pwd" id="user_pass"<?php
echo $aria_describedby; ?> class="input password-input" value="" size="20"
autocomplete="current-password" spellcheck="false" />
        <button type="button" class="button button-secondary wp-
hide-pw hide-if-no-js" data-toggle="0" aria-label="<?php esc_attr_e( 'Show password' ); ?>">
            <span class="dashicons dashicons-visibility" aria-
hidden="true"></span>
        </button>
    </div>
</div>
<?php

/**
 * Fires following the 'Password' field in the login form.
 *
 * @since 2.1.0
 */
do_action( 'login_form' );

?>
<p class="forgetmenot"><input name="rememberme" type="checkbox"
id="rememberme" value="forever" <?php checked( $rememberme ); ?> /> <label
for="rememberme"><?php esc_html_e( 'Remember Me' ); ?></label></p>
<p class="submit">
    <input type="submit" name="wp-submit" id="wp-submit"
class="button button-primary button-large" value="<?php esc_attr_e( 'Log In' ); ?>" />
    <?php

```

```

        if ( $interim_login ) {
            ?>
            <input type="hidden" name="interim-login" value="1" />
            <?php
        } else {
            ?>
            <input type="hidden" name="redirect_to" value="<?php echo
esc_attr( $redirect_to ); ?>" />
            <?php
        }

        if ( $customize_login ) {
            ?>
            <input type="hidden" name="customize-login" value="1" />
            <?php
        }

        ?>
        <input type="hidden" name="testcookie" value="1" />
    </p>
</form>

<?php

if ( ! $interim_login ) {
    ?>
    <p id="nav">
        <?php

            if ( get_option( 'users_can_register' ) ) {
                $registration_url = sprintf( '<a href="%s">%s</a>', esc_url(
wp_registration_url() ), __( 'Register' ) );

```

```

template.php */
        /** This filter is documented in wp-includes/general-
        echo apply_filters( 'register', $registration_url );

        echo esc_html( $login_link_separator );
    }

    $html_link = sprintf( '<a href="%s">%s</a>', esc_url(
wp_lostpassword_url() ), __( 'Lost your password?' ) );

    /**
     * Filters the link that allows the user to reset the lost password.
     *
     * @since 6.1.0
     *
     * @param string $html_link HTML link to the lost password form.
     */
    echo apply_filters( 'lost_password_html_link', $html_link );

    ?>
</p>
<?php
}

$login_script = 'function wp_attempt_focus() {';
$login_script .= 'setTimeout( function() {';
$login_script .= 'try {';

if ( $user_login ) {
    $login_script .= 'd = document.getElementById( "user_pass" ); d.value = "";' ;
} else {

```

```

$login_script .= 'd = document.getElementById( "user_login" );';

if ( $errors->get_error_code() === 'invalid_username' ) {
    $login_script .= 'd.value = ""';
}
}

$login_script .= 'd.focus(); d.select();';
$login_script .= '} catch( er ) {';
$login_script .= '}, 200);';
$login_script .= "\n"; // End of wp_attempt_focus().

/**
 * Filters whether to print the call to `wp_attempt_focus()` on the login screen.
 *
 * @since 4.8.0
 *
 * @param bool $print Whether to print the function call. Default true.
 */
if ( apply_filters( 'enable_login autofocus', true ) && ! $error ) {
    $login_script .= "wp_attempt_focus();\n";
}

// Run `wpOnload()` if defined.
$login_script .= "if ( typeof wpOnload === 'function' ) { wpOnload() }";

?>
<script type="text/javascript">
    <?php echo $login_script; ?>
</script>
<?php

```

```
if ( $interim_login ) {
    ?>
    <script type="text/javascript">
    ( function() {
        try {
            var i, links = document.getElementsByTagName( 'a' );
            for ( i in links ) {
                if ( links[i].href ) {
                    links[i].target = '_blank';
                    links[i].rel = 'noopener';
                }
            }
        } catch( er ) {}
    }());
    </script>
    <?php
}

login_footer();
break;
} // End action switch.
```

Додаток Б. Відгук керівника кваліфікаційної роботи

Додаток В. Рецензія

Додаток Г. Довідка