Міністерство освіти і науки України

Природничо-гуманітарний фаховий коледж

ДВНЗ «Ужгородський національний університет»

Циклова комісія програмування та інформаційних технологій

##### Реєстраційний номер \_\_\_\_\_\_\_\_\_\_\_

Дата реєстрації **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## Пояснювальна записка

до дипломного проєкту фахового молодшого бакалавра

на тему

**Розробка навчально-пізнавальної платформи**

**«Скандинавский епос»**

Виконав: студент 4 курсу, групи КН-41

спеціальності 121

«Інженерія програмного забезпечення»

**Чорнак Юрій Васильович**

Керівник:

**викл. Лінчук Наталія Леонідівна**

Рецензент:

**доктор технічних наук, професор, декан факультету інформаційних технологій ДВНЗ «УжНУ»**

**Повхан Ігор Федорович**

Ужгород – 2024

Автор проєкту: Чорнак Юрій Васильович

(прізвище, ім’я, по батькові)

Тема дипломного проєкту: Розробка навчально-пізнавальної платформи «Скандинавський епос»

Науковий керівник: викл. Лінчук Наталія Леонідівнавна

(науковий ступінь, посада, прізвище, ім’я, по батькові)

Рецензент: доктор технічних наук, професор, декан факультету інформаційних технологій ДВНЗ «УжНУ» Повхан Ігор Федорович;

(науковий ступінь, посада, прізвище, ім’я, по батькові)

Дата захисту: « »\_\_\_\_\_\_\_\_\_\_\_2024 року

Анотація

Метою дипломного проєкту є розробка вебресурсу, який реалізує платформу навчально-пізнавального призначення та заповнити матеріалами, що розкривають тему скандинавского епосу.

Пояснювальна записка складається з чотирьох розділів.

У першому розділі розглянуто загальні теоретичні відомості, проаналізовано знайдені аналоги, описано предметну область та постановку задачі, дано обґрунтування вибору методу розв’язку задачі.

Другий розділ присвячено практичній частині проєкту.

В економічній частині проведено економічні розрахунки витрат та можливих прибутків, пов’язаних з проєктом.

В розділі «Охорона праці» описано загальні вимоги охорони праці при роботі з електронно-обчислювальними машинами, розраховано кількість комп’ютеризованих робочих місць та сформована інструкція з охорони праці при роботі з ПЕОМ.

Ключові слова: вебресурс, публікації, React, HTML, CSS, JavaScript, Node.js, Express, MySQL, json.

Обсяг пояснювальної записки: 78 сторінок.

Перелік додатків: діаграма сутностей-зв’язків, схема БД, діаграма ієрархії файлів проекту, документація до API , лістинг програмного коду.

Project author: Chornack Yuriy

Project subject: Development of the Educational and Cognitive Platform “Scandinavian”

Supervisor: lect. Linchuck Natalia

Reviewer: senior lecturer of the Department of Informatics and Physical and Mathematical Sciences of the Faculty of Information Technologies of the State Higher Educational Institution "Uzhgorod National University" Matyashovska Beata Oleksandrivna

Date: “\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024

Annotation

The aim of the thesis project is to develop a web resource that implements an educational and cognitive platform and to populate it with materials that explore the theme of the Scandinavian epic.

The explanatory note consists of four sections.

In the first chapter, general theoretical information is considered, analyzed found analogs, the subject area and problem statement are described, and a justification for the choice of problem-solving method is provided.

The second section is devoted to the practical part of the project.

In the economic part, economic calculations of costs and possible profits related to the project were carried out.

In the “Occupational safety” section, the general requirements for occupational safety when working with electronic computers are described, the number of computerized workplaces is calculated, and an occupational safety instruction for personal computer users is created.

Keywords: web resource, publications, React, HTML, CSS, JavaScript, Node.js, Express, MySQL, json.

Volume of the explanatory note: 78 pages.

List of appendices: entity-relationship diagram, database schema, project file hierarchy diagram, API documentation, program code listing.

Перелік умовних позначень, символів, одиниць і термінів

|  |  |
| --- | --- |
| API | Application Programming Interface — інтерфейс програмування додатків. |
| СSS | це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду. |
| HTML | мова розмітки гіпертекстових документів. |
| JavaScript | динамічна, об’єктно-орієнтована прототипна мова програмування. |
| JSON | популярна [JavaScript](https://uk.wikipedia.org/wiki/JavaScript)-бібліотека з [відкритим сирцевим кодом](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%BA%D1%80%D0%B8%D1%82%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F). |
| MySQL | система керування базами даних |
| Node.js  REST API | середовище виконання, що дозволяє виконувати код JavaScript на стороні сервера.  Representational State Transfer API) — архітектурний стиль для розробки мережевих додатків |
| WEB | система доступу до пов’язаних між собою документів на різних комп’ютерах, підключених до Інтернету. |
| ВДТ | відеодисплейний термінал. |
| ЕОМ | електронно-обчислювальна машина. |
| ПЗ | програмне забезпечення. |

ЗМІСТ

[Вступ 8](#_Toc390601152)

[1 Теоретична частина 9](#_Toc390601153)

[1.1 Загальні відомості про використані технології 9](#_Toc390601155)

[1.1.1 Мова розмітки гіпертексту HTML 9](#_Toc390601158)

[1.1.2 Каскадні таблиці стилів CSS 10](#_Toc390601158)

[1.1.3 Скриптова мова програмування JavaScript 11](#_Toc390601158)

[1.1.4 Фреймворк Bootstrap 12](#_Toc390601158)

[1.1.5 Фреймворк jQuery 13](#_Toc390601158)

[1.1.6 Скриптової мови програмування PHP 14](#_Toc390601158)

[1.1.7 Фреймворк Laravel 16](#_Toc390601158)

[1.1.8 Системи керування базами даних MySQL 16](#_Toc390601158)

[1.2 Огляд аналогів 19](#_Toc390601158)

[1.4 Опис предметної області та постановка задачі 21](#_Toc390601171)

[1.5 Обґрунтування вибору методу розв’язку задачі 21](#_Toc390601172)

[2 Практична частина 24](#_Toc390601173)

[2.1 Інформаційна структура моделі 2](#_Toc390601174)4

[2.2 Функціональні частини програмного продукту 26](#_Toc390601176)

[2.3 Інтерфейс та керівництво користувача 29](#_Toc390601176)

[2.4 Тестування проєкту 37](#_Toc390601176)

[3 економічна частина 46](#_Toc390601173)

[3.1 Характеристика та призначення роботи 46](#_Toc390601177)

[3.2 Витрати на створення програмного продукту 46](#_Toc390601177)

[3.3 Витрати на утримання та експлуатацію ПЕОМ 48](#_Toc390601177)

3.4 Загальновиробничі витрати 50

3.5 Обчислення собівартості програмного продукту 50

3.6 Обчислення гуртової ціни програмного продукту 50

3.7 Висновки економічної доцільності проєкту 51

[4 Охорона праці 46](#_Toc390601173)

[4.1 Аналізу потенційних небезпек, відносно до обладнання, устаткувань або](#_Toc390601177)

[приміщень, що розглядаються в дипломному проєкті 46](#_Toc390601177)

[4.2 Аналіз стану охорони праці. Розрахунок кількість комп’ютеризованих робочих місць та кількіть вогегасників 47](#_Toc390601177)

[4.3 Заходи щодо поліпшення умов з охорони праці та виробничої санітарії у](#_Toc390601177)

[досліджуваному приміщені 50](#_Toc390601177)

4.4 Інструкція з охорони праці при роботі з ПЕОМ та відео‑дисплейними терміналами 51

4.4.1 Загальні положення 51

4.4.2 Вимоги безпеки перед початком роботи 52

4.4.3 Вимоги безпеки під час роботи 53

4.4.4 Вимоги безпеки після закінчення роботи 54

4.4.5 Вимоги безпеки в аварійних ситуаціях 54

[ВИСНОВКИ 56](#_Toc390601190)

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 57](#_Toc390601200)

[Додаток 1 Лістинг програми](#_Toc390601202) 59

Представлений дипломний проєкт присвячений розробці навчально-пізнавальної платформи «Скандинавський епос», яка спрямована на популяризацію та глибше розуміння скандинавської культури, міфології та літератури. Головна мета проекту — створення інтерактивного вебресурсу, що надасть користувачам можливість зануритися в атмосферу скандинавських міфів через інтерактивні освітні матеріали. Об’єктом дослідження у проєкті виступають електронні освітні платформи, зокрема, їх здатність забезпечити ефективне навчання через інтерактивність та мультимедійність. Предметом дослідження є методи викладання історії та культури через цифрові технології, з особливим акцентом на використанні графіки, відео, аудіо та інших елементів для залучення користувачів.

Цільова аудиторія проекту охоплює не тільки студентів та школярів, але й усіх інших, хто має інтерес до скандинавської міфології та хоче дізнатися більше про цю колоритну культуру. Проєкт має значну соціальну важливість, адже сприяє культурному обміну та збереженню культурної спадщини в умовах глобалізації.

З технічної точки зору, проєкт використовує сучасні технології веброзробки. Фронтенд платформи реалізовано за допомогою бібліотеки React, що дозволяє створювати динамічний та зручний інтерфейс. Бекенд підтримується за допомогою Node.js, що забезпечує швидку обробку запитів і високу продуктивність сервера. Для зберігання даних використовується система управління базами даних MySQL, яка гарантує надійність і масштабованість інформаційної структури.

Загалом, проєкт «Скандинавський епос» має на меті створити повноцінне освітнє середовище, що стимулюватиме інтерес та розвиватиме знання користувачів у сфері скандинавської культури і міфології, використовуючи передові технології та інноваційні підходучасний світ заповнений неймовірним розмаїттям культурних та історичних скарбів, які розкидалися по всій планеті.

# Теоретична частина

### Основні концепції розробки навчально-пізнавальної платформи

Метою розділу є детальне роз’яснення ключових понять та теоретичних основ, на яких будується навчально-пізнавальна платформа. Таке глибоке розуміння необхідне для правильного сприйняття всіх наступних розділів проекту, де будуть обговорюватися більш конкретні аспекти розробки і впровадження платформи. Цей розділ задає фундаментальну базу для всього проекту, оскільки він розкриває основні концептуальні ідеї, що лежать в основі платформи, дозволяючи зрозуміти, як саме платформа буде служити своїй цільовій аудиторії.

Визначає цілі, які платформа прагне досягнути, показуючи її орієнтацію на залучення користувачів та надання їм якісних та взаємодійних навчальних матеріалів.

Висвітлює роль та значення скандинавської культури у сучасному освітньому просторі, підкреслюючи важливість збереження та популяризації цієї унікальної культурної спадщини через інноваційні технологічні рішення.

Завдяки чіткому розумінню цих основних концепцій, розробники та потенційні користувачі платформи зможуть краще оцінити, як платформа «Скандинавський епос» може збагатити їхнє знання і залученість у вивчення скандинавських традицій і міфології

#### 1.1.1 Визначення, цілі і завдання навчально-пізнавальної платформи

Навчально-пізнавальна платформа — це цифровий інструмент, призначений для самостійного дослідження та вивчення різноманітних тем. Така платформа забезпечує доступ до багатого архіву інформації, включаючи текст, зображення, аудіо та відео матеріали, і сприяє глибшому розумінню вивчаємої теми через інтерактивні функції, такі як віртуальні тури, інтерактивні мапи чи симуляції.

Перерахуємо та проаналізуємо основні цілі навчально-пізнавальної платформи.

Збагачення знань — надання користувачам доступу до широкого спектру інформації, яка дозволяє зануритись в дослідження інтересів без традиційних обмежень формальної освіти.

Підтримка самостійного навчання — стимулювання користувачів до самостійного пошуку та аналізу інформації, сприяючи розвитку критичного мислення та незалежних досліджень.

Інтерактивне дослідження — використання інтерактивних елементів для створення захоплюючого та залучаючого досвіду, який підвищує залученість та сприяє глибшому розумінню матеріалу.

Доз авданя навчально-пізнавальної платформи віднесемо: кураторство, організацію контенту, технічна оптимізацію, залучення інтерактивних засобів.

Кураторство та організація контенту — створення організованої структури контенту, що дозволяє легко навігувати та доступати інформацію, забезпечуючи користувачам можливість знаходити відповідні матеріали за їхніми інтересами.

Технічна оптимізація має на меті забезпечення стабільної та надійної роботи платформи на різноманітних пристроях, забезпечуючи доступність інформації для широкого кола користувачів.

Розробка та інтеграція інтерактивних функцій, які залучають користувачів до активного дослідження та взаємодії з контентом, підвищує їхнє залучення та збагачує досвід користування платформою.

#### 1.1.2 Цільова аудиторія, особливості контенту та його структура

Навчально-пізнавальні платформи орієнтовані на учнів середніх та старших класів, студентів коледжів та університетів, а також всіх, хто цікавиться додатковими джерелами знань, в основному гуманітарніих. Учні шкіл знаходять на платформах підтримку для засвоєння матеріалу з гуманітарних дисциплін, зокрема з історії, літератури, культурології, релігієзнавства тощо. Студенти можуть використовувати їх як додатковий ресурс для поглибленого вивчення своїх предметів. Вчителі та викладачі зможуть урізноманітнити навчальний процес, використовуючи інтерактивні елементи платформ.

Контент навчально-пізнавальних платформ має відзначатись високим рівнем інтерактивності, що забезпечує користувачам можливість взаємодії з навчальним матеріалом через інтерактивні карти, квізи, тести та інші елементи. Використання яскравих ілюстрацій, діаграм, відео та анімацій значно полегшує сприйняття складних тем і робить навчальний процес більш захоплюючим. Мультимедійний підхід платформ, який включає текстові, аудіо- та відеоматеріали, сприяє залученню різних типів учнів, враховуючи їхні індивідуальні потреби та уподобання.

Якісне інформаційне наповнення є ключовим аспектом успішної навчально-пізнавальної платформи. Детальні історичні огляди, точні описи міфів та легенд, біографії ключових персонажів та аналіз культурного впливу створюють багату і достовірну базу знань. Наукові статті, дослідження та академічні матеріали, представлені на платформі, забезпечують глибоке розуміння теми та сприяють формуванню критичного мислення. Висока якість контенту гарантує, що користувачі отримують надійну і актуальну інформацію, яка відповідає академічним стандартам і сприяє ефективному навчанню та самоосвіті.

#### 1.1.3 Інтерактівні елементи пізнання

Інтерактивні елементи є важливою складовою платформи, що робить процес навчання більш захоплюючим та ефективним. Серед таких елементів варто виділити:

1. Інтерактивні карти — вони дозволяють користувачам досліджувати історичні події, географічні локації та важливі об'єкти, взаємодіючи з ними через інтерактивний інтерфейс.

2. Квізи та тести допомагають перевірити знання користувачів, закріпити вивчений матеріал та виявити прогалини у знаннях. Вони також сприяють розвитку критичного мислення та вміння аналізувати інформацію.

3. Мультимедійні матеріали: Використання відео, аудіо, анімацій та інтерактивних презентацій допомагає залучити різні канали сприйняття інформації, що підвищує ефективність навчання.



Рисунок 5.1

Рисунок 5.2 Інтерактивні елементи пізнання

4. Віртуальні екскурсії та симуляції дозволяють користувачам зануритися у вивчення теми через віртуальні подорожі та симуляції історичних подій або процесів, що робить навчання більш наочним та захоплюючим.

Інтерактивні елементи не лише роблять навчання більш цікавим, але й сприяють глибшому розумінню матеріалу, закріпленню знань та розвитку навичок самостійного дослідження та критичного мислення.

#### 1.1.4 Технологічні рішення для реалізації платформи

Для створення навчально-пізнавальної платформи використовуються сучасні технологічні рішення, які забезпечують її ефективність, зручність та інтерактивність.

Фронтенд платформи зазвичай реалізується за допомогою сучасних JavaScript-фреймворків, таких як React або Angular. Ці фреймворки дозволяють створювати динамічний і швидкий користувацький інтерфейс. Для управління станом додатку можуть використовуватися бібліотеки, такі як Redux або Context API, які допомагають ефективно обробляти і зберігати дані додатку. Сучасні CSS-фреймворки, як-от Material-UI або Bootstrap, забезпечують стильове оформлення і адаптивність інтерфейсу, роблячи його привабливим і зручним для користувача.

Бекенд платформи часто розробляється з використанням серверних технологій, таких як Node.js, Django або Ruby on Rails. Ці платформи забезпечують високопродуктивну та масштабовану архітектуру для обробки запитів та управління даними. Використання RESTful або GraphQL API дозволяє фронтенду взаємодіяти з сервером, забезпечуючи ефективний обмін даними. Для роботи з базами даних можуть використовуватися реляційні системи керування базами даних (СУБД), такі як PostgreSQL або MySQL, які забезпечують надійне зберігання та швидкий доступ до даних. Використання ORM (Object-Relational Mapping) бібліотек, таких як TypeORM або Sequelize, спрощує роботу з базами даних, дозволяючи розробникам працювати з ними на рівні об'єктів.

Для забезпечення незалежності додатку від середовища виконання та спрощення процесу розгортання часто використовуються технології контейнеризації, такі як Docker. Це дозволяє створювати ізольовані середовища для розробки, тестування та виробництва, забезпечуючи стабільність і передбачуваність роботи додатку в різних умовах.

Використання CI/CD (Continuous Integration/Continuous Deployment) інструментів, таких як Jenkins або GitHub Actions, автоматизує процеси збірки, тестування та розгортання додатку, що підвищує ефективність розробки та знижує ризик помилок при введенні змін у коді.

Таким чином, поєднання сучасних фронтенд і бекенд технологій, ефективне управління даними та автоматизація процесів розробки забезпечують високу якість і функціональність навчально-пізнавальної платформи.

#### 1.1.5 Структура скандинавського епосу та джерела інформації

Скандинавський епос є важливою частиною культурної спадщини народів Північної Європи. Він охоплює міфологічні, героїчні та історичні сюжети, що передавалися з покоління в покоління через усну та письмову традиції. Скандинавський епос відображає світогляд, моральні цінності та історичний досвід вікінгів та інших скандинавських народів.

Основні джерела, які містять скандинавський епос, включають «Старшу Едду», «Молодшу Едду», «Саги про богів» та «Саги про героїв».

«Старша Едда» — зібрання поетичних творів, що містить міфологічні та героїчні пісні. Вона є одним з найважливіших джерел знань про скандинавську міфологію.

«Молодша Едда» написана Сноррі Стурлусоном, вона складається з прози та містить пояснення міфологічних історій, а також правила поетичного мистецтва.

Саги про богів розповідають про подвиги та пригоди скандинавських богів, таких як Один, Тор, Локі та інші.

Саги про героїв зосереджені на історіях про легендарних героїв, їхні подвиги та битви, такі як саги про Сігурда, Брюнгільду та інших.

Скандинавський епос в основному охоплює кілька ключових тем та сюжетів, які перераховно нижче.

Створення світу: оповідання про те, як боги створили світ з тіла велетня Іміра, є важливим аспектом скандинавської космології.

Битви богів та велетнів: постійні конфлікти між богами (Асами) та велетнями (Йотунами) є центральною темою багатьох міфів.

Подвиги героїв: історії про героїчні діяння, такі як здобуття скарбів, вбивство драконів та інші подвиги.

Рагнарок: кінцева битва, в якій загине більшість богів та велетнів, а світ буде знищено і відновлено знову.

Скандинавський епос багатий на персонажів, кожен з яких має свою унікальну роль та значення:

Один: Верховний бог, пов’язаний з мудрістю, війною та поезією. Він жертвує своїм оком за знання та має воронів, які приносять йому новини з усього світу.

Тор: Бог грому та війни, відомий своєю силою та молотом Мьольніром, який захищає людей і богів від велетнів.

Локі: Трикстер і змінник форм, відомий своїми підступними витівками, що часто призводять до великих проблем, але інколи допомагають богам.

Фрейя: Богиня любові, краси та війни, яка має вагому роль у багатьох міфах і пов'язана з магією та долею.

Цей розділ надає огляд структури скандинавського епосу, розкриваючи його інформаційне наповнення та основні джерела. Розуміння цих аспектів допомагає краще оцінити культурну значимість та багатство скандинавської міфології.

* + 1. **Мова розмітки гіпертексту HTML**

HTML (англ. HyperText Markup Language — мова розмітки гіпертекстових документів) — стандартна мова розмітки вебсторінок в інтернеті. Більшість вебсторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані користувача.

HTML є похідною мовою від SGML, успадкувавши від неї визначення типу документу та ідеологію структурної розмітки тексту. Попри те, що HTML — штучна комп’ютерна мова, вона не є мовою програмування. HTML разом із каскадними таблицями стилів та вбудованими скриптами — це три основні технології побудови вебсторінок.

HTML надає такі можливості: створення структурованого документу шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше; отримання інформації із Всесвітньої мережі через гіперпосилання; створення інтерактивних форм; включення зображень, звуку, відео, та інших об’єктів до тексту.[1]

* + 1. **Каскадні таблиці стилів CSS**

Каскадні таблиці стилів (англ. Cascading Style Sheets або скорочено CSS) — спеціальна мова, що використовується для опису сторінок, написаних мовами розмітки даних. Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі. CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці вебсторінок. Головна перевага блочної верстки — розділення змісту сторінки (даних) та їхньої візуальної презентації.

CSS використовується авторами та відвідувачами вебсторінок, щоб визначити кольори, шрифти, верстку та інші аспекти вигляду сторінки. Одна з головних переваг — можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, XML або подібна мова розмітки) від вигляду документу (що описується в CSS).

Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори тощо. CSS також дозволяє адаптувати контент до різних умов відображення (на екрані монітора, мобільного пристрою (КПК), у роздрукованому вигляді, на екрані телевізора, пристроях з підтримкою шрифту Брайля або голосових браузерах та ін.).

Один і той самий HTML або XML документ може бути відображений по-різному залежно від використаного CSS. Стилі для відображення сторінки можуть бути:

* Стилі автора (інформація надана автором сторінки):
* зовнішні таблиці стилів (англ. stylesheet), найчастіше окремий файл або файли .css;
* внутрішні таблиці стилів, включені як частина документу або блоку;
* стилі для окремого елементу.
* Стилі користувача локальний .css-файл, вказаний користувачем для використання на сторінках і вказаний в налаштуваннях браузера.
* Стилі браузера стандартний стиль переглядача, стандартні стилі для елементів, визначені браузером, використовуються коли немає інформації про стиль елемента або вона неповна.

Стандарт CSS визначає порядок та діапазон застосування стилів, тобто, в якій послідовності і для яких елементів застосовуються стилі. Таким чином, використовується принцип каскадності, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена загальнішими стилями [2].

* + 1. **Скриптова мова програмування JavaScript**

JavaScript (JS) — динамічна, [об’єктно-орієнтована](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)[прототипна](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D1%82%D0%B8%D0%BF%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) [мова програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F). Реалізація стандарту [ECMAScript](https://uk.wikipedia.org/wiki/ECMAScript). Найчастіше використовується для створення сценаріїв [вебсторінок](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0), що надає можливість на стороні [клієнта](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%BD%D0%B0_%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0) (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, [асинхронно](https://uk.wikipedia.org/wiki/AJAX) обмінюватися даними з [сервером](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%BD%D0%B0_%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0), змінювати [структуру](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%B0_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0) та [зовнішній вигляд](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD) [вебсторінки](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0).

JavaScript класифікують як [прототипну](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D1%82%D0%B8%D0%BF%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) (підмножина [об’єктно-орієнтованої](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)), [скриптову мову](https://uk.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B2%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) програмування з динамічною типізацією.

Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування ([імперативну](https://uk.wikipedia.org/wiki/%D0%86%D0%BC%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F" \o "Імперативне програмування) та частково [функціональну](https://uk.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D1%96%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) і деякі відповідні архітектурні властивості, зокрема: [динамічна](https://uk.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D1%96%D1%87%D0%BD%D0%B0_%D1%82%D0%B8%D0%BF%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F) та слабка [типізація](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%82%D0%B8%D0%BF%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%97), автоматичне керування пам’яттю, прототипне [наслідування](https://uk.wikipedia.org/wiki/%D0%9D%D0%B0%D1%81%D0%BB%D1%96%D0%B4%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)), [функції](https://uk.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) як [об’єкти першого класу](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82_%D0%BF%D0%B5%D1%80%D1%88%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BB%D0%B0%D1%81%D1%83).

Мова JavaScript використовується для:

* написання сценаріїв [вебсторінок](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0) для надання їм інтерактивності;
* створення односторінкових [вебзастосунків](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) ([React](https://uk.wikipedia.org/wiki/React" \o "React), [AngularJS](https://uk.wikipedia.org/wiki/AngularJS), [Vue.js](https://uk.wikipedia.org/wiki/Vue.js));
* [програмування](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) на стороні сервера ([Node.js](https://uk.wikipedia.org/wiki/Node.js));
* стаціонарних [застосунків](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) ([Electron](https://electron.atom.io/), [NW.js](https://nwjs.io/));
* мобільних застосунків ([React Native](https://facebook.github.io/react-native/), [Cordova](https://cordova.apache.org/));
* сценаріїв в прикладному [ПЗ](https://uk.wikipedia.org/wiki/%D0%9F%D0%97) (наприклад, в програмах зі складу [Adobe Creative Suite](https://uk.wikipedia.org/wiki/Adobe_Creative_Suite) чи [Apache JMeter](https://uk.wikipedia.org/wiki/Apache_JMeter));
* всередині [PDF](https://uk.wikipedia.org/wiki/PDF)-документів тощо.

Незважаючи на схожість назв, мови [Java](https://uk.wikipedia.org/wiki/Java) та JavaScript є двома різними мовами, що мають відмінну [семантику](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), хоча й мають схожі риси в [стандартних бібліотеках](https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D0%B0_%D0%B1%D1%96%D0%B1%D0%BB%D1%96%D0%BE%D1%82%D0%B5%D0%BA%D0%B0) та правилах іменування. [Синтаксис](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81_%D0%BC%D0%BE%D0%B2%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) обох мов отриманий «у спадок» від мови [С](https://uk.wikipedia.org/wiki/C_(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)), але семантика та дизайн JavaScript є результатом впливу мов [Self](https://uk.wikipedia.org/wiki/Self_(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) та [Scheme](https://uk.wikipedia.org/wiki/Scheme).

JavaScript, наразі, є однією з найпопулярніших мов програмування в [інтернеті](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82). Але спочатку багато професійних програмістів скептично ставилися до мови, цільова аудиторія якої складалася з програмістів-любителів. Поява [AJAX](https://uk.wikipedia.org/wiki/AJAX) змінила ситуацію та повернула увагу професійної спільноти до мови, а подальші модифікації мови за стандартами ES2015 та ES2017 внесли багато корисних можливостей, яких не вистачало для ефективного програмування. В результаті, були розроблені та покращені багато практик використання JavaScript (зокрема, [тестування](https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F) та [налагодження](https://uk.wikipedia.org/wiki/%D0%9D%D0%B0%D0%BB%D0%B0%D0%B3%D0%BE%D0%B4%D0%B6%D0%B5%D0%BD%D0%BD%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC)), створені бібліотеки та [фреймворки](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B8%D0%B9_%D0%BA%D0%B0%D1%80%D0%BA%D0%B0%D1%81), поширилося використання JavaScript поза браузером.[3]

* + 1. **Фреймворк Bootstrap**

Bootstrap — вільний набір інструментів для створення сайтів і вебдодатків. Включає в себе HTML- і CSS-шаблони оформлення для типографіки, вебформ, кнопок, міток, блоків навігації та інших компонентів вебінтерфейсу, включаючи JavaScript-розширення.

Bootstrap використовує сучасні напрацювання в області CSS і HTML, тому необхідно бути уважним при підтримці старих браузерів.

Ця бібліотека почала розроблятися як внутрішня бібліотека компанії Twitter під назвою Twitter Blueprint. Після кількох місяців розробки вона була відкрита під назвою Bootstrap 19 серпня 2011.

Основними нововведеннями другої версії, що з’явилася 31 січня 2012 року, стали 12-колоночна сітка і підтримка адаптивності.

Третя версія випущена 19 серпня 2013 року. У ній адаптивність отримала подальший розвиток, був здійснений перехід до концепції mobile first, оптимізації перш за все під мобільні пристрої. Дизайн за замовчуванням став плоским.

Роботу над четвертою версією розпочато 29 жовтня 2014 года . Альфа версія вийшла 19 серпня 2015 года. Перша бета версія випущена 10 серпня 2017. Друга бета версія випущена 19 жовтня 2017. 18 січня 2018 випущена перша стабільна версія Bootstrap 4. Bootstrap 4 — це майже повністю переписаний Bootstrap 3.[4]

* + 1. **Фреймворк jQuery**

jQuery — популярна [JavaScript](https://uk.wikipedia.org/wiki/JavaScript)-бібліотека з [відкритим кодом](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%BA%D1%80%D0%B8%D1%82%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F). Вона була представлена у січні 2006 року у BarCamp NYC Джоном Ресіґом (John Resig). Згідно з дослідженнями організації W3Techs, JQuery використовується понад половиною від мільйона найвідвідуваніших [сайтів](https://uk.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82).jQuery є найпопулярнішою бібліотекою JavaScript, яка посилено використовується на сьогоднішній день.

jQuery є [вільним програмним забезпеченням](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%BB%D1%8C%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F) під [ліцензією MIT](https://uk.wikipedia.org/wiki/%D0%9B%D1%96%D1%86%D0%B5%D0%BD%D0%B7%D1%96%D1%8F_MIT) (до вересня 2012 було подвійне ліцензування під [MIT](https://uk.wikipedia.org/wiki/MIT_License) та [GNU General Public License](https://uk.wikipedia.org/wiki/GNU_General_Public_License) другої версії).

Синтаксис jQuery розроблений, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів [DOM](https://uk.wikipedia.org/wiki/DOM), створенню анімації, обробки подій, і розробки [AJAX](https://uk.wikipedia.org/wiki/AJAX)-[застосунків](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA" \o "Застосунок). jQuery також надає можливості для розробників, для створення [плагінів](https://uk.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%B3%D1%96%D0%BD) у верхній частині бібліотеки JavaScript. Використовуючи ці об’єкти, розробники можуть створювати абстракції для низькорівневої взаємодії та створювати анімацію для ефектів високого рівня. Це сприяє створенню потужних і динамічних [вебсторінок](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0).[5]

* + 1. **Скриптової мови програмування PHP**

PHP ― це широко використовувана мова сценаріїв загального призначення з відкритим вихідним кодом. Говорячи простіше, PHP це мова програмування, спеціально розроблена для написання web-додатків (сценаріїв), що виконуються на Web-сервері.

Абревіатура PHP означає "Hypertext Preprocessor (Препроцесор Гіпертексту)". Синтаксис мови бере початок з C, Java і Perl. PHP досить простий для вивчення. Перевагою PHP є надання web-розробникам можливості швидкого створення динамічних web-сторінок.

Важливоюперевагою мови PHP перед такими мовами, як мов Perl і C є можливість створення HTML документів із вбудованими командами PHP.

Значною відмінністю PHP від якого-небудь коду, що виконується на стороні клієнта, наприклад, JavaScript, є те, що PHP-скрипти виконуються на стороні сервера. Можна конфігурувати свій сервер таким чином, щоб HTML‑файли оброблялися процесором PHP, так що клієнти навіть не зможуть дізнатися, чи отримують вони звичайний HTML-файл або результат виконання скрипта.

PHP дозволяє створювати якісні Web-додатки за дуже короткі терміни, отримуючи продукти, що легко модифікуються і підтримуються в майбутньому. PHP простий для освоєння, і разом з тим здатний задовольнити запити професійних програмістів.

Мова PHP постійно удосконалюється, і їй, напевно забезпечене довге домінування в області мов web-програмування, принаймні, найближчим часом.

Існують три основні області застосування PHP.

* Створення скриптів для виконання на стороні сервера. PHP найбільш широко використовується саме таким чином. Все, що знадобиться, це інтерпретатор PHP (у вигляді програми CGI або серверного модуля), вебсервер і браузер. Для того щоб можна було переглядати результати виконання PHP-скриптів в браузері, потрібен працюючий вебсервер і встановлений PHP.
* Створення скриптів для виконання в командному рядку. Можна створити PHP-скрипт, здатний запускатися незалежно від вебсервера та браузера. Все, що буде потрібно ― парсер PHP. Такий спосіб використання PHP ідеально підходить для скриптів, які повинні виконуватися регулярно, наприклад, за допомогою cron (на платформах \* nix або Linux) або за допомогою планувальника завдань (Task Scheduler) на платформах Windows. Ці скрипти також можуть бути використані в задачах простої обробки текстів.
* Створення віконних додатків, що виконуються на стороні клієнта. Можливо, PHP є не найкращою мовою для створення подібних додатків, але, якщо необхідно використовувати саме PHP для створення повноцінних функціональних клієнт-додатків, можна використовувати PHP-GTK для створення таких додатків. Подібним чином можна створювати і крос-платформні додатки. PHP-GTK є розширенням PHP і не поставляється разом з дистрибутивом PHP.
* У 2014 році було проведено голосування, за результатами якого наступна версія отримала назву PHP7. Вихід нової версії планувався в середині жовтня 2015 року. У березні 2015 року Zend представили інфографіку в якій описані основні нововведення PHP7.

3 грудня 2015 року було оголошено про вихід PHP версії 7.0.0.

Нова версія грунтується на експериментальній гілці PHP, яка спочатку називалася phpng (PHP Next Generation ― наступне покоління), і розроблялася з упором на збільшення продуктивності і зменшення споживання пам’яті. У новій версії додана можливість вказувати тип отримуваних з функцій даних, доданий контроль переданих типів для скалярних даних, а також нові оператори.[6]

* + 1. **Фреймворк Laravel**

Laravel — безкоштовний, з відкритим кодом PHP-фреймворк, створений Тейлором Отвелом і призначений для розробки вебдодатків відповідно до шаблону model–view–controller.

Серед особливостей Laravel можна назвати: модульну систему упакування з виділеним менеджером залежностей Composer, різні способи для доступу до реляційних баз даних, утиліти, які допомагають в розгортанні додатків і технічного обслуговування.

Станом на березень 2015 року, Laravel вважався одним з найпопулярніших PHP фреймворком, разом з Symfony5, Nette, CodeIgniter, Yii2 й іншими.[7]

* + 1. **Системи керування базами даних MySQL**

MySQL ― це популярна система управління базами даних (СУБД), дуже часто застосовується в поєднанні з PHP.

База даних являє собою структуровану сукупність даних. Ці дані можуть бути будь-якими ― від простого списку майбутніх покупок до переліку експонатів картинної галереї або величезної кількості інформації в корпоративній мережі. Для запису, вибірки й обробки даних, що зберігаються в комп’ютерній базі даних, необхідна система управління базою даних, якою і є MySQL. Оскільки комп’ютери чудово справляються з обробкою великих обсягів даних, управління базами даних відіграє центральну роль в обчисленнях. Реалізовано таке управління може бути по-різному ― як у вигляді окремих утиліт, так і у вигляді коду, що входить до складу інших додатків.

MySQL ― це система управління реляційними базами даних. У реляційній базі даних дані зберігаються в окремих таблицях, завдяки чому досягається виграш у швидкості й гнучкості. Таблиці зв’язуються між собою за допомогою відносин, завдяки чому забезпечується можливість поєднувати при виконанні запиту дані з декількох таблиць. SQL як частину системи MySQL можна охарактеризувати як мову структурованих запитів плюс найбільш поширена стандартна мова, яка використовується для доступу до баз даних. MySQL ― це ПЗ з відкритим кодом. Застосовувати його і модифікувати може будь-хто. Таке ПЗ можна одержувати по Internet і використовувати безкоштовно. При цьому кожен користувач може вивчити вихідний код і змінити його у відповідності зі своїми потребами. Використання програмного забезпечення MySQL регламентується ліцензією GPL (GNU General Public License), http://www.gnu.org/licenses/, в якій зазначено, що можна і чого не можна робити з цим програмним забезпеченням у різних ситуаціях.

MySQL є дуже швидким, надійним і легким у використанні. MySQL володіє також рядом зручних можливостей, розроблених у тісному контакті з користувачами. Спочатку сервер MySQL розроблявся для керування великими базами даних з метою забезпечити більш високу швидкість роботи в порівнянні з існуючими на той момент аналогами. І ось вже протягом декількох років даний сервер успішно використовується в умовах промислової експлуатації з високими вимогами. Незважаючи на те, що MySQL постійно вдосконалюється, він уже сьогодні забезпечує широкий спектр корисних функцій. Завдяки своїй доступності, швидкості та безпеки MySQL дуже добре підходить для доступу до баз даних по Internet.

MySQL є системою клієнт-сервер, яка містить багатопоточний SQL-сервер, що забезпечує підтримку різних обчислювальних машин баз даних, а також кілька різних клієнтських програм і бібліотек, засоби адміністрування і широкий спектр програмних інтерфейсів (API).

MySQL складається з двох частин: серверної і клієнтської. Сервер MySQL постійно працює на комп’ютері. Клієнтські програми (наприклад, скрипти PHP) посилають серверу MySQL SQL-запити через механізм сокетів (тобто за допомогою мережевих засобів), сервер їх обробляє і запам’ятовує результат. Тобто скрипт (клієнт) вказує, яку інформацію він хоче отримати від сервера баз даних. Потім сервер баз даних посилає відповідь (результат) клієнту (скрипту).

Не завжди передається весь результат тому, що розмір результуючого набору даних може бути занадто великим, і на його передачу по мережі піде надто багато часу. Та й рідко коли буває потрібно отримувати відразу весь висновок запиту (тобто всі записи, що задовольняють висловом запиту). Наприклад, може знадобитися лише підрахувати, скільки записів задовольняється тій чи іншій умові, або ж вибрати з даних тільки перші 10 записів. Механізм використання сокетів базується на технології клієнт-сервер, а це означає, що в системі повинна бути запущена спеціальна програма ― MySQL-сервер, яка приймає і обробляє запити від програм. Тому що вся робота відбувається в дійсності на одній машині, накладні витрати по роботі з мережевими засобами незначні (установка і підтримка з’єднання з MySQL-сервером обходиться досить дешево).

Cтруктура MySQL трирівнева: бази даних ― таблиці ― записи. Бази даних і таблиці MySQL фізично представляються файлами з розширеннями frm, MYD, MYI. Логічно ― таблиця являє собою сукупність записів. А записи — це сукупність полів різного типу. Ім’я бази даних MySQL унікальне в межах системи, а таблиці ― в межах бази даних, поля ― в межах таблиці. Один сервер MySQL може підтримувати відразу декілька баз даних, доступ до яких може розмежовуватись логіном і паролем. Знаючи логін і пароль, можна працювати з конкретною базою даних. Наприклад, можна створити або видалити в ній таблицю, додати записи і т. д. Зазвичай ім’я-ідентифікатор та пароль призначаються хостинг провайдерами, які і забезпечують підтримку MySQL для своїх користувачів.[8]

1.2 Огляд аналогів

В мережі Інтернет є багато вебресурсів, які можна розглядати як аналоги даного проєкту, наприклад майже на всіх сайтах туроператорів є розділ, який стосується теми дипломного проєктування. Також можна знайти багато довідкових порталів, де розміщена інформація про цікаві містя світу. Серед них було знайдено ресурс «Triphearts», на якому розміщена інформація, яка буде корисна мандрівникові. Тут є можливість спланувати свій бюджет подорожі, підібрати варіанти перельотів та проживання, а також створити власні публікації.

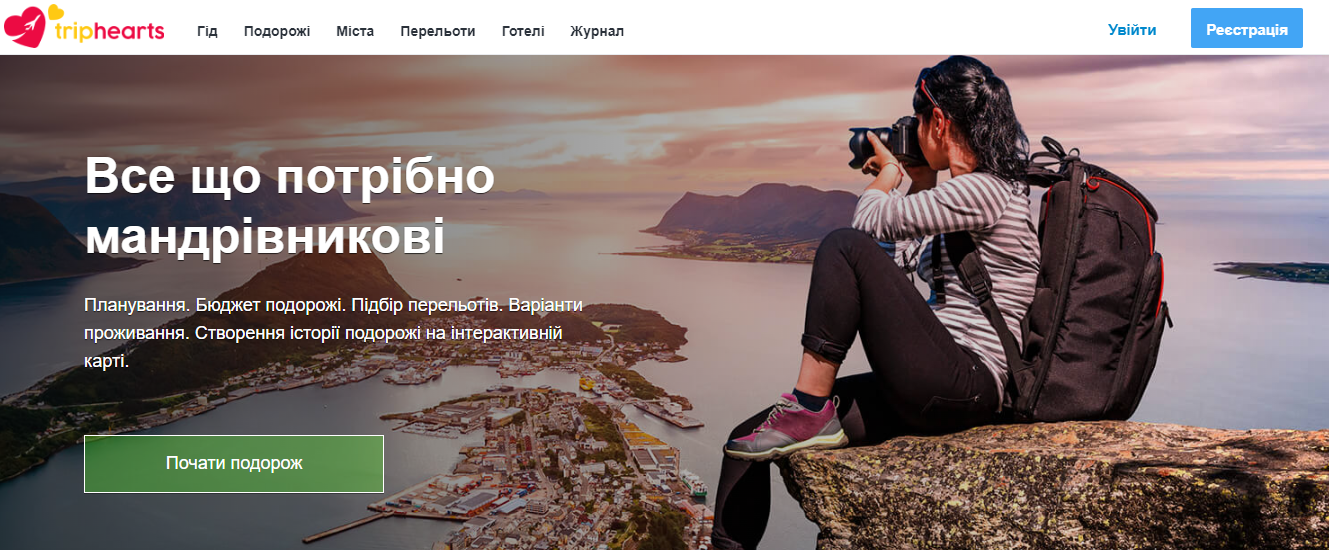


Рисунок 1.1 Головна сторінка ресурсу «Triphearts»

У розділі «Гід» (рисунок 1.2) знаходиться путівник по країнам, який розповідає куди піти, що подивитися, тобто тут є інформація про визначні пам’ятки та місця, розваги, транспорт, харчування, авіаквитки та варіанти проживання. Ознайомившись із інформацією користувачі можуть спланувати бюджет поїздки та розробити план подорожі.

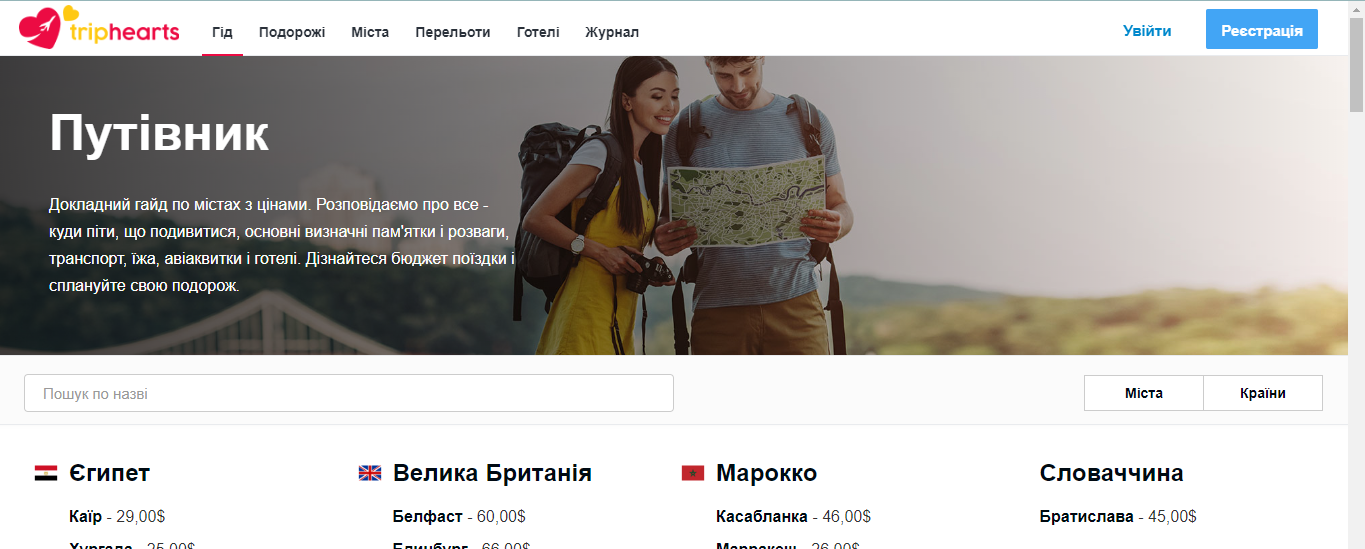


Рисунок 1.2 Розділ «Гід»

Найбільш цікавим є розділ «Журнал» (рисунок 1.3), адже він надає можливість користувачам створювати власні публікації про історії власних подорожей, а також переглядати дописи інших. Мандрівники пишуть про цікаві містя, лайфаки, екзотичні маршрути, події і т.д.



Рисунок 1.3 Розділ «Журнал»

Було досліджено механізм створення публікацій адже це і є основним завданням дипломного проєктування: надавати користувачам можливість створювали свої публікації про цікаві містя світу.

Вебресурс «Triphearts» взято за основу під час розробки програмного продукту.

[**1.4 Опис предметної області та постановка задачі**](#_Toc355967843)

Під час дипломного проєктування необхідно вебрусурс для створення публікацій «Цікаві місця світу».

Основні розділи сайту:

- Головна.

- Журнал публікацій.

- Реєстрація/Авторизація.

- Корисні ресурси.

В системі необхідно реалізувати такі категорії користувачів:

- Адміністратор.

- Незареєстрований користувач.

- Зареєстрований користувач.

Звичайний користувач (незареєстрований) повинен мати можливість переглядати публікації інших користувачів.

Зареєстрований та авторизований користувач має мати можливість створювати власні публікації та залишати відгуки.

Адміністратор каталогу повинен користуватися всіма можливостями користувача, але крім цього має мати можливість керувати інформацією про користувачів, модерувати їх публікації та відгуки.

На вебресурсі необхідно реалізувати пошук, сортування та фільтрування публікацій за різними критеріями.

Система має мати зручний та інтуїтивно зрозумілий інтерфейс і приємний дизайн.

[**1.5 Обґрунтування вибору методу розв’язку задачі**](#_Toc355967844)

Для розробки клієнтської частина проєкту було вирішено використовувати такі технології: фреймворк Bootstrap, фреймворк jQuery. Щодо серверної частини вибір зупинився на скриптовоній мові програмування PHP, фреймворку Laravel, системі керування базами даних MySQL.

Існує безліч різноманітних інструментів, які можуть створювати та генерувати вебсторінки на вебсайтах. Одним із таких популярних фреймворків є Bootstrap. Він допомагає швидше та простіше розробляти зовнішній вигляд вебсторінок. Він підходить для людей з будь-яким рівнем досвіду, для пристроїв будь-яких форматів, та проєктів будь-якого розміру.

