

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Навчально-науковий інститут атомної та теплової енергетики
Кафедра інженерії програмного забезпечення в енергетиці

ДО ЗАХИСТУ ДОПУЩЕНО
В.о. завідувача кафедри
_____Олександр КОВАЛЬ
« _____ » _____ 2023р.

Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інженерія програмного забезпечення
інтелектуальних кібер-фізичних систем і веб-технологій»
спеціальності 121 Інженерія програмного забезпечення
на тему: Інформаційна система «Щотижневий звіт викладача»

Виконав:
студент IV курсу, групи ТВ-391
Шлик Олександр Олегович

Керівник:
Доцент, к.ф.-м.н., доц. Свинчук О.В.

Рецензент:
Професор кафедри ЦТЕ,
д.т.н., проф. Отрох С.І.

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ — 2023 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Навчально-науковий інститут атомної та теплової енергетики

Кафедра інженерії програмного забезпечення в енергетиці

Рівень вищої освіти перший (бакалаврський)

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем і веб-технологій»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр КОВАЛЬ

(підпис)

«___» _____ 2023 р.

ЗАВДАННЯ

на дипломну роботу студенту

Шлику Олександр Олександровичу

1. Тема роботи Інформаційна система «Щотижневий звіт викладача»

керівник роботи Свинчук О.В., к.ф.-м.н., доцент

(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом університету від «__» _____ 2023 р. № _____

2. Строк подання студентом роботи: _____

3. Вихідні дані до роботи: веб-сайт, HTML+CSS+JS, мова програмування PHP, СУБД MySQL, середовище розробки Sublime Text

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) провести аналіз предметної області, проаналізувати новітні технології розробки вебсайтів, проаналізувати особливості розвитку сучасних вебсайтів з точки зору проектування дизайну вебсайту, вивчити сучасні засоби розробки вебсайтів, реалізувати необхідний функціонал, розробити засоби адміністрування програмного продукту, провести тестування та перевірити працездатність розробленого вебсайту.

5. Перелік ілюстративного матеріалу: актуальність, мета і задача роботи, аналіз існуючих систем, засоби розробки системи, структура системи, концептуальна модель, логічна модель, інтерфейс адміністратора системи, інтерфейс користувача системи, приклад звіту, висновки

6. Дата видачі завдання «__» _____ 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Строки виконання етапів роботи	Примітка
1	Отримання завдання		Виконано
2	Дослідження предметної області	17.04.2023-18.04.2023	Виконано
3	Постановка вимог до проєктування системи	19.04.2023-20.04.2023	Виконано
4	Дослідження існуючих рішень	21.04.2023-23.04.2023	Виконано
5	Розробка програмного продукту	24.04.2023-14.05.2023	Виконано
6	Тестування	15.05.2023-21.05.2023	Виконано
7	Захист програмного продукту	15.05.2023-19.05.2023	Виконано
8	Оформлення дипломної роботи	22.05.2023-12.06.2023	Виконано
9	Передзахист	05.06.2023-11.06.2023	Виконано
10	Захист	19.06.2023-25.06.2023	Виконано

Студент

(підпис)

Олександр ШЛИК
(ім'я, прізвище)

Керівник роботи

(підпис)

Ольга СВИНЧУК
(ім'я, прізвище)

РЕФЕРАТ

Структура та опис дипломної роботи. Робота містить 72 сторінки, 29 рисунків, 1 додаток та 23 посилання.

Метою роботи є розробка інформаційної системи для подання щотижневих звітів для викладачів.

Для досягнення поставленої мети були виконані такі завдання:

- описати процес подання звітів;
- провести аналіз наявних застосунків у предметній області;
- описати інструментальні засоби та методи розробки веб сайтів;
- створити власний програмний продукт для подання щотижневих звітів

та провести його тестування.

Програмний застосунок у вигляді вебсайту був розроблений для подання щотижневих звітів викладача, який може бути використаний в діяльності наукових, навчальних установ, підприємств усіх форм власності з метою поліпшення робіт викладачів.