З розвитком технологій функціональність вебсторінок постійно зростає й наближається до функціональності настільних прикладних програм. Ця зростаюча функціональність реалізується за допомогою Javascript.

Для того щоб особисто не прописувати всі необхідні ефекти є величезна кількість бібліотек, що дозволяють реалізовувати ці ефекти. І серед безлічі цих бібліотек по праву найбільш зручною й доступною для розуміння є **бібліотека jQuery**.jQuery ― бібліотека Javascript, що фокусується на взаємодії Javascript і HTML.

PHP має свої відмінності від [JavaScript](https://astwellsoft.com/uk/blog/tehnology/javascript.html). Його сценарії виконуються на стороні сервера і створюють HTML-код, який відправляється клієнту. Клієнт отримує готовий результат, і сервер можна налаштувати так, що користувачі не зможуть відрізнити звичайний HTML-документ від результатів роботи скриптів.

Мова PHP відносно проста для програмування, але при цьому вона задовольняє безліч запитів професійних розробників. Програмні продукти, створені на PHP, працюють дуже швидко і підтримуються переважною більшістю серверного обладнання.

Щоб розширити можливості PHP був використаний фреймворк Laravel. Це безкоштовний PHP-фреймворк з відкритим вихідним кодом спеціально розроблений для створення складних сайтів і вебдодатків. Дозволяє спростити автентифікацію, маршрутизацію, сесії, кешування, архітектуру програми, роботу з базою даних. Він допомагає визначити структуру вебзастосунку і служить для неї каркасом.

Для сховища даних вирішено використовувати ― **MySQL, оскільки це** вільна реляційна система керування базами даних, яка працює як сервер для забезпечення багатокористувацького доступу до великої кількості баз даних.

**2** **Практична частина**

**2.1 Інформаційна структура моделі**

Всі вхідні та вихідні дані проєкту зберігаються у базі даних. База даних складається з 7 таблиць (рисунок 2.1).

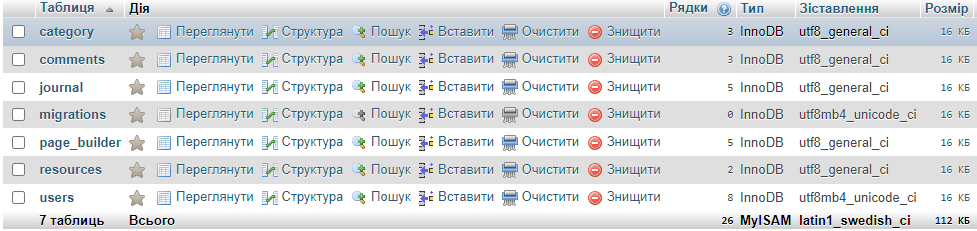


Рисунок 2.1 Структура бази даних

Короткий опис таблиць:

- «users» — зберігає інформацію про користувачів (рисунок 2.2),

- «comments» — зберігає інформацію про відгуки користтувачів (рисунок 2.3),

- «category» — зберігає інформацію про категорії публікацій (рисунок 2.4),

- «journal» — зберігає публікації створені користувачами (рисунок 2.5),

- «page\_builder» — зберігає інформацію, яка відображається на головній сторінці вебресурсу (рисунок 2.6),

- «resources» — зберігає інформацію про інформацію про корисні ресурси (рисунок 2.7),

- «migrations» — зберігає службову інформацію, яку використовує фреймворком Laravel (рисунок 2.8).



Рисунок 2.2 Структура таблиці «users»



Рисунок 2.3 Структура таблиці «comments»

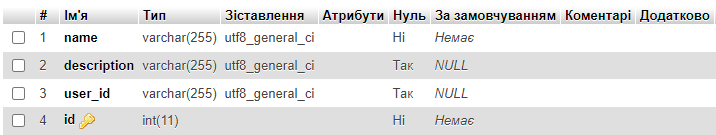


Рисунок 2.4 Структура таблиці «category»

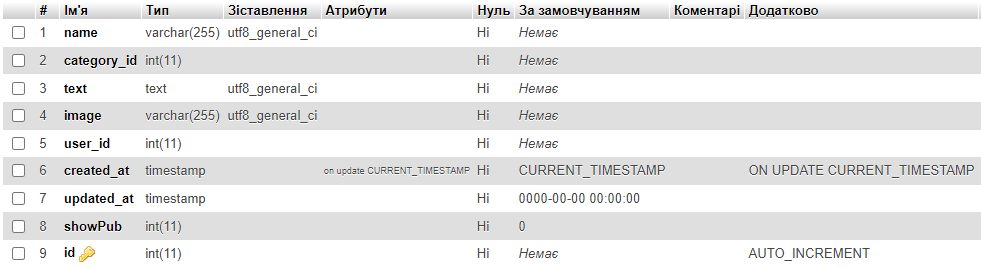


Рисунок 2.5 Структура таблиці «journal»



Рисунок 2.6 Структура таблиці «page\_builder»



Рисунок 2.7 Структура таблиці «resources»

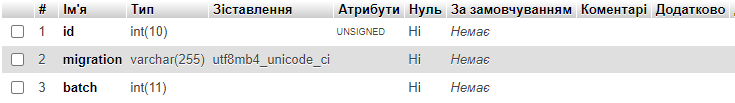


Рисунок 2.8 Структура таблиці «migrations»

* 1. **Функціональні частини програмного продукту**

Бекенд проєкту розроблено за допомогою фреймворку Laravel 10. Для фронтенду було використано фреймворк Bootstrap та бібліотекуjQuery. Всі файли проєкту із вихідним кодом знаходяться в каталозі «inter.ttele.me». Структуру каталогу проєкту наведено на рисунку 2.9.

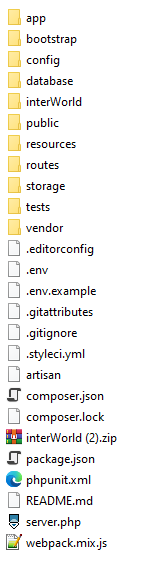


Рисунок 2.9 Вміст каталогу проєкту «inter.ttele.me»

На рисунку 2.10 зображено структуру каталогу «public», який є точкою входу на вебсайт.

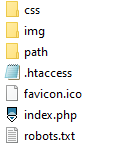


Рисунок 2.10 Структура каталогу «public»

На рисунку 2.11 зображено структуру каталогу «database», в якому знаходяться файли для створення та наповнення таблиць бази даних.

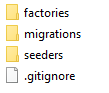


Рисунок 2.11 Структура каталогу «database»

На рисунку 2.12 зображено структуру каталогу «app», в якому знаходяться файли, що відповідають за функціонал бекенду.

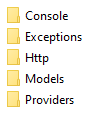


Рисунок 2.12 Структура каталогу «app»

На рисунку 2.13 зображено структуру каталогу «resources», в якому знаходяться файли, що відповідають за функціонал фронтенду та бекенду.

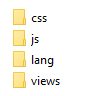


Рисунок 2.13 Структура каталогу «resources»

На рисунку 2.14 зображено структуру каталогу «storage», в якому знаходяться файли та зображення з якими працює вебсайт.

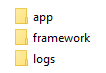


Рисунок 2.14 Структура каталогу «storage»

На рисунку 2.15 зображено структуру каталогу «views», в якому знаходяться файли, що відповідають за представлення. Кожен файл у цій директиві відповідає за певну частину або вид сторінки вебдодатку, а також за відображення даних та взаємодію з користувачем.

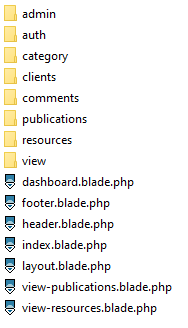


Рисунок 2.15 Структура каталогу «views»

Основні функціональні частини проєкту:

* index.blade.php/header.blade.php/footer.blade.php ― файли, що відповідають за головну сторінку вебресурсу;
* create-user.blade.php/edit-user.blade.php ― файли, що відповідають за керування інформацією про користувачів;
* comments.blade.php/comments-create.blade.php ― файли, що відповідають за керування інформацією про відгуки користувачів;
* category.blade.php/category‑create.blade.php/edit‑‑category.blade.php — файли, що відповідають за керування інформацією про категорії;
* content.blade.php/content‑create.blade.php/edit‑content.blade.php — файли, що відповідають за керування інформацією, яка відображається на головній сторінці;
* edit‑publications.blade.php/publications.blade.php/publications‑create.blade.php — файли, що відповідають за керування інформацією про публікації.

**2.3 Інтерфейс та керівництво користувача**

Розроблений вебресурс система розміщена за адресою: http://inter.ttele.me/. Після завантаження системи, у вікні браузера з’являється головна сторінка (рисунок 2.16), на якій зверху зліва знаходиться назва, а справа посилання на розділи сайту («Журнал публікацій» (рисунок 2.17), «Корисні ресурси» (рисунок 2.18).) і сторінку для авторизації адміністратора та користувачів. Нижче розміщена інформація про деякі цікаві містя світу.

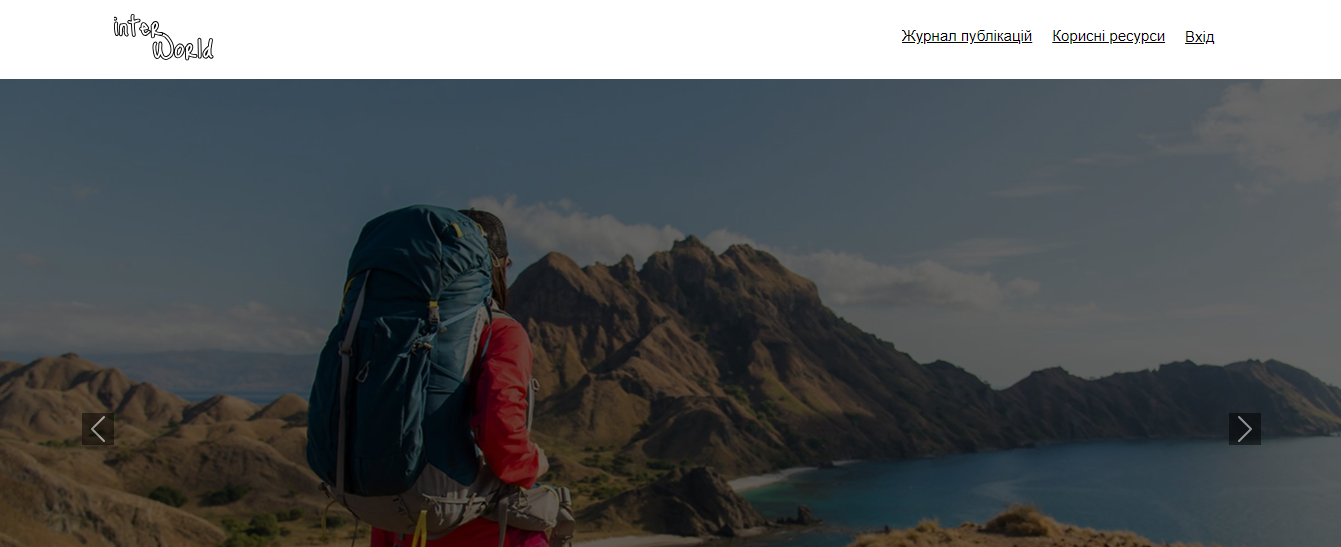


Рисунок 2.16 Головна сторінка вебресурсу

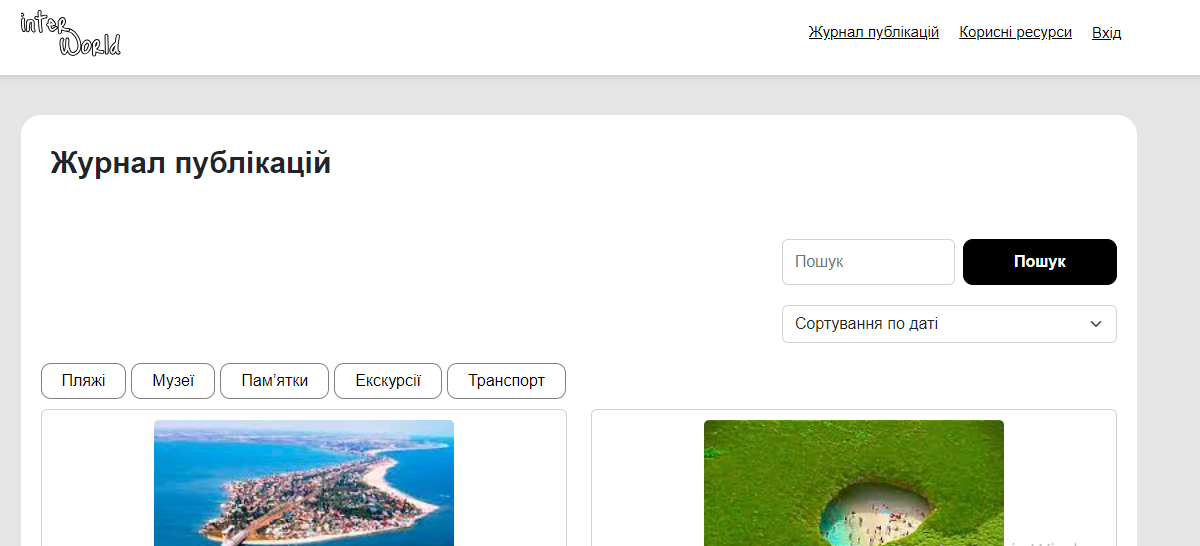


Рисунок 2.17 Сторінка «Журнал публікацій»

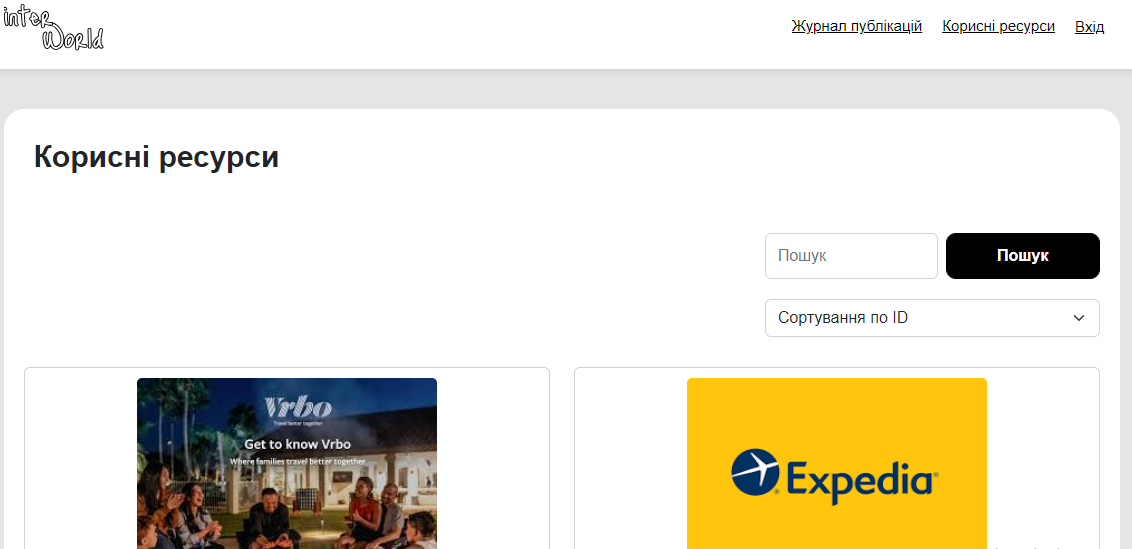


Рисунок 2.18 Сторінка «Корисні ресурси»

Працювати з вебресурсом можуть три категорії користувачів, а саме адміністратор, неавторизований (гість) та авторизований користувачі.

Звичайний користувач має можливість переглядати публікації інших, а також використовувати фільтрування, пошук та сортування дописів.

Створювати публікації може тільки зареєстрований та авторизований користувач. Для реєстрації необхідно натиснути на посилання «Вхід» у верхній частині головної сторінки. Після чого відкриється сторінка із відповідною формою (рисунок 2.19), де потрібно ввести ім’я, електронну адресу та пароль.

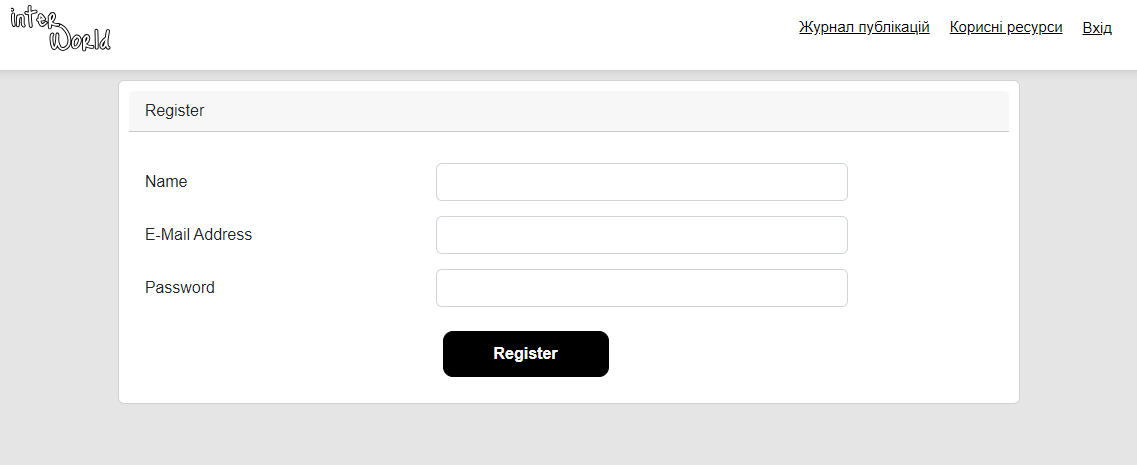


Рисунок 2.19 Сторінка реєстрації користувача

Зареєструвавшись користувач може авторизуватися і додавати публікації, дописи в розділі корисні ресурси, а також залишати відгуки під публікаціями інших (рисунки 2.20-2.21).

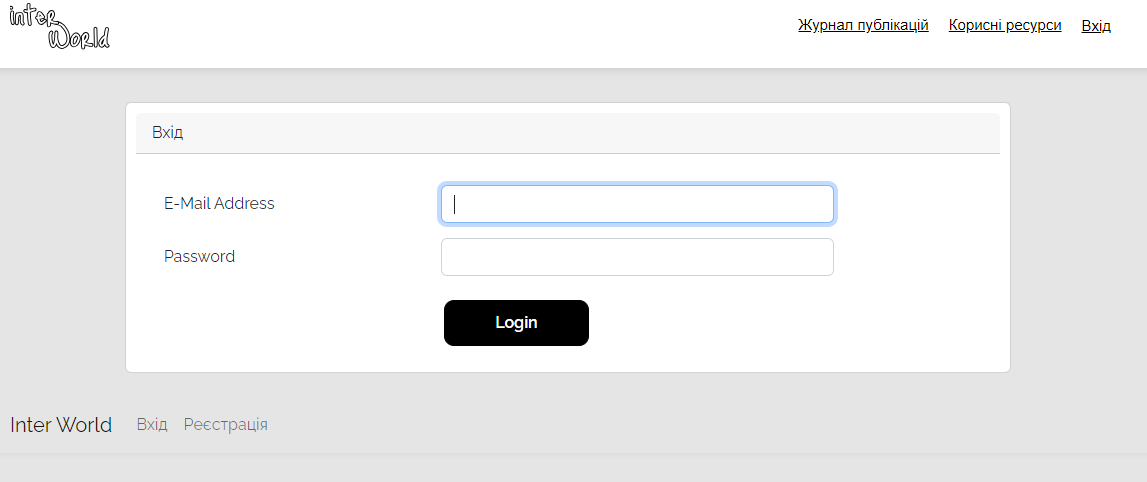


Рисунок 2.20 Сторінка авторизації користувача



Рисунок 2.21 Вигляд меню авторизованого користувача

Щоб додати публікацію необхідно натиснути на олівець біля назви розділу, відкриється сторінка із власними публікаціями, на якій треба натиснути на кнопку «Створити» (рисунок 2.22).

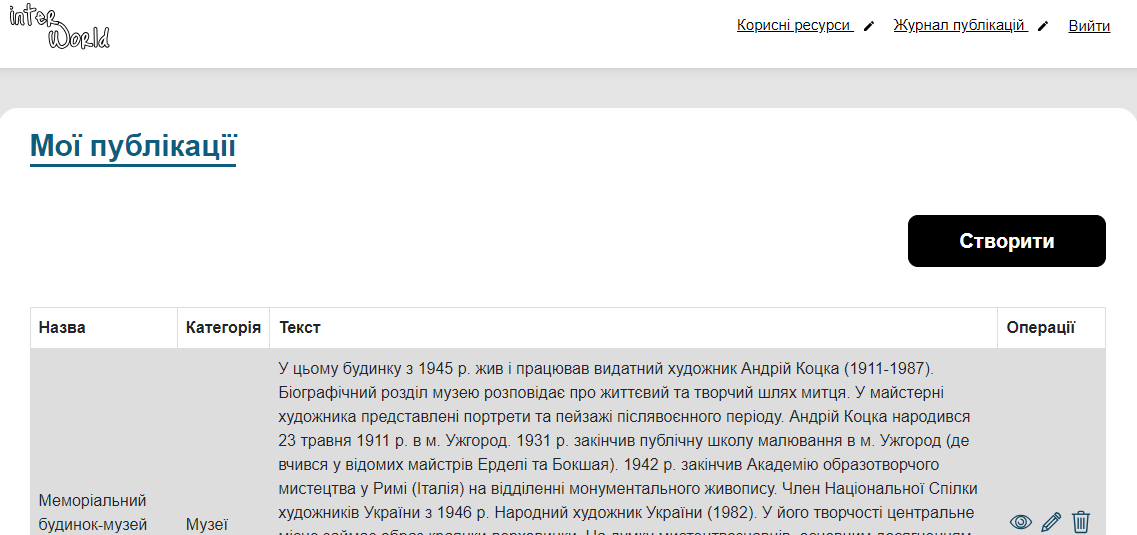


Рисунок 2.22 Сторінка «Журнал публікацій» в режимі авторизованого користувача

Для додавання публікації необхідно ввести її назву, обрати категорію, додати опис та завантажити зображення (рисунок 2.23).

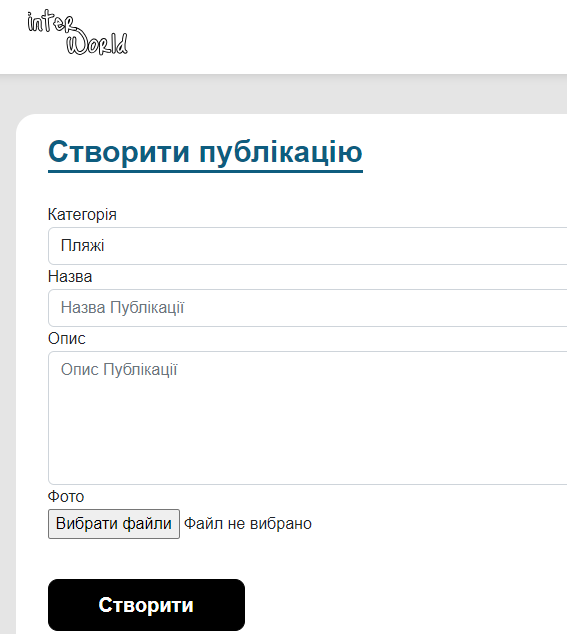


Рисунок 2.23 Сторінка додавання публікацій

Свої публікації користувач може редагувати, видаляти, відкрити попередній перегляд натиснувши на відповідні кнопки .

Зареєстрований користувач має можливість створювати дописи у розділі «Корисні посилання». Для цього необхідно так само натиснути на олівець біля назви розділу, а потім на кнопку «Створити» (рисунок 2.24) і заповнити поля форми (рисунок 2.25).



Рисунок 2.24 Сторінка «Корисні ресурси» в режимі авторизованого користувача

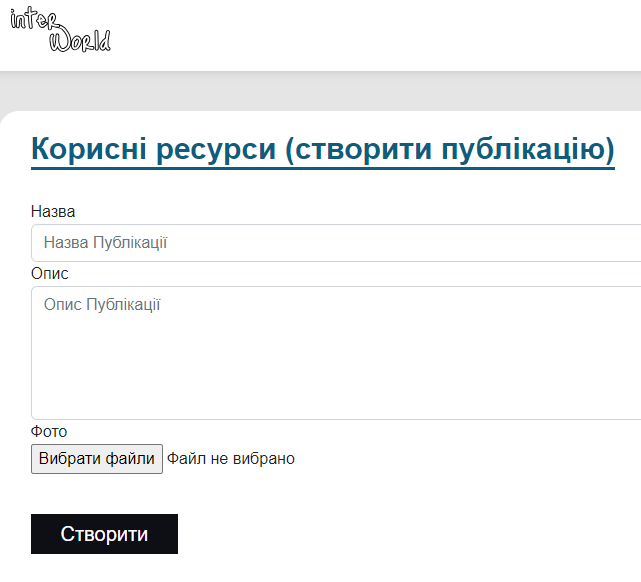


Рисунок 2.25 Сторінка додавання публікацій в розділі «Корисні ресурси»

Для виходу з акаунту користувача необхідно натиснути на посилання «Вихід» у верхній частині сторінки.

У адміністратора вебресурсу є такі можливості:

* керування інформацією про користувачів (додавання/редагування/видалення);
* керування відгуками користувачів (публікування/видалення);
* керування інформацією про категорії (додавання/редагування/видалення);
* керування інформацією у розділах «Журнал публікацій» та «Корисні ресурси» (додавання/редагування/видалення);
* керування інформацією на головній сторінці (додавання/редагування/видалення).

Адміністратор має можливість додавати/видаляти/редагувати інформацію про користувачів. Для цього в меню потрібно перейти на сторінку «Користувачі» (рисунок 2.26). Щоб додати інформацію про нового користувача необхідно натиснути «Створити» і заповнити дані форми (рисунок 2.27.) Редагування і видалення інформації про користувачів здійснюється за допомогою відповідних кнопок навпроти кожного запису .

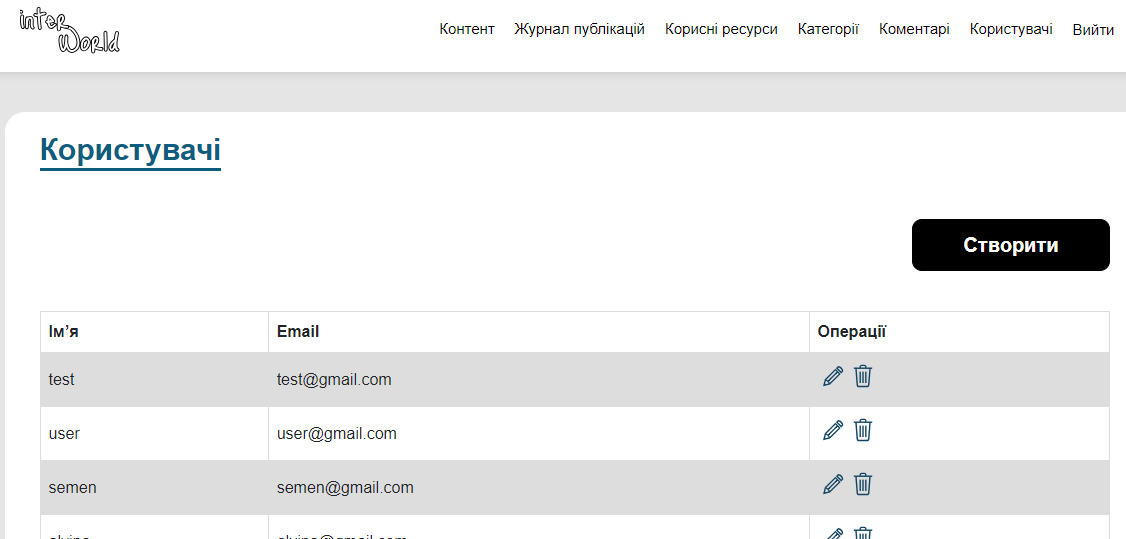


Рисунок 2.26 Сторінка «Користувачі» в режимі адміністратора

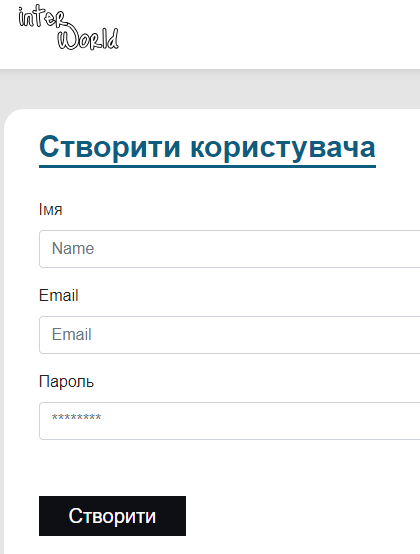


Рисунок 2.27 Сторінка додавання інформації про нового користувача

Написаний користувачем відгук з’являється на сайті після того як адміністратор дозволить публікування. Якщо відгук не відповідає нормам вебресурсу, то він може бути видалений (рисунок 2.28).

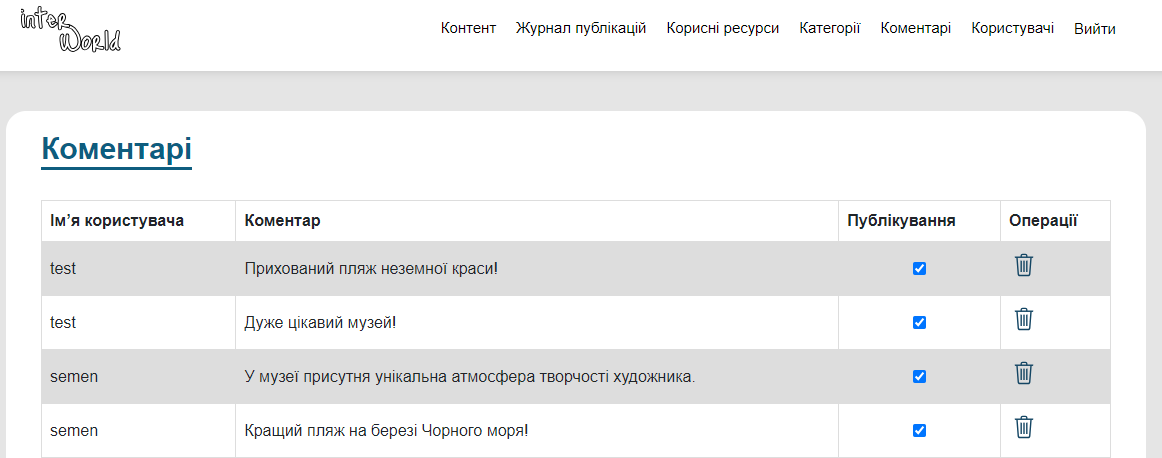


Рисунок 2.28 Сторінка «Коментарі» у режимі адміністратора

Адміністратор може додавати нові категорії публікацій на сторінці «Категорії» (рисунок 2.29). Для цього необхідно натиснути на кнопку створити і заповнити форму (назву та опис категорії) (рисунок 2.30). Видалення та редагування здійснюється відповідними інструментами в полі «Операції».

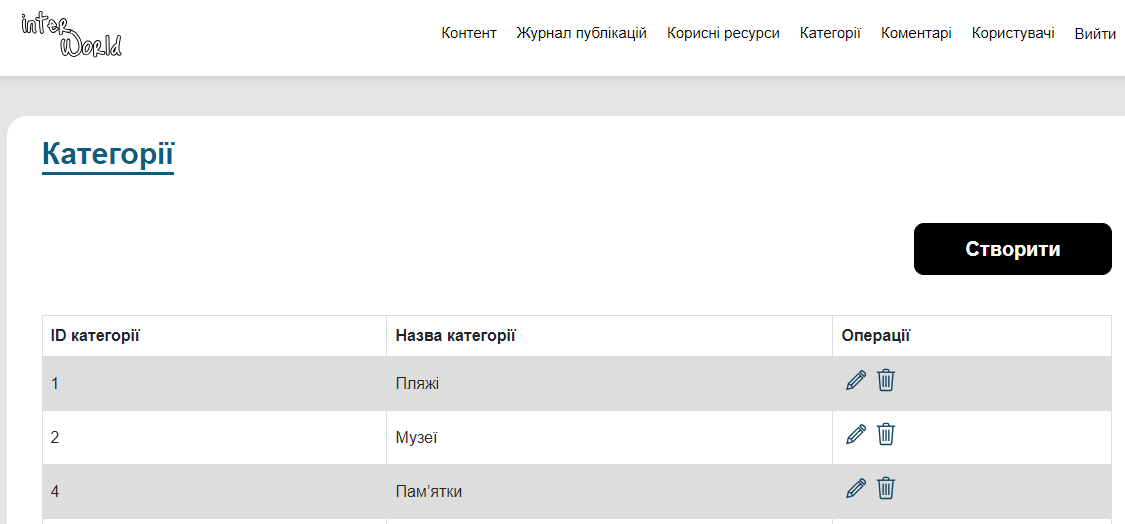


Рисунок 2.29 Сторінка «Категорії» у режимі адміністратора

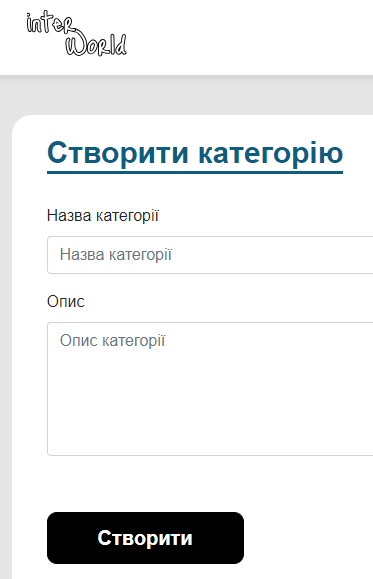


Рисунок 2.30 Сторінка додавання інформації про нову категорію

Керування інформацією в розділах «Журнал публікацій» та «Корисні ресурси» відбувається так само як і в режимі користувача. Відмінність в тому, що адміністратор надає дозвіл на публікування дописів користувачів у цих розділах.

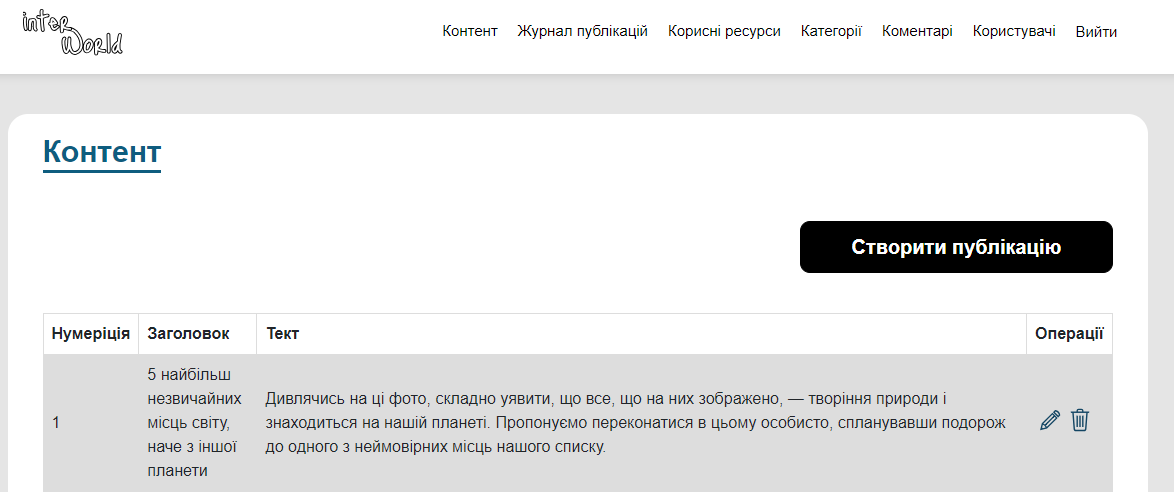
Адміністратор має ще доступ до керування інформацією, яка відображається на головній сторінці вебресурсу. На сторінці «Контент» можна створити нову статтю або відредагувати чи видалити вже опубліковані (рисунки 2.31-2.32).

Рисунок 2.31 Сторінка «Контент» у режимі адміністратора

* 1. **Тестування проєкту**

Для виявлення помилок та усунення недоліків програмного продукту було ретельно протестовано всі його функціональні можливості:

* в режимі користувача:
* реєстрація/авторизація в системі (рисунки 2.35-2.36),
* керування інформацією у розділах «Журнал публікацій» та «Корисні ресурси» (додавання/редагування/видалення) (рисунки 2.37-2.41),
* написання відгуків (рисунки 2.42-2.43),
* вихід з акаунту;
* в режимі адміністратора:
* авторизація в системі (рисунок 2.44),
* керування інформацією про користувачів (додавання/редагування/видалення),
* керування відгуками користувачів (публікування/видалення) (рисунки 2.45-2.46);
* керування інформацією про категорії (додавання/редагування/видалення),
* керування інформацією у розділах «Журнал публікацій» та «Корисні ресурси» (додавання/редагування/видалення),
* керування інформацією на головній сторінці (додавання/редагування/видалення),
* вихід з акаунту.

Нижче наведено приклади тестування.

На всіх формах вебресурсу присутня валідація даних. Так, наприклад, при реєстації перевіряється чи заповнив користувач всі поля (рисунок 2.33), а також чи коректна введена електронна адреса і чи достатньо складний пароль (рисунок 2.34).

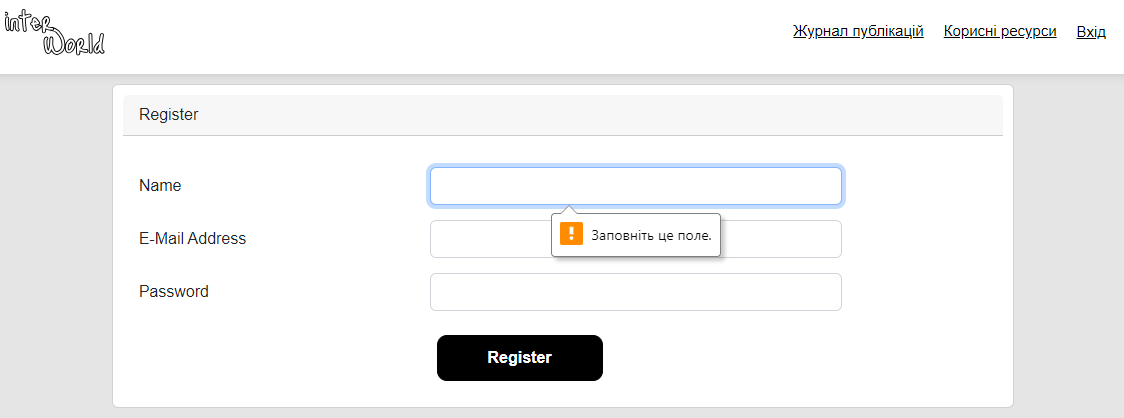


Рисунок 2.33 Виведення повідомлення, якщо користувач не заповнив поля реєстрації

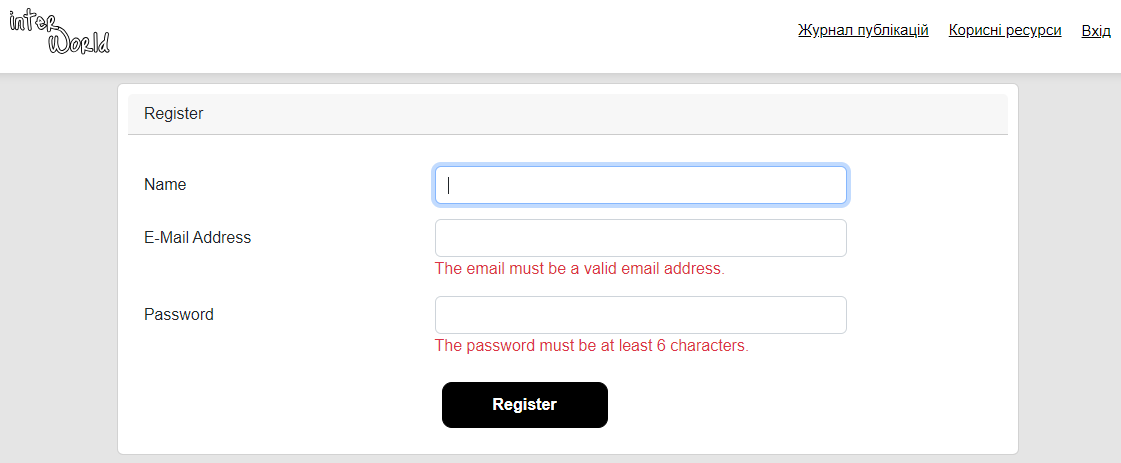


Рисунок 2.34 Виведення повідомлення, якщо введено некоректну електронну адресу або короткий пароль

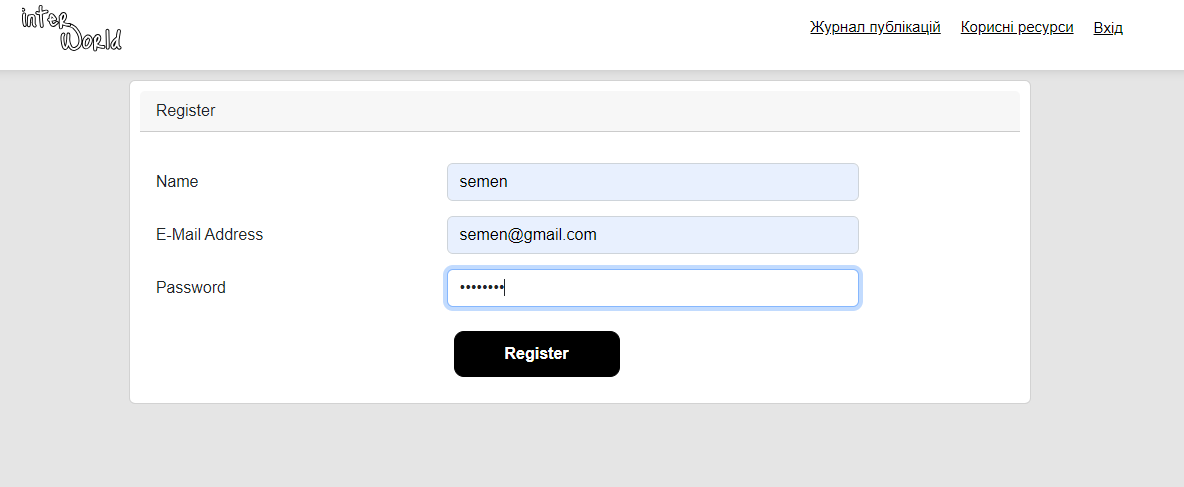


Рисунок 2.35 Реєстрація нового користувача у системі

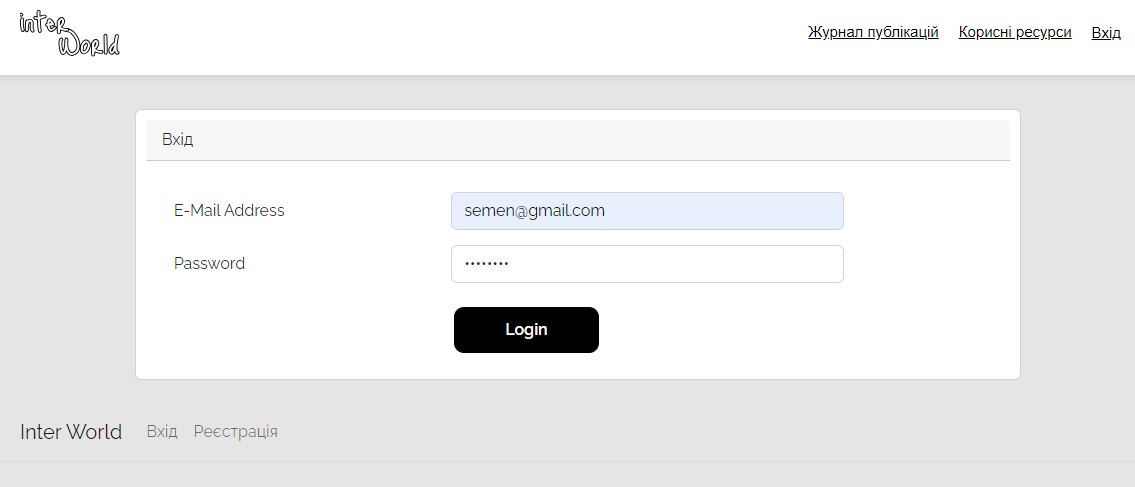


Рисунок 2.36 Авторизація користувача у системі

Після реєстрації та авторизації було протестовано додавання публікацій користувачем. Якщо користувач авторизувався вперше, то в розділах «Журнал публікацій» та «Корисні ресурси» в нього не буде публікацій (рисунки 2.37-2.38).

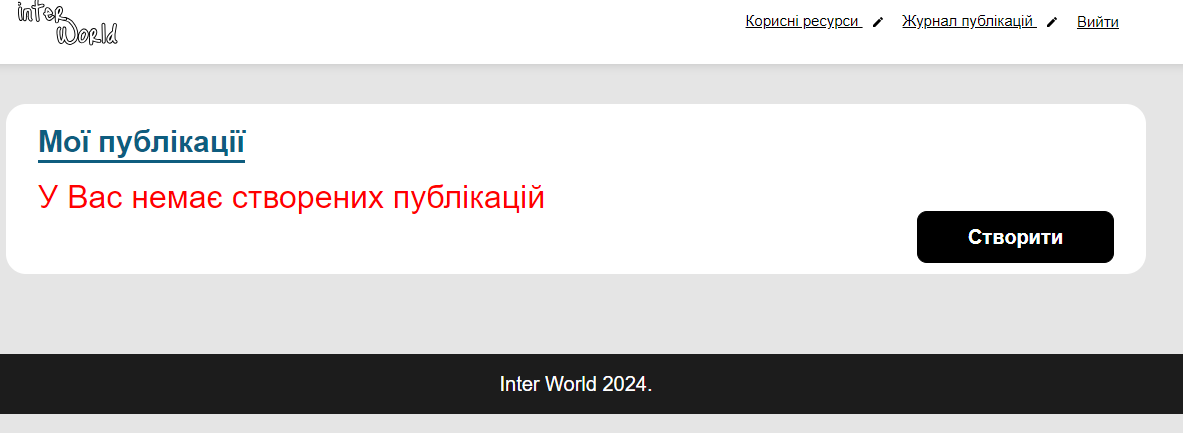


Рисунок 2.37 Сторінка «Журнал публікацій»

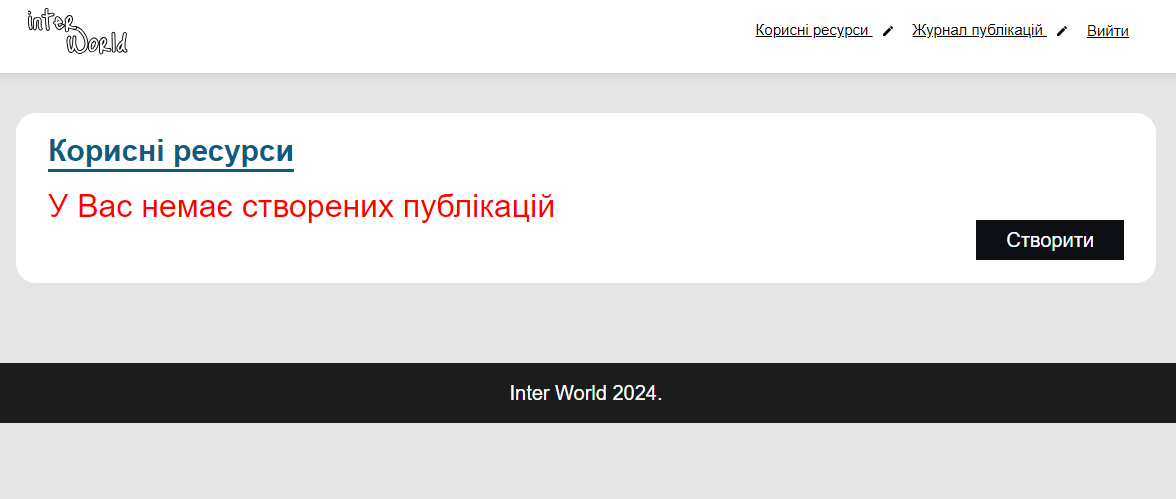


Рисунок 2.38 Сторінка «Корисні ресурси»

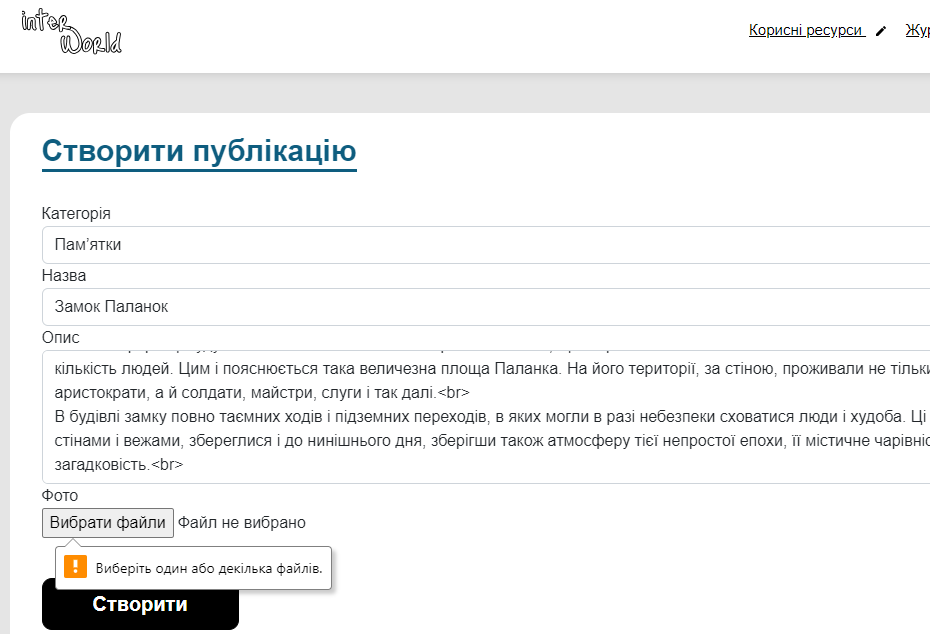


Рисунок 2.39 Виведення повідомлення, якщо користувач не завантажив зображення для нової публікації

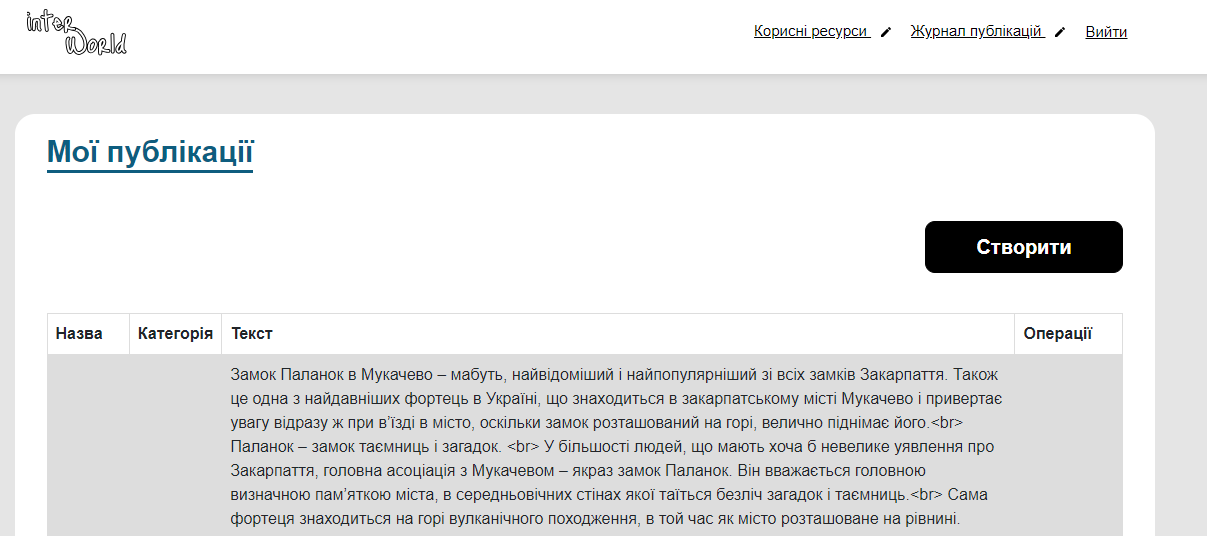
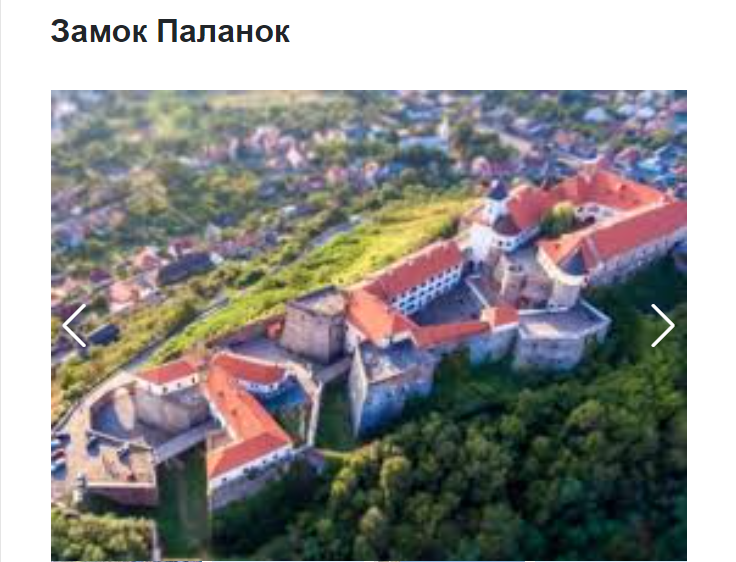


Рисунок 2.40 Сторінка «Журнал публікацій» після додавання нової публікації





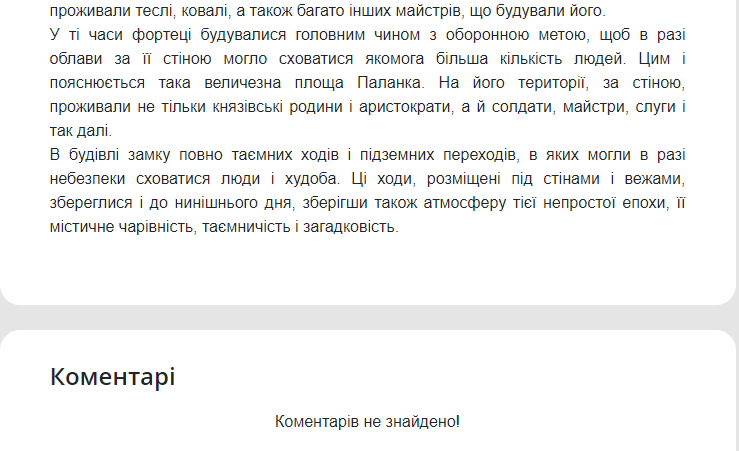


Рисунок 2.41 Попередній перегляд публікації

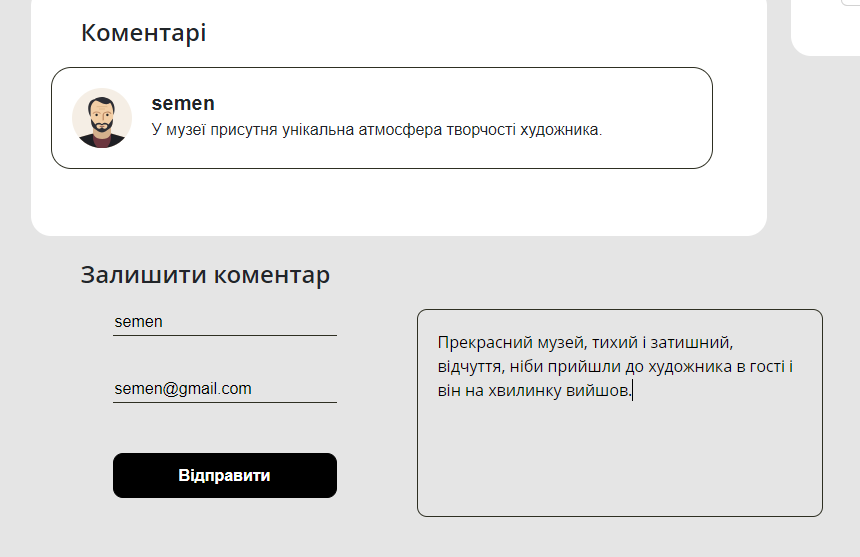


Рисунок 2.42 Додавання відгуку

Щоб відгук користувача з’явився на сайті його публікацію повинен підтвердити адмін (рисунки 2.43-2.45).

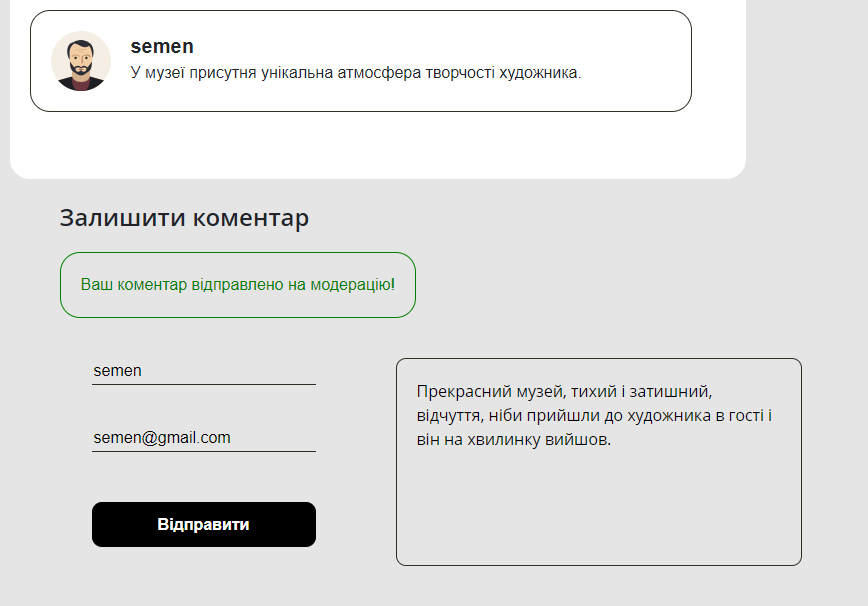


Рисунок 2.43 Виведення повідомлення про те, що відгук відправлено на модерацію

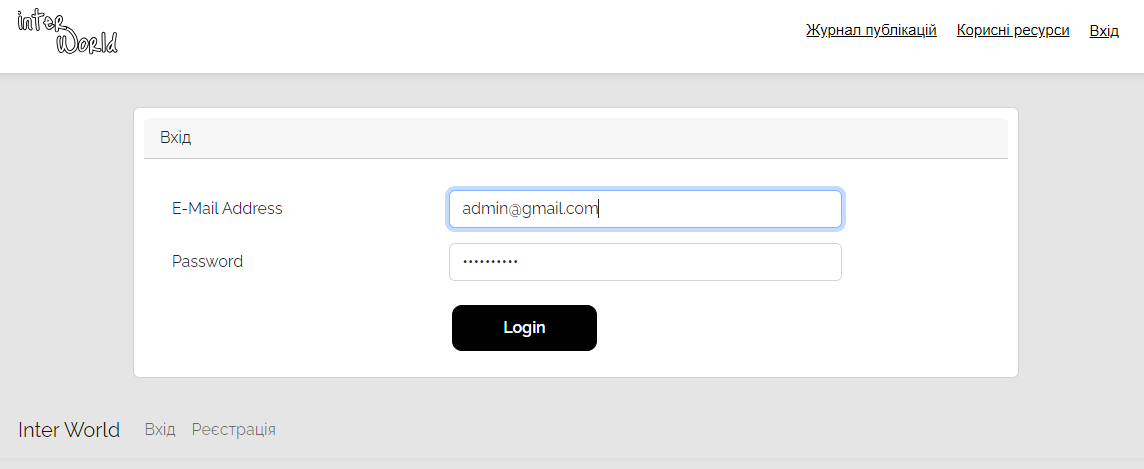


Рисунок 2.44 Авторизація адміністратора

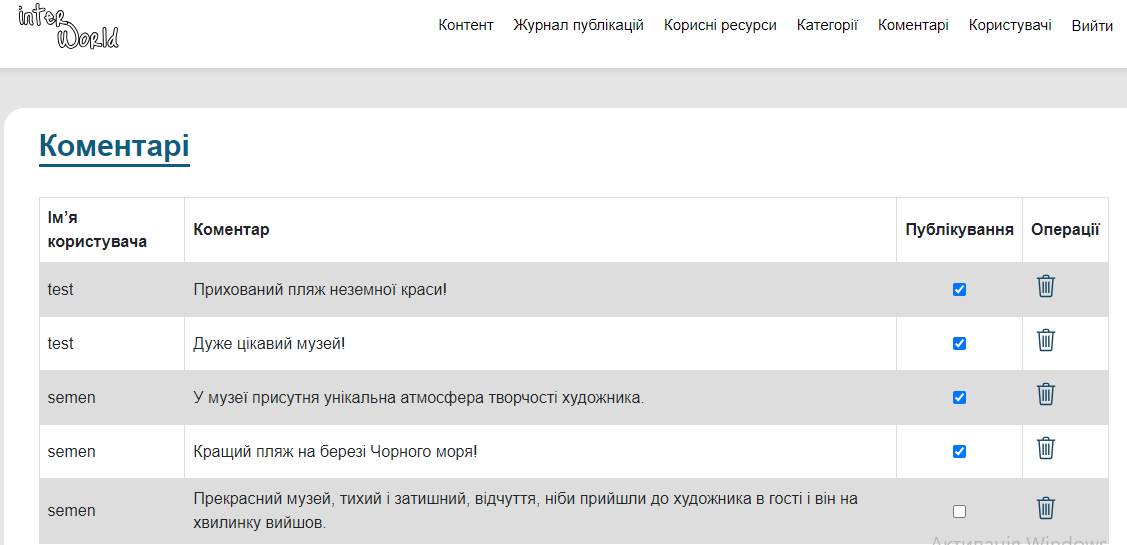


Рисунок 2.45 Відгук користувача ще не опублікований адміном

Після підтвердження публікації відгук користувача з’являється на сайті (рисунок 2.46).

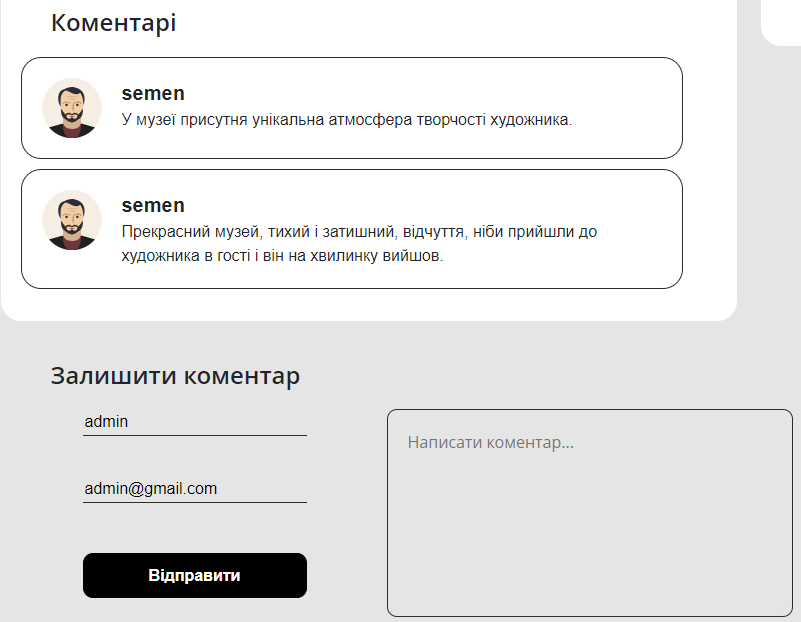


Рисунок 2.46 Відгук користувача опубліковано на сайті

Далі наведені приклади тестування інших можливостей адміністратора.

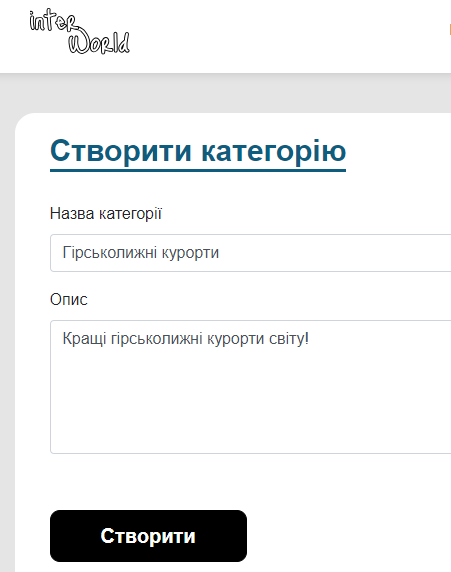


Рисунок 2.47 Додавання інформації про нову категорію «Гірськолижний курорт»

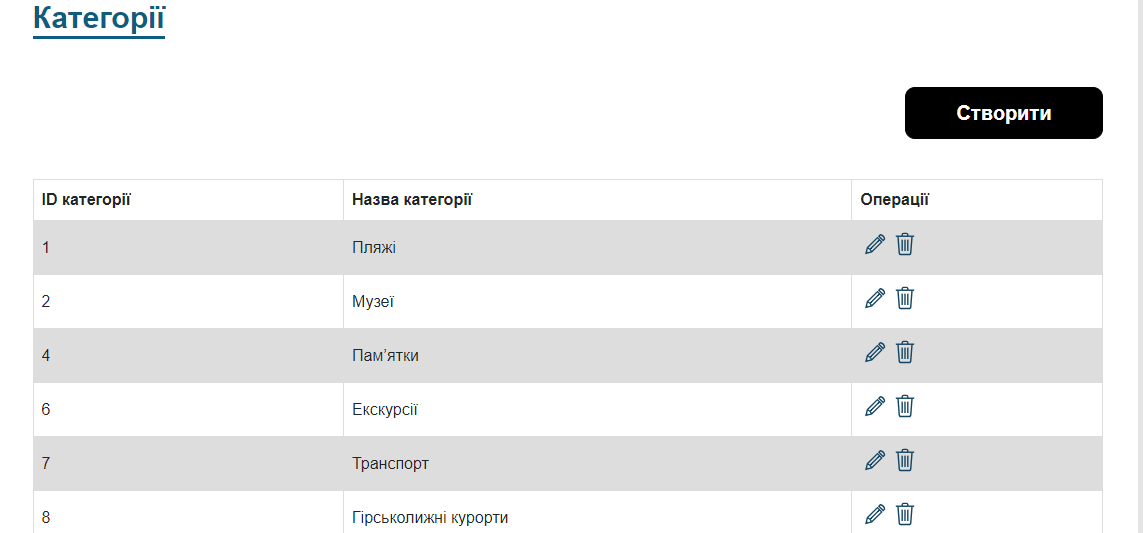


Рисунок 2.48 У списку категорій з’явилася нова категорія «Гірськолижний курорт»

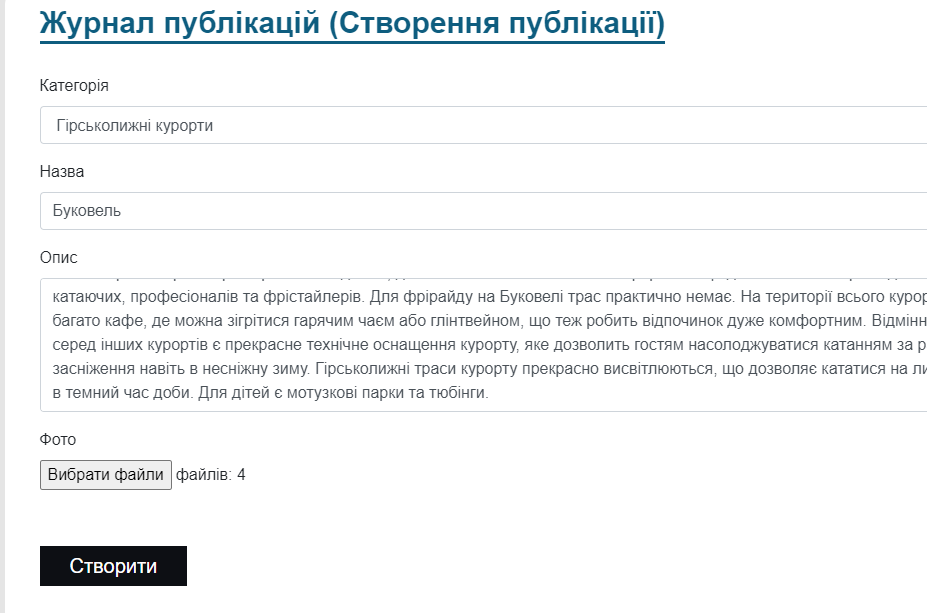


Рисунок 2.49 Додавання нової публікації у режимі адміністратора



Рисунок 2.50 Підтвердження публікування у режимі адміністратора

**3 ЕКОНОМІЧНА ЧАСТИНА**

**3.1** **Характеристика роботи та призначення роботи**

Для будь-якої людини, яка збирається в подорож щоб отримати багато приємних вражень важливим є ретельно спланувати бюджет поїздки, розважальну програму, подумати про те, де варто побувати. Тому й виникла ідея розробити ресурс, на якому люди зможуть ділитися своїми враженнями після відвідування того чи іншого містя створюючи публкації.

У рамках дипломного проєкту необхідно розробити вебресурс для створення публікацій «Цікаві містя світу».

**3.2 Витрати на створення програмного продукту**

Розрахунок вартості витрат на матеріали

Розрахунок виконуємо виходячи з витрат на допоміжні матеріали, перелік яких приведений в таблиці 3.1.

Таблиця 3.1 Допоміжні матеріали

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Найменування  Матеріалу | Виробник і  Марка | Одиниця  Виміру | Норма  затрат | Ціна за одиницю,  грн. | Вартість,  грн. |
| Картридж | CANON 725 | шт. | 1 | 1499 | 1499 |
| Оптичний диск CD–R | VERBATIM Extra Protection 700MB (43351) | шт. | 3 | 12 | 36 |
| Офісний папір формату А4 | Maestro standart, 80 г, 500 листів | пач. | 1 | 170 | 170 |
| Паперовий конверт А5 | Kraft Paper | шт. | 2 | 8 | 16 |
| Всього | | | | | 1721 |

Загальна вартість допоміжних матеріалів складає Звм = 1721 грн.

Розрахунок витрат на оплату праці працівника

Для нарахування заробітної плати працівника приймаємо погодинну форму оплати.

|  |  |
| --- | --- |
| ЗПпог = t \* ТЗПГОД ,  (3.1) |  |

де t – час на створення програмного продукту, год.;

ТЗПГОД – тарифна ставка працівника певного розряду за годину праці, грн./год.;

Дані про трудомісткість операцій, виконуваних в процесі створення програмного продукту, приведені в таблиці 3.2.

Таблиця 3.2 Трудомісткість операцій

|  |  |
| --- | --- |
| Найменування операції | Витрати, год |
| Постановка задачі | 2 |
| Проєктування програмного інтерфейсу | 10 |
| Реалізація програмних модулів | 50 |
| Відлагодження | 20 |
| Тестування | 20 |
| Оформлення документації | 20 |
| Всього | 122 |

Згідно з єдиною тарифною сіткою в 2024 році тарифна ставка за годину програміста 16-го розряду становить 57,07 грн/год. Виходячи з цього, витрати на оплату праці програміста згідно з формулою (3.1) дорівнюватимуть:

 = 57,07 \* 122 = 6962,54 грн.

Нарахування на заробітну плату складають 22%

ЗНАРАХ= \* 0,22= 6962,54 \* 0,22 = 1531,76 грн

Тоді загальні витрати на оплату праці складають

ЗПзаг =+ ЗНАРАХ = 6962,54 + 1531,76 = 8494,3‬ грн

**3.3 Витрати на утримання та експлуатацію ПЕОМ**

Річний фонд часу роботи ПЕОМ згідно прийнятої в Україні норми тривалості робочого часу на 2024 рік з урахуванням воєнного стану складає 2000 годин.

Балансову вартість ПЕОМ обчислимо за формуло (3.2):

(3.2)

|  |
| --- |
| СБАЛ = СРИН + ЗвСТ , |

де СРИН – ринкова вартість ПЕОМ, грн.;

ЗвСТ – витрати на доставку й встановлення ПЕОМ, грн.

Витрати на доставку й встановлення ПЕОМ складають 10% від ринкової вартості ПЕОМ. Покладаємо ринкову вартість ПЕОМ рівною СРИН = 25000 грн і отримуємо балансову вартість ПЕОМ:

СБАЛ = СРИН + ЗвСТ = 25000 + 2500 = 27500 грн.

Для нарахування амортизації використовуємо прямолінійний метод.

ЗАМ = СБАЛ / ТЕКСПЛ. ,

(3.3)

де СБАЛ – балансова вартість ПЕОМ, грн.;

ТЕКСПЛ. – термін експлуатації (років).

ТЕКСПЛ. приймаємо 5 років.

Таким чином річні амортизаційні відрахування складають:

ЗАМ = 27500 / 5 = 5500 грн.

Вартість електроенергії, споживаної за рік, визначаємо за формулою (3.4):

(3.4)

ЗЕЛ = PПЕОМ \* ТПЕОМ \* СЕЛ \* А ,

де PПЕОМ – сумарна потужність ПЕОМ, кВт;

ТПЕОМ – річний час роботи ПЕОМ, год.;

СЕЛ – вартість 1 кВт \* год. електроенергії, 2,64 грн.;

СЕЛ = 2,64 грн;

А — коефіцієнт інтенсивного використання потужності ПЕОМ.

Сумарна потужність ПЕОМ PПЕОМ = 0,06 кВт. Приймаємо коефіцієнт інтенсивного використання потужності ПЕОМ рівним А = 0,9 і обчислюємо вартість електроенергії, споживаної за 2000 годин:

ЗЕЛ = 0,06 \* 2000 \* 2,64 \* 0,9 = 285,12 грн.

Інші витрати на експлуатацію ПЕОМ (в тому числі й витрати на поточний та профілактичний ремонт) складають 10% від балансової вартості ПЕОМ і дорівнюють:

ЗПР = 0,1 \* СБАЛ = 0,1 \* 27500 = 2750.

Повні витрати на експлуатацію ПЕОМ визначаємо за формулою (3.5):

ЗПЕОМ = ЗЗП + ЗАМ + ЗЕЛ + ЗПР ,

(3.5)

де ЗЗП – витрати на заробітну плату, грн;

ЗАМ – амортизаційні відрахування, грн;

ЗЕЛ – витрати на електроенергію, грн;

ЗПР – інші витрати на експлуатацію ПЕОМ, грн.;

тоді:

ЗПЕОМ = 6962,54 + 5500+ 285,12 + 2750 = 15497,66 грн.

Вартість машинної години визначаємо за формулою(3.6):

СГОД = ЗПЕОМ / ТПЕОМ ,

(3.6)

де ЗПЕОМ – повні витрати на експлуатацію ПЕОМ, грн.;

ТПЕОМ – час роботи ПЕОМ, год.

СГОД = 15 497,66 / 122 = 127,03 грн./год.

Витрати на оплату машинного часу при відлагодженні програми визначаємо за формулою (3.7):

|  |  |
| --- | --- |
| = СГОД \* ТВ , |  |

де СГОД – вартість машинної години, грн./год.;

(3.7)

ТВ – фактичний час відлагодження програмного продукту на ПЕОМ, год.

З таблиці 3.2 видно, що ТВ = 20 год., тоді:

 = СГОД \* 20 = 127,03 грн \* 20 = 2540,6 грн.

**3.4 Загальновиробничі витрати**

Витрати на освітлення, опалення, комунальні послуги та інші в нашому випадку приймаємо 30% від витрат на оплату праці

|  |  |
| --- | --- |
| ЗЗАГ = \* 30% / 100 ,  (3.8) |  |

де  – витрати на оплату праці програміста, грн.

ЗЗАГ = 6962,54 \* 0,3 = 2088,76 грн.

**3.5 Обчислення собівартості програмного продукту**

Для обчислення складаємо калькуляцію собівартості продукту за статтями витрат в таблиці 3.3

Таблиця 3.3 Собівартість програмного продукту за статтями витрат

|  |  |  |
| --- | --- | --- |
| № п/п | Статті витрат | Сума, грн. |
| 1 | Сировина та матеріали | 1721 |
| 2 | Основна заробітна плата | 6962,54 |
| 3 | Нарахування на зарплату | 1531,76 |
| 4 | Витрати на оплату машинного часу | 2540,6 |
| 5 | Загальновиробничі витрати | 2088,76 |
| 6 | Повна собівартість | 14844,66 |

**3.6 Обчислення гуртової ціни програмного продукту**

Гуртову ціну програмного продукту визначаємо за формулою (3.9).

ЦРЕАЛ = СПОВ + РН

(3.9)

де СПОВ – повна собівартість виробу, грн.;

РН – нормативний рівень рентабельності, %.

В нашому випадку РН приймаємо в розмірі 20% від повної собівартості програмного продукту.

РН = СПОВ \* (РН / 100)

РН = 14844,66 \* 0,20 = 3265,82 грн.

ЦРЕАЛ = 14844,66 + 3265,82 = 18110,48 грн.

* 1. **Висновки економічної доцільності проєкту**

Розроблений програмний продукт буде надавати користувачам можливість ділитися враженнями з іншими про різні місця, в яких вони побували під час своїх подорожей. Написані публікації будуть корисні людям, які перебувають на етапі планування поїздки.

# 4 Охорона праці

# 4.1 Аналізу потенційних небезпек, відносно до обладнання, устаткувань або приміщень, що розглядаються в дипломному проєкті

* Потенційні небезпеки фізичного характеру:
* Наявність (підвищеного значення) напруги в електричному колі, при порушенні ізоляції проводів може спричинити до електротравм для людини різного ступеню важкості.
* Механічне травмування, що може бути пов’язане з падінням якогось предмета для прикладу на ногу, що може спричинити до синяка, або перелому.
* Порушення розу як наслідок незадовільних ергономічних характеристик моніторів.
* Потенційні небезпеки хімічного характеру:
* Небезпека отруєння в наслідок аварії на поряд розташованому хімічному заводі, що може призвести до витоку газоподібних хімікатів.
* Небезпека отруєння в наслідок аварії при експлуатації газонаповнених трансформаторів, газобалонного обладнання або речовинами, що виділяються при роботі персональних комп’ютерів та копіювальної техніки.
* Потенційні небезпеки психофізіологічного характеру:
* Незадовільна організація робочого місця.
* Недостатнє врахування вимог ергономіки.
* Нераціональне розташування технологічного обладнання та захаращеність робочої зони або підвищені психофізіологічні навантаження при роботі з персональним комп’ютером, які виникають через нераціональну організацію робочого часу.
* Потенційні небезпеки, що пов’язані з порушенням санітарно-гігієнічних умов:
* Незадовільний стан повітряного середовища.
* Незадовільний рівень освітленості, підвищений рівень шуму, вібрації, електромагнітних полів або іонізуючих випромінювань.
* Потенційні небезпеки, що пов’язані з порушенням правил пожежної безпеки:
* Коротке замикання, що може призвести до виникнення пожеж.
* Відсутність або неправильний вибір типу та необхідної кількості первинних засобів гасіння пожеж (вогнегасників).
* Не герметичність систем живлення, порушення герметичності газового обладнання, може стати причиною поширення пожежі.

## 4.2 Аналіз стану охорони праці. Розрахунок кількість комп’ютеризованих робочих місць та кількіть вогегасників

Основним елементом виробничої структури підприємства є робоче місце, від правильної організації якого залежать ефективність і безпека кожної технологічної операції та виробничого процесу в цілому, а також безпека життя та здоров’я людей. Тому важливо, щоб організація та стан робочих місць забезпечували зручні та безпечні дії з матеріалами, приладами.

Для аналізу взято кімнату, де відбувалась розробка дипломного проєкту. Дано приміщення з параметрами ***а***= 7,8 м, ***b***= 7,7 м, ***h***= 4,3 м. Перед визначенням кількості робочих місць, які можна розмістити у приміщенні, слід визначити його площу та об’єм:

|  |  |
| --- | --- |
| Sпр = а \* b, | (4.1) |
| Vпр. = а \* b \* h, | (4.2) |

Sпр = а \* b = 7,8 \* 7,7 = 60,06 м2

Vпр. = а \* b \* h = 7,8 \* 7,7 \* 4,3 = 258,26 м3

Також потрібно врахувати розміри меблів на комп’ютеризованих робочих місцях, зокрема робочого столу (таблицю 4.1).

Відповідно до НПАОП 0.00.1.28–10 рекомендовані розміри столу для робочого місця з ВДТ становлять: висота – 725 мм, ширина – 600–1400 мм, глибина – 800–1000 мм.

Таблиця 4.1 Параметри устаткування

|  |  |  |  |
| --- | --- | --- | --- |
| Параметри устаткування | Стіл | Крісло | Шафа |
| Довжина, (м) | 0,9 | 0,7 | 0,6 |
| Ширина, (м) | 0,8 | 0,7 | 0,8 |
| Висота, (м) | 0,725 | 0,6 | 1,8 |
| Займана площа, (м2) | 0,72 | 0,49 | 0,48 |
| Займаний обсяг, (м3) | 0,522 | 0,294 | 0,864 |

Обрахуємо площу за формулою (4.3) та об’єм за формулою (4.4) устаткування, яке розташоване на одному робочому місці :

|  |  |
| --- | --- |
| Sуст. = Sстола+ Sкрісла + ½\*Sшафи, | (4.3) |
| Vуст. = Vстола+ Vкрісла + ½\*Vшафи, | (4.4) |

Sуст. = Sстола+ Sкрісла + ½\*Sшафи = 0,72 + 0,49 + 0,48 / 2 = 1,45 м2;

Vуст. = Vстола+ Vкрісла + ½\*Vшафи = 0,435 + 0,168 + 0,432 / 2 = 1,68 м3;

Кількість комп’ютеризованих робочих місць, яку можна розмістити в заданому приміщенні за умови, що на одне таке робоче місце повинно доводитися не менш ніж 6 м2 вільної площі та 20 м3 за об’ємом, не зайнятим устаткуванням розраховується за однією з наступних формул:

|  |  |
| --- | --- |
| **Nр.м. по S** = S / (Sуст.+ 6) | (4.5) |
| **Nр.м. по V** = V / (Vуст.+ 20) | (4.6) |

Розрахуємо кількість робочих місць за заданими площею та об’ємом:

**Nр.м. по S** = 60,06 / (1,45 + 6) = 8,06;

**Nр.м. по V** = 258,26 / (1,68 +20) = 11,91;

З отриманих результатів приймаємо мінімальне значення, округляючи його до найближчого меншого цілого числа. В даному випадку виходить 8.

Відповідно до правил експлуатації та типових норм належності вогнегасників та НОРМ №6 (Приміщення, у яких розміщені електронно‑обчислювальні машини, слід оснащувати переносними газовими вогнегасниками з розрахунку один вогнегасник ВВК-1,4 на три ЕОМ, але не менше ніж один вогнегасник зазначеного типу на приміщення. В даному приміщенні повинно розміщуватись — 3 вогнегасники.

На рисунку 4.1 наведено схему розміщення 8 комп’ютеризованих робочих місць у приміщенні з параметрами ***а***= 7,8 м, ***b***= 7,7 м, ***h***= 4,3 м, із врахуванням наступних вимог:

* робочі місця з ВДТ розміщуються на відстані не менше 1 м від стіни зі світловими прорізами;
* відстань між бічними поверхнями ВДТ має бути не менше за 1,2 м;
* відстань між тильною поверхнею одного ВДТ та екраном іншого не повинна бути меншою за 1,5 м;
* прохід між рядами робочих місць має бути не меншим ніж 1 м.;
* одна шафа необхідна для двох робочих місць.

Найкраще розмістити комп’ютеризовані робочі місця рядами вздовж стіни з вікнами. Це дасть змогу виключити дзеркальне відбиття на екрані ВДТ джерел природного світла та їх потрапляння у поле зору операторів, що погіршує умови їх зорової роботи.

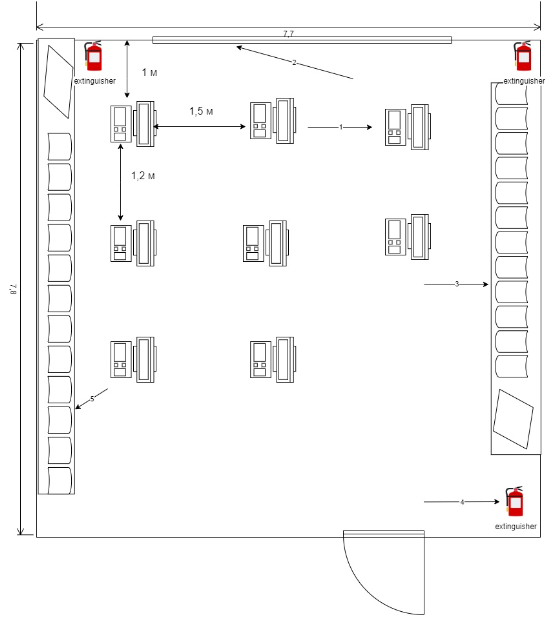


Рисунок 4.1 План виробничого приміщення з комп’ютеризованими робочими місцями

1 — комп’ютеризоване робоче місце з ВДТ,

2 — сонцезахисні жалюзі,

3 — шафи для зберігання програмного забезпечення,

4 — шафи, для зберігання документації та фахової літератури,

5 — вогнегасник.

## 4.3 Заходи щодо поліпшення умов з охорони праці та виробничої санітарії у досліджуваному приміщені

Для поліпшення умов праці в даному приміщенні рекомендується періодично переривати роботу за екраном монітора на регламентовані перерви, які встановлюються для забезпечення працездатності і збереження здоров’я, або замінювати іншою роботою з метою зменшення робочого навантаження біля екрану.

Для зменшення напруження та фактору відволікання слід зробити максимальну естетичну гармонійність кольорової схеми обладнання в приміщенні.

Важливо конструктивне розташування елементів з метою створення полегшеного режиму роботи.

Для зниження впливу шкідливих чинників на користувачів, рекомендується застосовувати монітори зі зниженою випромінювальною здатністю, встановлювати захисні екрани, і дотримуватися регламентованих режимів праці та відпочинку.

Для ергономіки слід застосовувати спокійні та естетично гармонійні кольорові схеми в оформленні інтер’єру та стін приміщення, де відбувається робочий процес.

Рекомендується встановити вентиляційну систему для забезпечення чистоти повітряного середовища в приміщенні, де відбувається робочий процес.

Комп’ютери потрібно розташувати уникаючи відблисків природнього освітлення в моніторах.

## 4.4 Інструкція з охорони праці при роботі з ПЕОМ та відео‑дисплейними терміналами

**4.4.1 Загальні положення**

Ця інструкція містить обов’язкові вимоги для дотримання фахівцем з інформаційних технологій (надалі – фахівець) правил з охорони праці при виконанні робіт, визначених його функціональними обов’язками.

Інструкція розроблена відповідно до:

* Положення про розробку інструкцій з охорони праці, затвердженого наказом Держнаглядохоронпраці від 29.01.1998 р. № 9 (в редакції наказу Міністерства соціальної політики України від 30.03.2017 р. №  526 ).
* Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого  наказом Держнаглядохоронпраці від 26.01.2005 р. № 15, з наступними змінами.
* Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями, затверджених наказом Міністерства  
  соціальної політики України від 14.02.2018  № 207.
* Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин (ДСанПІН 3.3.2.007-98), затверджених постановою Головного державного санітарного лікаря України від 10.12.1998 N 7.

Фахівцем з інформаційних технологій  може працювати особа, яка досягла 18-річного віку, має вищу освіту відповідного напрямку, володіє екранним пристроєм на рівні стандартного прикладного програмного забезпечення, основами мережевих технологій, пройшла вступний інструктаж з питань охорони праці, первинний інструктаж на робочому  місці, інструктаж з пожежної безпеки, а також пройшла медичний огляд та не має протипоказань до виконання обов’язків за станом здоров’я.

З метою дотримання правил охорони праці фахівець повинен:

* керуватися вимогами правил внутрішнього трудового розпорядку, інструкцій з охорони праці, пожежної безпеки та електробезпеки;
* суворо дотримуватись правил особистої гігієни та санітарних правил;
* не допускати на своє робоче місце сторонніх осіб;
* не приступати і не виконувати роботу у стані алкогольного, наркотичного або медикаментозного сп'яніння, у хворобливому або стомленому стані;
* знати місця розташування аптечки та первинних засобів пожежогасіння і вміти користуватися ними.

Під час роботи на фахівця можуть впливати такі шкідливі виробничі фактори, як: неякісне освітлення робочого місця, враження електричним струмом, випромінювання від екранного пристрою та засобів оргтехніки, нервово-психологічні та емоційні навантаження, перенапруження зорових аналізаторів, гіподинамія, інші негативні фактори.

Робочий стіл для роботи фахівця підбирається достатнього розміру, з поверхнею, що має низьку відбивну здатність та допускає можливість зручного розміщення екрана, клавіатури, документів і відповідного устаткування. Крісло має бути стійким, легко рухатися та займати зручне положення. Сидіння повинно регулюватися по висоті, а спинка сидіння - як по висоті, так і по нахилу.

Екран повинен вільно повертатися і нахилятися відповідно до потреби працівника, зображення на ньому має бути стабільним, яскравість та контрастність символів повинні легко регулюватися.

* + 1. **Вимоги безпеки перед початком роботи**

4.4.2.1 Перед початком роботи фахівець повинен отримати інструктаж від безпосереднього керівника.

4.4.2.2 Переконатися візуально у відсутності механічних ушкоджень корпусів екранного пристрою та засобів оргтехніки, у наявності інженерно-технічних засобів безпеки, аптечки і засобів пожежогасіння.

4.4.2.3 Впевнитись в якості освітлення, наявності рекомендованого контрасту між екраном і навколишнім середовищем. Робоче місце має бути розташованим таким чином, щоб природнє світло падало збоку, переважно зліва.

4.4.2.4 Перевірити справність шнурів живлення, вимикачів, розеток, штепсельних з’єднань, надійність роботи вентиляційної системи.

4.4.2.5 Очистити екранний пристрій від пилу та інших забруднень.

4.4.2.6 Повідомити безпосереднього керівника про несправність екранного та периферійних пристроїв, засобів оргтехніки, меблів, приладів, електропроводки, іншого обладнання та не приступати до роботи до повного усунення недоліків.

* + 1. **Вимоги безпеки під час виконання роботи**

4.4.3.1 Утримувати в порядку і чистоті робоче місце. Не захаращувати проходи до робочого місця сторонніми предметами і речами, які зменшують тепловіддачу засобів оргтехніки та іншого обладнання.

4.4.3.2 Стежити за справністю екранного пристрою, засобів оргтехніки та іншого устаткування, дотримуватися правил їх експлуатації.

4.4.3.3 При тривалій відсутності на робочому місці відключати від електромережі засоби оргтехніки та інше устаткування, крім обладнання, яке використовується цілодобово.

4.4.3.4 Забезпечувати щоденне вологе прибирання та щогодинне провітрювання робочого приміщення.

4.4.3.5 Контролювати виконання рекомендацій щодо:

* проведення регламентованих законодавством і правилами внутрішнього розпорядку короткочасних перерв в роботі;
* проведення комплексу вправ для очей, для рук, для хребта (вироблення правильної постави), для поліпшення мозкового кровообігу, а також сеансів психофізіологічного розвантаження.

4.4.3.6 Не допускається:

* виконання технічного обслуговування і ремонту екранного пристрою безпосередньо на робочому місці працівника;
* відключення захисних пристроїв, самочинне проводення зміни у конструкції екранного пристрою;
* робота з екранними пристроями, у яких під час роботи виникають нехарактерні сигнали, нестабільне зображення на екрані та інші несправності;
* робота при знятих і пошкоджених кожухах захисту засобів оргтехніки та іншого обладнання;
* доторкатись руками до рухомих частин засобів оргтехніки та іншого обладнання.
  + 1. **Вимоги безпеки після закінчення роботи**

Після закінчення роботи фахівець зобов’язаний:

* відключити від електромережі екранний пристрій, засоби оргтехніки та інше устаткування, крім обладнання, яке використовується цілодобово (апарати факсимільного зв’язку, мережеві сервери тощо);
* впорядкувати робоче місце, провітрити приміщення;
* привести себе у порядок, вимити руки і обличчя та перевдягнутись.
  + 1. **Вимоги безпеки в аварійних ситуаціях**

4.4.5.1 При виникненні аварії або ситуації, яка може призвести до неї потрібно вжити всіх можливих заходів з її усунення, попередити оточуючих про небезпеку, повідомити про подію керівництво підприємства та діяти згідно з планом ліквідації аварії.

4.4.5.2 У разі несправності електрообладнання, яке знаходилось в експлуатації, перериванні подачі електроенергії чи появі незвичного шуму, диму слід негайно зупинити його роботу.

4.4.5.3 У випадку пожежі необхідно:

* вжити заходів щодо евакуації працівників із приміщення відповідно до плану евакуації та негайно повідомити пожежну службу;
* сповістити про подію керівництво підприємства;
* приступити до ліквідації загоряння відповідно до інструкції з пожежної безпеки наявними засобами пожежогасіння.

4.4.5.4 Якщо стався нещасний випадок і є потерпілі, а також при раптовому захворюванні працівника необхідно:

* усунути дію на організм небезпечних та шкідливих факторів, які загрожують здоров’ю і життю потерпілого (звільнити його від дії електричного струму, винести із зараженої території, загасити одяг, що горить тощо);
* надати потерпілому першу допомогу (в залежності від враження - відновити прохідність дихальних шляхів, провести штучне дихання, зовнішній масаж серця, зупинити кровотечу, іммобілізувати місце перелому, наклавши пов’язку тощо) та викликати швидку медичну допомогу або ж невідкладно здійснити заходи щодо транспортування постраждалого у найближчий лікарський заклад.

4.4.5.5 В подальшому необхідно виконувати вказівки керівництва підприємства.

# ВИСНОВКИ

У рамках дипломного проєктування розроблено вебресурс для розміщення публікацій про відвідуваня цікавих мість користувачами. На даному ресурсі мандрівники можуть ділитися реальним досвідом, який бути більш достовірним та детальним порівняно з офіційними туристичними путівниками. Відгуки та поради від реальних людей допоможуть іншим краще зрозуміти, чого очікувати від певного місця. Люди, які планують подорож, можуть використовувати сайт для збору інформації про місця, куди вони хочуть поїхати. озповіді та фотографії з подорожей інших людей можуть надихнути користувачів на нові напрямки та пригоди. Це може розширити їх горизонти і допомогти відкрити нові місця, про які вони раніше не знали.

В системі реалізовано 3 категорії користувачів: адміністратор, незареєстрований користувач, зареєстрований користувач.

Звичайний користувач (незареєстрований) має можливість переглядати публікації інших користувачів.

Зареєстрований та авторизований користувач може створювати власні публікації та залишати відгуки.

Адміністратор каталогу користується всіма можливостями користувача, але крім цього керує інформацією про користувачів, модерує їх публікації та відгуки.

На вебресурсі реалізовано пошук, сортування та фільтрування публікацій.

Система має зручний та інтуїтивно зрозумілий інтерфейс і приємний дизайн.

**ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. HTML. URL: http://surl.li/vxcd (дата звернення: 15.05.24).
2. CSS. URL: https://uk.wikipedia.org/wiki/CSS (дата звернення: 15.05.24).
3. JavaScript. URL: <https://uk.wikipedia.org/wiki/JavaScript> (дата звернення: 15.05.24).
4. Bootstrap (фреймворк). URL: https://uk.wikipedia.org/wiki/Bootstrap (дата звернення: 15.05.24).
5. jQuery. URL: <https://uk.wikipedia.org/wiki/JQuery> (дата звернення: 15.05.24).
6. Php. URL: https://uk.wikipedia.org/wiki/PHP (дата звернення: 15.05.24).
7. Laravel. URL: https://uk.wikipedia.org/wiki/Laravel (дата звернення: 15.05.24).
8. Система керування базами даних MySQL. URL: https://lemon.school/blog/systema-upravlinnya-bazamy-danyh-mysql (дата звернення: 15.05.24).
9. А. Ю. Берко, О. М. Верес, В. В. Пасічник Системи баз даних та знань. Книга 2. Системи управління базами даних та знань: навч. посібник. — Львів : «Магнолія-2006», 2012. — 584 с.
10. HTML5 ― що це?. URL: https://www.armedsoft.com/ua/blog/html5-shcho-ce (дата звернення: 15.05.24).
11. Р.А. Мельник, Програмування веб-застосувань (фронт-енд та бек-енд): навч. Посібник. — Львівська політехніка, 2018. — 248 с.
12. Ерік Фрімен, Елізабет Робсон, Програмування на JavaScript. — Фабула, 2022. — 672 с.
13. Н.Александра. Методичні рекомендації до виконання дипломного проєкту. Для студентів галузь знань 12 «Інформаційні технології», спеціальність 121 «Інженерія програмного забезпечення», за освітньо-кваліфікаційним рівнем «молодший спеціаліст». Ужгород, 2019.
14. Щербан І. І. Методичні вказівки до виконання розділу «Охорона праці» в дипломних роботах (проєктах) для студентів спеціальност.і спеціальність 121 «Інженерія програмного забезпечення» за освітньо-кваліфікаційним рівнем «молодший спеціаліст». Ужгород, 2020, 14 ст.
15. Поп І.М. Методичні вказівки для виконання економічної частини дипломного проєкту для студентів спеціальності 5.05010301 «Розробка програмного забезпечення»». Ужгород, 2020 — 7 cт.
16. Наказ Мінсоцполітики від 14.02.2018 №207 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями».
17. НПАОП 0.00-1.28-10. «Правила охорони праці під час експлуатації електронно-обчислювальних машин». Затверджено Наказом Держгірпромнагляду України 26.03.2010 № 65 «Про затвердження Правил охорони праці під час експлуатації електронно-обчислювальних машин»; зареєстровано в Міністерстві юстиції України 19 квітня 2010 р.за № 293/1758.
18. Закон України «Про охорону праці». URL: https://zakon.rada.gov.ua/laws/show/2694-12 (дата звернення 27.04.2024).

**Додаток 1 Лістинг програми**

**index.blade.php**

@include('header')

<style type="text/css">

.slider{

position: relative;

z-index: 99999;

}

.slider img{

width: 100%;

height: 700px;

object-fit: cover;

}

.carousel-control-next-icon {

background-color: black;

}

.carousel-control-prev-icon {

background-color: black;

}

.block-content {

padding: 20px;

}

.block-content h2 {

font-weight: bold;

padding-left: 40px;

margin-top: 20px;

}

.block-content img {

width: 60%;

display: block;

margin: auto;

margin-top: 40px;

margin-bottom: 40px;

}

hr {

width: 80%;

display: block;

margin: auto;

border-bottom: 1px solid #e5e5e5;

}

.block-content p {

padding: 15px;

line-height: 30px;

text-indent: 50px;

}

</style>

<div class="slider">

<div id="carouselExampleIndicators" class="carousel slide" data-bs-ride="true">

<div class="carousel-indicators">

<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>

<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="1" aria-label="Slide 2"></button>

</div>

<div class="carousel-inner">

<div class="carousel-item active">

<img src="/img/image2.png" class="d-block w-100" alt="...">

<div class="carousel-caption d-none d-md-block">

<h5>Цікаві місця світу</h5>

<p>На цьому сайті Ви зможете знайти інформацію про деякі цікаві місця світу.</p>

</div>

</div>

<div class="carousel-item">

<img src="/img/image3.png" class="d-block w-100" alt="...">

</div>

</div>

<button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="prev">

<span class="carousel-control-prev-icon" aria-hidden="true"></span>

<span class="visually-hidden">Previous</span>

</button>

<button class="carousel-control-next" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="next">

<span class="carousel-control-next-icon" aria-hidden="true"></span>

<span class="visually-hidden">Next</span>

</button>

</div>

</div>

**header.blade.php**

<!DOCTYPE html>

<html lang="{{ str\_replace('\_', '-', app()->getLocale()) }}">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta name="csrf-token" content="{{ csrf\_token() }}">

<title>Inter-world</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.4/jquery.min.js" integrity="sha512-pumBsjNRGGqkPzKHndZMaAG+bir374sORyzM3uulLV14lN5LyykqNk8eEeUlUkB3U0M4FApyaHraT65ihJhDpQ==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>

<link rel="stylesheet" type="text/css" href="/css/style.css">

</head>

<body>

<header class="header">

<div class="container">

<div class="row">

<div class="col-4">

<div class="header-logo"><a href="/"><img src="/img/logo.svg"></a> </div>

</div>

<div class="col-8">

<nav class="nav">

<ul class="menu">

@if (Route::has('login'))

@auth

<li class="item active">

<a href="/view-resources">Корисні ресурси <a href="/resources" style="margin-left:10px"><svg height="10px" version="1.1" viewBox="0 0 18 18" width="10px" xmlns="http://www.w3.org/2000/svg" xmlns:sketch="http://www.bohemiancoding.com/sketch/ns" xmlns:xlink="http://www.w3.org/1999/xlink"><title/><desc/><defs/><g fill="none" fill-rule="evenodd" id="Page-1" stroke="none" stroke-width="1"><g fill="#000000" id="Core" transform="translate(-213.000000, -129.000000)"><g id="create" transform="translate(213.000000, 129.000000)"><path d="M0,14.2 L0,18 L3.8,18 L14.8,6.9 L11,3.1 L0,14.2 L0,14.2 Z M17.7,4 C18.1,3.6 18.1,3 17.7,2.6 L15.4,0.3 C15,-0.1 14.4,-0.1 14,0.3 L12.2,2.1 L16,5.9 L17.7,4 L17.7,4 Z" id="Shape"/></g></g></g></svg></a></a>

</li>

<li class="item active">

<a href="/view-publications">Журнал публікацій <a href="/publications" style="margin-left:10px"><svg height="10px" version="1.1" viewBox="0 0 18 18" width="10px" xmlns="http://www.w3.org/2000/svg" xmlns:sketch="http://www.bohemiancoding.com/sketch/ns" xmlns:xlink="http://www.w3.org/1999/xlink"><title/><desc/><defs/><g fill="none" fill-rule="evenodd" id="Page-1" stroke="none" stroke-width="1"><g fill="#000000" id="Core" transform="translate(-213.000000, -129.000000)"><g id="create" transform="translate(213.000000, 129.000000)"><path d="M0,14.2 L0,18 L3.8,18 L14.8,6.9 L11,3.1 L0,14.2 L0,14.2 Z M17.7,4 C18.1,3.6 18.1,3 17.7,2.6 L15.4,0.3 C15,-0.1 14.4,-0.1 14,0.3 L12.2,2.1 L16,5.9 L17.7,4 L17.7,4 Z" id="Shape"/></g></g></g></svg></a></a>

</li>

<li class="item">

<a href="{{ route('logout') }}">Вийти</a>

</li>

@else

<li class="item active">

<a href="/view-publications">Журнал публікацій</a>

</li>

<li class="item active">

<a href="/view-resources">Корисні ресурси</a>

</li>

<li class="item">

<a href="{{ route('login') }}">Вхід</a>

</li>

@endauth

@endif

<!-- <li class="item active">

<a href="/category">Категорії</a>

</li>

<li class="item">

<a href="/clients">Клієнти</a>

</li>

<li class="item">

<a href="/statistics">Статистика</a>

</li> -->

</ul>

<div class="hamburger">

<span></span>

<span></span>

<span></span>

</div>

</nav>

</div>

</div>

</div>

</header>

<!-- <div class="relative flex items-top justify-center min-h-screen bg-gray-100 dark:bg-gray-900 sm:items-center py-4 sm:pt-0">

@if (Route::has('login'))

<div class="hidden fixed top-0 right-0 px-6 py-4 sm:block">

@auth

<a href="{{ url('/home') }}" class="text-sm text-gray-700 dark:text-gray-500 underline">Home</a>

@else

<a href="{{ route('login') }}" class="text-sm text-gray-700 dark:text-gray-500 underline">Log in</a>

@if (Route::has('register'))

<a href="{{ route('register') }}" class="ml-4 text-sm text-gray-700 dark:text-gray-500 underline">Register</a>

@endif

@endauth

</div>

@endif

</div> -->

<script type="text/javascript">

$(document).ready(function(){

$('.hamburger').on('click', function(){

document.querySelector(".header").classList.toggle("mobile");

document.querySelector("body").classList.toggle("hidden");

});

});

</script>

**footer.blade.php**

<footer>

<p style="text-align: center;">Inter World 2024.</p>

</footer>

</body>

</html>

**dashboard.blade.php**

@extends('layout')

@section('content')

<div class="container">

<div class="row justify-content-center">

<div class="col-md-8">

<div class="card">

<div class="card-header">{{ \_\_('Dashboard') }}</div>

<div class="card-body">

@if (session('success'))

<div class="alert alert-success" role="alert">

{{ session('success') }}

</div>

@endif

You are Logged In

</div>

</div>

</div>

</div>

</div>

@endsection

**layout.blade.php**

<!DOCTYPE html>

<html>

<head>

<style type="text/css">

@import url(https://fonts.googleapis.com/css?family=Raleway:300,400,600);

.navbar-laravel

{

box-shadow: 0 2px 4px rgba(0,0,0,.04);

}

.navbar-brand , .nav-link, .my-form, .login-form

{

font-family: Raleway, sans-serif;

}

.my-form

{

padding-top: 1.5rem;

padding-bottom: 1.5rem;

}

.my-form .row

{

margin-left: 0;

margin-right: 0;

}

.login-form

{

padding-top: 1.5rem;

padding-bottom: 1.5rem;

}

.login-form .row

{

margin-left: 0;

margin-right: 0;

}

</style>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-light navbar-laravel">

<div class="container">

<a class="navbar-brand" href="#">Inter World</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav ml-auto">

@guest

<li class="nav-item">

<a class="nav-link" href="{{ route('login') }}">Вхід</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ route('register') }}">Реєстрація</a>

</li>

@else

<li class="nav-item">

<a class="nav-link" href="{{ route('logout') }}">Logout</a>

</li>

@endguest

</ul>

</div>

</div>

</nav>

@yield('content')

</body>

</html>

**view-publications.blade.php**

@include('header')

<div class="container" style="position: relative;">

<img src="/img/globe.svg" class="globe">

<img src="/img/globe.svg" class="globe\_bottom">

<div class="main-container">

<h2 class="customh2">Корисні ресурси</h2>

<div class="block-search">

<form class="d-flex" role="search" method="GET">

<input class="form-control me-2" type="search" placeholder="Пошук" aria-label="Пошук" name="search">

<button class="btn btn-outline-success custom\_but" type="submit">Пошук</button>

</form>

</div>

<div class="block-filter">

<select class="form-select" onchange="sort(this.value)">

@if(isset($\_GET['sort']))

@if($\_GET['sort'] == 'date')

<option value="date" selected>Сортування по даті</option>

<option value="id">Сортування по ID</option>

@else($\_GET['sort'] == 'id')

<option value="date">Сортування по даті</option>

<option value="id" selected>Сортування по ID</option>

@endif

@else

<option value="date">Сортування по даті</option>

<option value="id">Сортування по ID</option>

@endif

</select>

</div>

<div class="my-catalog" id="my-catalog">

<div class="block-category">

@foreach($category as $cat)

<div class="card-cat"><a href="/view-publications?cat={{$cat->id}}">{{$cat->name}}</a></div>

@endforeach

</div>

<div class="journal">

<div class="row row-cols-1 row-cols-md-2 g-4">

@if($catalog == "[]")

<div class="four\_zero\_four\_bg">

<h1 class="text-center ">404</h1>

</div>

<h3 class="err404">По цьому запиті нічого не знайдено!</h3>

@else

@foreach($catalog as $journal)

<a href="/view-publication/<?=$journal->id?>">

<div class="col">

<div class="card">

@if(!empty($journal->image))

<img src="/path/{{ explode('|', $journal->image)[0]}}" class="card-img-top" alt="...">

@endif

<div class="card-body">

<h5 class="card-title">{{$journal->name}}</h5>

<div class="card-text">{!! substr($journal->text, 0, 500)!!}

<div class="q grad">&nbsp;</div>

</div>

<p class="smallDate">Дата публікації: {{$journal->created\_at}}</p>

</div>

</div>

</div>

</a>

@endforeach

@endif

</div>

</div>

</div>

</div>

</div>

<?php if(strpos($\_SERVER['REQUEST\_URI'], '?') !== false){ ?>

<script type="text/javascript">

$([document.documentElement, document.body]).animate({

scrollTop: $("#my-catalog").offset().top

}, 500);

</script>

<?php } ?>

<script type="text/javascript">

function sort($val) {

window.location = '?sort='+$val;

}

</script>

@include('footer')

**view-resources.blade.php**

@include('header')

<div class="container" style="position: relative;">

<img src="/img/globe.svg" class="globe">

<img src="/img/globe.svg" class="globe\_bottom">

<div class="main-container">

<h2 class="customh2">Журнал публікацій</h2>

<div class="block-search">

<form class="d-flex" role="search" method="GET">

<input class="form-control me-2" type="search" placeholder="Пошук" aria-label="Пошук" name="search">

<button class="btn btn-outline-success custom\_but" type="submit">Пошук</button>

</form>

</div>

<div class="block-filter">

<select class="form-select" onchange="sort(this.value)">

@if(isset($\_GET['sort']))

@if($\_GET['sort'] == 'date')

<option value="date" selected>Сортування по даті</option>

<option value="id">Сортування по ID</option>

@else($\_GET['sort'] == 'id')

<option value="date">Сортування по даті</option>

<option value="id" selected>Сортування по ID</option>

@endif

@else

<option value="date">Сортування по даті</option>

<option value="id">Сортування по ID</option>

@endif

</select>

</div>

<div class="my-catalog" id="my-catalog">

<div class="journal">

<div class="row row-cols-1 row-cols-md-2 g-4">

@if($catalog == "[]")

<div class="four\_zero\_four\_bg">

<h1 class="text-center ">404</h1>

</div>

<h3 class="err404">По цьому запиті нічого не знайдено!</h3>

@else

@foreach($catalog as $journal)

<a href="/view-resource/<?=$journal->id?>">

<div class="col">

<div class="card">

@if(!empty($journal->image))

<img src="/path/{{ explode('|', $journal->image)[0]}}" class="card-img-top" alt="...">

@endif

<div class="card-body">

<h5 class="card-title">{{$journal->name}}</h5>

<div class="card-text">{!! substr($journal->text, 0, 500)!!}

<div class="q grad">&nbsp;</div>

</div>

<p class="smallDate">Дата публікації: {{$journal->created\_at}}</p>

</div>

</div>

</div>

</a>

@endforeach

@endif

</div>

</div>

</div>

</div>

</div>

<?php if(strpos($\_SERVER['REQUEST\_URI'], '?') !== false){ ?>

<script type="text/javascript">

$([document.documentElement, document.body]).animate({

scrollTop: $("#my-catalog").offset().top

}, 500);

</script>

<?php } ?>

<script type="text/javascript">

function sort($val) {

window.location = '?sort='+$val;

}

</script>

@include('footer')

**category.blade.php**

@include('admin.header')

<div class="container main-container">

<style type="text/css">

.main-container{

background-color: white;

}

.my-category{

padding: 20px;

}

.main-title h2{

font-size: 30px;

font-weight: bold;

color: #0e5d7f;

width: fit-content;

border-bottom: 3px solid #0e5d7f;

}

table {

font-family: arial, sans-serif;

border-collapse: collapse;

width: 100%;

margin-top: 30px;

text-align: center;

}

td, th {

border: 1px solid #dddddd;

text-align: left;

padding: 8px;

}

tr:nth-child(even) {

background-color: #dddddd;

}

.icon-operation {

display: inline-flex;

}

.icon-operation img{

width: 30px;

display: block;

text-align: center;

}

#my-custom-butt{

background-color: #0d0f14;

color: white;

border: none;

padding: 5px;

padding-left: 30px;

padding-right: 30px;

margin-top: 40px;

margin-bottom: 40px;

font-size: 20px;

}

#my-custom-butt:hover{

background-color: #0e5d7f;

cursor: pointer;

}

</style>

<div class="my-category">

<div class="main-title"><h2>Категорії</h2></div>

<button id="my-custom-butt" class="custom\_but" style="float: right;" onclick="window.location='/admin/category/create'">Створити</button>

<p style="color:red; margin-top:15px; margin-bottom: 15px;">{{$errors}}</p>

@if($category->isEmpty())

<h2 style="color:red; margin-top:15px;">Немає створених категорій</h2>

@else

<table>

<tr>

<th>ID категорії</th>

<th>Назва категорії</th>

<th>Операції</th>

</tr>

@foreach($category as $cat)

<tr>

<td>{{$cat->id}}</td>

<td>{{$cat->name}}</td>

<td>

<div class="icon-operation">

<a href="/admin/category/edit/{{$cat->id}}"><img src="/img/edit.svg"></a>

<a href="/admin/category/delete/{{$cat->id}}" onclick="return confirm('Ви дійсно хочете видалити цю категорію?')"><img src="/img/delete.svg"></a>

</div>

</td>

</tr>

@endforeach

</table>

@endif

</div>

</div>

@include('footer')

**category-create.blade.php**

@include('admin.header')

<div class="container main-container">

<style type="text/css">

.main-container{

background-color: white;

}

.my-category{

padding: 20px;

}

#my-form{

margin-top: 30px;

}

.main-title h2{

font-size: 30px;

font-weight: bold;

color: #0e5d7f;

width: fit-content;

border-bottom: 3px solid #0e5d7f;

}

#my-custom-butt{

background-color: #0d0f14;

color: white;

border: none;

padding: 5px;

padding-left: 30px;

padding-right: 30px;

margin-top: 40px;

margin-bottom: 40px;

font-size: 20px;

}

#my-custom-butt:hover{

background-color: #0e5d7f;

cursor: pointer;

}

</style>

<div class="my-category">

<div class="main-title"><h2>Створити категорію</h2></div>

<form id="my-form" action="/admin/category/create/add" method="POST">

@csrf

<div class="form-group">

<label for="name">Назва категорії</label>

<input type="text" class="form-control" name="name" id="name" placeholder="Назва категорії" required>

</div>

<div class="form-group">

<label for="description">Опис</label>

<textarea class="form-control" name="description" id="description" placeholder="Опис категорії" rows="5" required></textarea>

<input type="hidden" name="user\_id" value="{{$user\_id}}">

</div>

<button type="submit" id="my-custom-butt" class="custom\_but">Створити</button>

</form>

</div>

</div>

</div>

@include('footer')

**edit-category.blade.php**

@include('admin.header')

<div class="container main-container">

<style type="text/css">

.main-container{

background-color: white;

}

.my-category-edit{

padding: 20px;

}

.main-title h2{

font-size: 30px;

font-weight: bold;

color: #0e5d7f;

width: fit-content;

border-bottom: 3px solid #0e5d7f;

}

.my-category-edit #my-form{

margin-top: 40px;

}

</style>

<div class="my-category-edit">

<div class="main-title"><h2>Редагування категорії</h2></div>

<form id="my-form" action="/admin/update-category/{{$id\_category}}" method="POST">

@csrf

@foreach($category as $cat)

<div class="form-group">

<label for="name">Назва</label>

<input type="text" class="form-control" id="name" name="name" placeholder="Назва категорії" value="{{$cat->name}}" required>

</div>

<div class="form-group">

<label for="description">Опис</label>

<textarea class="form-control" id="description" name="description" placeholder="Опис категорії" value="{{$cat->description}}" rows="5" required>{{$cat->description}}</textarea>

</div>

<input type="hidden" class="form-control" id="user\_id" name="user\_id" value="{{$cat->user\_id}}" required>

@endforeach

<button type="submit" class="btn btn-primary custom\_but">Оновити</button>

</form>

</div>

</div>

@include('footer')

**comments.blade.php**

@include('admin.header')

<div class="container main-container">

<style type="text/css">

.main-container{

background-color: white;

}

.my-brand{

padding: 20px;

}

.main-title h2{

font-size: 30px;

font-weight: bold;

color: #0e5d7f;

width: fit-content;

border-bottom: 3px solid #0e5d7f;

}

table {

font-family: arial, sans-serif;

border-collapse: collapse;

width: 100%;

margin-top: 30px;

text-align: center;

}

td, th {

border: 1px solid #dddddd;

text-align: left;

padding: 8px;

}

tr:nth-child(even) {

background-color: #dddddd;

}

.icon-operation {

display: inline-flex;

}

.icon-operation img{

width: 30px;

display: block;

text-align: center;

}

#my-custom-butt{

background-color: #0d0f14;

color: white;

border: none;

padding: 5px;

padding-left: 30px;

padding-right: 30px;

margin-top: 40px;

margin-bottom: 40px;

font-size: 20px;

}

#my-custom-butt:hover{

background-color: #0e5d7f;

cursor: pointer;

}

.checks{

display: block;

margin: auto;

}

</style>

<div class="my-brand">

<div class="main-title"><h2>Коментарі</h2></div>

<p style="color:red; margin-top:15px; margin-bottom: 15px;">{{$errors}}</p>

@if($comments->isEmpty())

<h2 style="color:red; margin-top:15px;">Немає коментарів</h2>

@else

<table>

<tr>

<th>Ім’я користувача</th>

<th>Коментар</th>

<th>Публікування</th>

<th>Операції</th>

</tr>

@foreach($comments as $comment)

<tr>

<td>{{$comment->name}}</td>

<td>{{$comment->text}}</td>

@if($comment->showCom == 1)

<td><input type="checkbox" name="public" class="checks" onchange="showCom(this, '{{$comment->id}}')" checked></td>

@else

<td><input type="checkbox" name="public" class="checks" onchange="showCom(this, '{{$comment->id}}')"></td>

@endif

<td>

<div class="icon-operation">

<a href="/admin/comments/delete/{{$comment->id}}" onclick="return confirm('Ви дійсно хочете видалити ей коментар?')"><img src="/img/delete.svg"></a>

</div>

</td>

</tr>

@endforeach

</table>

@endif

</div>

</div>

<script type="text/javascript">

function showCom($el, $id){

if($($el).is(":checked")) {

$.ajax({

url: '/admin/showCom/'+$id,

method: 'get',

dataType: 'html',

data: {},

success: function(data){

}

});

}else{

$.ajax({

url: '/admin/hideCom/'+$id,

method: 'get',

dataType: 'html',

data: {},

success: function(data){

}

});

}

}

</script>

@include('footer')

**comments-create.blade.php**

@include('admin.header')

<div class="container main-container">

<style type="text/css">

.main-container{

background-color: white;

}

.my-brand{

padding: 20px;

}

#my-form{

margin-top: 30px;

}

.main-title h2{

font-size: 30px;

font-weight: bold;

color: #0e5d7f;

width: fit-content;

border-bottom: 3px solid #0e5d7f;

}

#my-custom-butt{

background-color: #0d0f14;

color: white;

border: none;

padding: 5px;

padding-left: 30px;

padding-right: 30px;

margin-top: 40px;

margin-bottom: 40px;

font-size: 20px;

}

#my-custom-butt:hover{

background-color: #0e5d7f;

cursor: pointer;

}

</style>

<div class="my-brand">

<div class="main-title"><h2>Створити марку авто</h2></div>

<form id="my-form" action="/admin/brand/create/add" method="POST">

@csrf

<div class="form-group">

<label for="name">Назва марки авто</label>

<input type="text" class="form-control" name="name" id="name" placeholder="Назва марки" required>

</div>

<button type="submit" id="my-custom-butt" class="custom\_but">Створити</button>

</form>

</div>

</div>

</div>

@include('footer')

**content.blade.php**

@include('admin.header')

<div class="container main-container">

<style type="text/css">

.main-container{

background-color: white;

}

.my-detail{

padding: 20px;

padding-bottom: 50px;

}

.main-title h2{

font-size: 30px;

font-weight: bold;

color: #0e5d7f;

width: fit-content;

border-bottom: 3px solid #0e5d7f;

}

table {

font-family: arial, sans-serif;

border-collapse: collapse;

width: 100%;

margin-top: 30px;

text-align: center;

}

td, th {

border: 1px solid #dddddd;

text-align: left;

padding: 8px;

}

tr:nth-child(even) {

background-color: #dddddd;

}

.icon-operation {

display: inline-flex;

}

.icon-operation img{

width: 30px;

display: block;

text-align: center;

}

#my-custom-butt{

background-color: #0d0f14;

color: white;

border: none;

padding: 5px;

padding-left: 30px;

padding-right: 30px;

margin-top: 40px;

margin-bottom: 40px;

font-size: 20px;

}

#my-custom-butt:hover{

background-color: #0e5d7f;

cursor: pointer;

}

.image\_td{

width: 50px;display: block;

margin: auto;

}

</style>

<div class="my-detail">

<div class="main-title"><h2>Контент</h2></div>

<button id="my-custom-butt" class="custom\_but" style="float: right;" onclick="window.location='/admin/content/create'">Створити публікацію</button>

@if($content->isEmpty())

<h2 style="color:red; margin-top:15px;">У Вас немає створених розділів</h2>

@else

<table>

<tr>

<th>Нумеріція</th>

<th>Заголовок</th>

<th>Тект</th>

<th>Операції</th>

</tr>

@foreach($content as $cont)

<tr>

<td>{{$cont->nums}}</td>

<td>{{$cont->title}}</td>

<td>{{$cont->text}}</td>

<td>

<div class="icon-operation">

<!-- <a href="/search/content/{{$cont->id}}"><img src="/img/show.svg"></a> -->

<a href="/admin/content/edit/{{$cont->id}}"><img src="/img/edit.svg"></a>

<a href="/admin/content/delete/{{$cont->id}}" onclick="return confirm('Ви дійсно хочете видалити цей розділ?')"><img src="/img/delete.svg"></a>

</div>

</td>

</tr>

@endforeach

</table>

@endif

</div>

</div>

@include('footer')

**content-create.blade.php**

@include('admin.header')

<div class="container main-container">

<style type="text/css">

.form-input input {

display:none;

}

.form-input label {

padding: 10px;

text-align:center;

background:#1F244B;

color:white;

font-size:15px;

text-transform:Uppercase;

font-weight:600;

border-radius:3px;

cursor:pointer;

}

.form-input img {

height: 200px;

object-fit: cover;

margin-bottom: 20px;

}

</style>

<div class="my-detail">

<div class="main-title"><h2>Створити розділ</h2></div>

<form id="my-form" action="/admin/content/create/add" method="POST">

@csrf

<div class="form-group">

<label for="nums">Нумерація</label>

<input type="text" class="form-control" name="nums" id="nums" placeholder="Нумерація" required>

</div>

<div class="form-group">

<label for="title">Заголовок</label>

<input type="text" class="form-control" name="title" id="title" placeholder="Заголовок" required>

</div>

<div class="form-group">

<label for="text">Текст</label>

<textarea class="form-control" name="text" id="text" placeholder="Текст " rows="5" required></textarea>

</div>

<button type="submit" id="my-custom-butt" class="custom\_but">Створити</button>

</form>

</div>

</div>

</div>

@include('footer')

**edit-content.blade.php**

@include('admin.header')

<div class="container main-container">

<style type="text/css">

.main-container{

background-color: white;

}

.my-detail-edit{

padding: 20px;

}

.main-title h2{

font-size: 30px;

font-weight: bold;

color: #0e5d7f;

width: fit-content;

border-bottom: 3px solid #0e5d7f;

}

.my-detail-edit #my-form{

margin-top: 40px;

}

</style>

<div class="my-detail-edit">

<div class="main-title"><h2>Редагування розділу</h2></div>

<form id="my-form" action="/admin/update-content/{{$id\_content}}" method="POST">

@csrf

@foreach($content as $cont)

<div class="form-group">

<label for="nums">Нумерація</label>

<input type="text" class="form-control" id="nums" name="nums" placeholder="Нумерація" value="{{$cont->nums}}" required>

</div>

<div class="form-group">

<label for="title">Заголовок</label>

<input type="text" class="form-control" id="title" name="title" placeholder="Заголовок" value="{{$cont->title}}" required>

</div>

<div class="form-group">

<label for="text">Текст</label>

<textarea class="form-control" id="text" name="text" placeholder="Текст" value="{{$cont->text}}" rows="5" required>{!!$cont->text!!}</textarea>

</div>

@endforeach

<button type="submit" class="btn btn-primary custom\_but ">Оновити</button>

</form>

</div>

</div>

@include('footer')

**edit-publications.blade.php**

@include('admin.header')

<div class="container main-container">

<style type="text/css">

.main-container{

background-color: white;

}

.my-detail-edit{

padding: 20px;

}

.main-title h2{

font-size: 30px;

font-weight: bold;

color: #0e5d7f;

width: fit-content;

border-bottom: 3px solid #0e5d7f;

}

.my-detail-edit #my-form{

margin-top: 40px;

}

</style>

<div class="my-detail-edit">

<div class="main-title"><h2>Журнал публікацій (редагування публікації)</h2></div>

<form id="my-form" action="/admin/update-publication/{{$id\_publication}}" method="POST" enctype="multipart/form-data">

@csrf

@foreach($publications as $publication)

<div class="form-group">

<label for="name">Категорія</label>

<select name="category\_id" required class="form-control">

@foreach($category as $cat)

@if($cat->id == $publication->category\_id)

<option value="{{$cat->id}}" selected>{{$cat->name}}</option>

@else

<option value="{{$cat->id}}">{{$cat->name}}</option>

@endif

@endforeach

</select>

</div>

<div class="form-group">

<label for="name">Назва</label>

<input type="text" class="form-control" id="name" name="name" placeholder="Name" value="{{$publication->name}}" required>

</div>

<div class="form-group">

<label for="text">Опис</label>

<textarea class="form-control" id="text" name="text" placeholder="Description" value="{{$publication->text}}" rows="5" required>{{$publication->text}}</textarea>

</div>

<div class="form-group">

<label for="image">Фото</label>

<br>

<input type="file" name="images[]" accept="image/\*" multiple>

<br><br>

<input type="text" class="form-control" id="image" name="image" placeholder="Image" value="{{$publication->image}}" style="pointer-events:none;">

</div>

@endforeach

<button type="submit" class="btn btn-primary">Оновити</button>

</form>

</div>

</div>

@include('footer')

**publications.blade.php**

@include('admin.header')

<div class="container main-container">

<style type="text/css">

.main-container{

background-color: white;

}

.my-detail{

padding: 20px;

padding-bottom: 50px;

}

.main-title h2{

font-size: 30px;

font-weight: bold;

color: #0e5d7f;

width: fit-content;

border-bottom: 3px solid #0e5d7f;

}

table {

font-family: arial, sans-serif;

border-collapse: collapse;

width: 100%;

margin-top: 30px;

text-align: center;

}

td, th {

border: 1px solid #dddddd;

text-align: left;

padding: 8px;

}

tr:nth-child(even) {

background-color: #dddddd;

}

.icon-operation {

display: inline-flex;

}

.icon-operation img{

width: 30px;

display: block;

text-align: center;

}

#my-custom-butt{

background-color: #0d0f14;

color: white;

border: none;

padding: 5px;

padding-left: 30px;

padding-right: 30px;

margin-top: 40px;

margin-bottom: 40px;

font-size: 20px;

}

#my-custom-butt:hover{

background-color: #0e5d7f;

cursor: pointer;

}

.image\_td{

width: 50px;display: block;

margin: auto;

}

.checks{

display: block;

margin: auto;

}

</style>

<div class="my-detail">

<div class="main-title"><h2>Журнал публікацій</h2></div>

<button id="my-custom-butt" style="float: right;" onclick="window.location='/admin/publications/create'">Створити</button>

@if($publications->isEmpty())

<h2 style="color:red; margin-top:15px;">У Вас немає створених публікацій</h2>

@else

<table>

<tr>

<th>Назва</th>

<th>Категорія</th>

<th>Текст</th>

<th>Публікувати</th>

<th>Операції</th>

</tr>

@foreach($publications as $publication)

<tr>

<td>{{$publication->name}}</td>

<td>

@foreach($category as $cat)

@if($cat->id == $publication->category\_id)

{{$cat->name}}

@endif

@endforeach

</td>

<td>{{$publication->text}}</td>

@if($publication->showPub == 1)

<td><input type="checkbox" name="public" class="checks" onchange="showPub(this, '{{$publication->id}}')" checked></td>

@else

<td><input type="checkbox" name="public" class="checks" onchange="showPub(this, '{{$publication->id}}')"></td>

@endif

<!-- <td><img class="image\_td" src="/path/{{$publication->image}}"></td> -->

<td>

<div class="icon-operation">

<a href="/view-publication/{{$publication->id}}" target="\_blank"><img src="/img/show.svg"></a>

<a href="/admin/publication/edit/{{$publication->id}}"><img src="/img/edit.svg"></a>

<a href="/admin/publication/delete/{{$publication->id}}" onclick="return confirm('Ви дійсно хочете видалити цю публікацію?')"><img src="/img/delete.svg"></a>

</div>

</td>

</tr>

@endforeach

</table>

@endif

</div>

</div>

<script type="text/javascript">

function showPub($el, $id){

if($($el).is(":checked")) {

$.ajax({

url: '/admin/showPub/'+$id,

method: 'get',

dataType: 'html',

data: {},

success: function(data){

}

});

}else{

$.ajax({

url: '/admin/hidePub/'+$id,

method: 'get',

dataType: 'html',

data: {},

success: function(data){

}

});

}

}

</script>

@include('footer')

**publications-create.blade.php**

@include('admin.header')

<div class="container main-container">

<style type="text/css">

.form-input input {

display:none;

}

.form-input label {

padding: 10px;

text-align:center;

background:#1F244B;

color:white;

font-size:15px;

text-transform:Uppercase;

font-weight:600;

border-radius:3px;

cursor:pointer;

}

.form-input img {

height: 200px;

object-fit: cover;

margin-bottom: 20px;

}

</style>

<div class="my-detail">

<div class="main-title"><h2>Журнал публікацій (Створення публікації)</h2></div>

<form id="my-form" action="/admin/publications/create/add" method="POST" enctype="multipart/form-data">

@csrf

<div class="form-group">

<label for="name">Категорія</label>

<select name="category\_id" required class="form-control">

@foreach($category as $cat)

<option value="{{$cat->id}}">{{$cat->name}}</option>

@endforeach

</select>

</div>

<div class="form-group">

<label for="name">Назва</label>

<input type="text" class="form-control" name="name" id="name" placeholder="Назва Публікації" required>

</div>

<div class="form-group">

<label for="text">Опис</label>

<textarea class="form-control" name="text" id="text" placeholder="Опис Публікації" rows="5" required></textarea>

</div>

<div class="form-group">

<label for="image">Фото</label>

<br>

<input type="file" name="images[]" accept="image/\*" multiple required>

<br>

<input type="text" class="form-control" name="images" id="image" placeholder="Image url" value="" style="pointer-events:none; display:none;" >

</div>

<input type="hidden" name="user\_id" value="{{$user\_id}}">

<button type="submit" id="my-custom-butt">Створити</button>

</form>

</div>

</div>

</div>

@include('footer')

**edit-publications.blade.php**

@include('admin.header')

<div class="container main-container">

<style type="text/css">

.main-container{

background-color: white;

}

.my-detail-edit{

padding: 20px;

}

.main-title h2{

font-size: 30px;

font-weight: bold;

color: #0e5d7f;

width: fit-content;

border-bottom: 3px solid #0e5d7f;

}

.my-detail-edit #my-form{

margin-top: 40px;

}

</style>

<div class="my-detail-edit">

<div class="main-title"><h2>Корисні ресурси (редагування публікації)</h2></div>

<form id="my-form" action="/admin/update-resources/{{$id\_publication}}" method="POST" enctype="multipart/form-data">

@csrf

@foreach($publications as $publication)

<div class="form-group" style="display: none;">

<label for="name">Категорія</label>

<select name="category\_id" required class="form-control">

@foreach($category as $cat)

@if($cat->id == $publication->category\_id)

<option value="{{$cat->id}}" selected>{{$cat->name}}</option>

@else

<option value="{{$cat->id}}">{{$cat->name}}</option>

@endif

@endforeach

</select>

</div>

<div class="form-group">

<label for="name">Назва</label>

<input type="text" class="form-control" id="name" name="name" placeholder="Name" value="{{$publication->name}}" required>

</div>

<div class="form-group">

<label for="text">Опис</label>

<textarea class="form-control" id="text" name="text" placeholder="Description" value="{{$publication->text}}" rows="5" required>{{$publication->text}}</textarea>

</div>

<div class="form-group">

<label for="image">Фото</label>

<br>

<input type="file" name="images[]" accept="image/\*" multiple>

<br><br>

<input type="text" class="form-control" id="image" name="image" placeholder="Image" value="{{$publication->image}}" style="pointer-events:none;">

</div>

@endforeach

<button type="submit" class="btn btn-primary">Оновити</button>

</form>

</div>

</div>

@include('footer')

**publications.blade.php**

@include('admin.header')

<div class="container main-container">

<style type="text/css">

.main-container{

background-color: white;

}

.my-detail{

padding: 20px;

padding-bottom: 50px;

}

.main-title h2{

font-size: 30px;

font-weight: bold;

color: #0e5d7f;

width: fit-content;

border-bottom: 3px solid #0e5d7f;

}

table {

font-family: arial, sans-serif;

border-collapse: collapse;

width: 100%;

margin-top: 30px;

text-align: center;

}

td, th {

border: 1px solid #dddddd;

text-align: left;

padding: 8px;

}

tr:nth-child(even) {

background-color: #dddddd;

}

.icon-operation {

display: inline-flex;

}

.icon-operation img{

width: 30px;

display: block;

text-align: center;

}

#my-custom-butt{

background-color: #0d0f14;

color: white;

border: none;

padding: 5px;

padding-left: 30px;

padding-right: 30px;

margin-top: 40px;

margin-bottom: 40px;

font-size: 20px;

}

#my-custom-butt:hover{

background-color: #0e5d7f;

cursor: pointer;

}

.image\_td{

width: 50px;display: block;

margin: auto;

}

.checks{

display: block;

margin: auto;

}

</style>

<div class="my-detail">

<div class="main-title"><h2>Корисні ресурси</h2></div>

<button id="my-custom-butt" style="float: right;" onclick="window.location='/admin/resources/create'">Створити</button>

@if($publications->isEmpty())

<h2 style="color:red; margin-top:15px;">У Вас немає створених публікацій</h2>

@else

<table>

<tr>

<th>Назва</th>

<th>Текст</th>

<th>Публікувати</th>

<th>Операції</th>

</tr>

@foreach($publications as $publication)

<tr>

<td>{{$publication->name}}</td>

<td>{{$publication->text}}</td>

@if($publication->showPub == 1)

<td><input type="checkbox" name="public" class="checks" onchange="showPub(this, '{{$publication->id}}')" checked></td>

@else

<td><input type="checkbox" name="public" class="checks" onchange="showPub(this, '{{$publication->id}}')"></td>

@endif

<!-- <td><img class="image\_td" src="/path/{{$publication->image}}"></td> -->

<td>

<div class="icon-operation">

<a href="/view-resource/{{$publication->id}}" target="\_blank"><img src="/img/show.svg"></a>

<a href="/admin/resources/edit/{{$publication->id}}"><img src="/img/edit.svg"></a>

<a href="/admin/resources/delete/{{$publication->id}}" onclick="return confirm('Ви дійсно хочете видалити цю публікацію?')"><img src="/img/delete.svg"></a>

</div>

</td>

</tr>

@endforeach

</table>

@endif

</div>

</div>

<script type="text/javascript">

function showPub($el, $id){

if($($el).is(":checked")) {

$.ajax({

url: '/admin/showPubJournal/'+$id,

method: 'get',

dataType: 'html',

data: {},

success: function(data){

}

});

}else{

$.ajax({

url: '/admin/hidePubJournal/'+$id,

method: 'get',

dataType: 'html',

data: {},

success: function(data){

}

});

}

}

</script>

@include('footer')

**publications-create.blade.php**

@include('admin.header')

<div class="container main-container">

<style type="text/css">

.form-input input {

display:none;

}

.form-input label {

padding: 10px;

text-align:center;

background:#1F244B;

color:white;

font-size:15px;

text-transform:Uppercase;

font-weight:600;

border-radius:3px;

cursor:pointer;

}

.form-input img {

height: 200px;

object-fit: cover;

margin-bottom: 20px;

}

</style>

<div class="my-detail">

<div class="main-title"><h2>Корисні ресурси (створити публікацію)</h2></div>

<form id="my-form" action="/admin/resources/create/add" method="POST" enctype="multipart/form-data">

@csrf

<div class="form-group" style="display: none;">

<label for="name">Категорія</label>

<select name="category\_id" required class="form-control">

@foreach($category as $cat)

<option value="{{$cat->id}}">{{$cat->name}}</option>

@endforeach

</select>

</div>

<div class="form-group">

<label for="name">Назва</label>

<input type="text" class="form-control" name="name" id="name" placeholder="Назва Публікації" required>

</div>

<div class="form-group">

<label for="text">Опис</label>

<textarea class="form-control" name="text" id="text" placeholder="Опис Публікації" rows="5" required></textarea>

</div>

<div class="form-group">

<label for="image">Фото</label>

<br>

<input type="file" name="images[]" accept="image/\*" multiple required>

<br>

<input type="text" class="form-control" name="images" id="image" placeholder="Image url" value="" style="pointer-events:none; display:none;" >

</div>

<input type="hidden" name="user\_id" value="{{$user\_id}}">

<button type="submit" id="my-custom-butt">Створити</button>

</form>

</div>

</div>

</div>

@include('footer')

**create-user.blade.php**

@include('admin.header')

<div class="container main-container">

<div class="my-detail">

<div class="main-title"><h2>Створити користувача</h2></div>

<form id="my-form" action="/admin/user/add" method="POST" enctype="multipart/form-data">

@csrf

<div class="form-group">

<label for="name">Імя</label>

<input type="text" class="form-control" id="name" name="name" placeholder="Name" required>

</div>

<div class="form-group">

<label for="email">Email</label>

<input type="text" class="form-control" id="email" name="email" placeholder="Email"required>

</div>

<div class="form-group">

<label for="password">Пароль</label>

<input type="text" class="form-control" id="password" name="password" placeholder="\*\*\*\*\*\*\*\*" min="6" value="" required>

</div>

<button type="submit" id="my-custom-butt">Створити</button>

</form>

</div>

</div>

</div>

@include('footer')

**edit-user.blade.php**

@include('admin.header')

<div class="container main-container">

<style type="text/css">

.main-container{

background-color: white;

}

.my-detail-edit{

padding: 20px;

}

.main-title h2{

font-size: 30px;

font-weight: bold;

color: #0e5d7f;

width: fit-content;

border-bottom: 3px solid #0e5d7f;

}

.my-detail-edit #my-form{

margin-top: 40px;

}

</style>

<div class="my-detail-edit">

<div class="main-title"><h2>Редагування даних користувача</h2></div>

<form id="my-form" action="/admin/update-user/{{$id\_user}}" method="POST">

@csrf

@foreach($users as $user)

<div class="form-group">

<label for="name">Імя</label>

<input type="text" class="form-control" id="name" name="name" placeholder="Name" value="{{$user->name}}" required>

</div>

<div class="form-group">

<label for="email">Email</label>

<input type="text" class="form-control" id="email" name="email" placeholder="Email" value="{{$user->email}}" required style="pointer-events:none;">

</div>

<div class="form-group">

<label for="password">Пароль</label>

<input type="text" class="form-control" id="password" name="password" placeholder="\*\*\*\*\*\*\*\*" min="6" value="">

</div>

@endforeach

<button type="submit" class="btn btn-primary">Оновити</button>

</form>

</div>

</div>

@include('footer')