Практичне значення отриманих результатів полягає в тому, що розробці вебсайту для автоматизації процесу звітної діяльності викладачів, який може бути використаний в діяльності наукових, навчальних установ, підприємств усіх форм власності з метою полегшення роботи науково-педагогічних працівників і відповідальних за звіти.

Ключові слова: звіт, вебсайт, MySQL, засоби розробки сайтів, методи адміністрування.

ABSTRACT

Structure and description of the thesis. The work contains 72 pages, 29 figures, 1 appendix and 23 references.

The purpose of the work is to develop an information system for submitting weekly reports for teachers.

To achieve the goal, the following tasks were performed:

- describe the reporting process;
- analyze the available types of applications in the subject area;
- describe tools and methods of web site development;
- create your own software product for submitting weekly test reports.

The software application in the form of a website was developed for submitting weekly teacher reports, which can be used in the activities of scientific, educational institutions, enterprises of all forms of ownership in order to improve the work of teachers.

The practical significance of the obtained results lies in the development of a website for automating the creation of weekly reports by teachers, which can be used in the activities of scientific, educational institutions, enterprises of all forms of ownership in order to facilitate the work of scientific and pedagogical workers and those responsible for reports.

Keywords: report, website, MySQL, site development tools, administration methods.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
1 ПОСТАНОВКА ЗАДАЧІ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ «ЩОТИЖНЕВИЙ ЗВІТ ВИКЛАДАЧА».....	9
1.1 Користувачі системи.....	9
1.2 Функціонал системи.....	10
1.3 Інструменти розробки системи.....	11
Висновки до розділу 1.....	12
2 АНАЛІЗ ІСНУЮЧИХ НА РИНКУ ІНФОРМАЦІЙНИХ СИСТЕМ ЩОТИЖНЕВИХ ЗВІТІВ.....	13
2.1 Система електронного документообігу Directum.....	13
2.2 Система DocsVision.....	16
2.3 «1С: Документообіг 8».....	17
Висновки до розділу 2.....	18
3 ЗАСОБИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ «ЩОТИЖНЕВИЙ ЗВІТ ВИКЛАДАЧА».....	19
3.1 Вибір інструментальних засобів розробки.....	19
3.2 Опис середовища та системи.....	21
3.3 Основні вимоги до запуску та роботи з вебсайтом.....	25
3.3.1 Вимоги до інтерфейсів інформаційної системи.....	25
3.3.2 Вимоги до технічного забезпечення.....	26
3.3.3 Вимоги до програмного забезпечення.....	26
Висновки до розділу 3.....	26
4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ	

«ЩОТИЖНЕВИЙ ЗВІТ ВИКЛАДАЧА».....	27
4.1 Структура вебсайту.....	27
4.2 Проектування та розробка БД.....	28
4.3 Концептуальна та логічна моделі бази даних.....	32
4.4 Аналіз можливих помилок.....	34
4.5 Опис основних форм інтерфейсу веб-сервісу.....	38
Висновки до розділу 4.....	38
5 РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ.....	40
5.1 Опис функціоналу.....	40
5.2 Тестування розробленої інформаційної системи.....	44
Висновки до розділу 5.....	50
ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53
ДОДАТОК А.....	55

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних.

ІС – інформаційна система.

ОЗУ – оперативний запам'ятовуючий пристрій.

ОС – операційна система.

ПЗ – програмне забезпечення.

ПК – персональний комп'ютер.

СУБД – система управління базами даних.

BIOS – basic input/output system.

DOS – disk operation system.

SQL – structured query language.

ВСТУП

У закладах вищої освіти викладачі займаються різними видами науково-педагогічної діяльності, а саме: навчальна, методична, організаційна, виховна, наукова, громадсько-педагогічна, тощо. Для того, щоб зафіксувати результати своєї роботи, кожного року викладачі заповнюють різноманітні звіти, за якими проводиться самоаналіз роботи кафедри/інституту/університету. Раніше такі звіти заповнювалися в паперовому вигляді. Проте більшість закладів вищої освіти сьогодні перейшли на дистанційну форму навчання, що унеможливило це зробити тепер в повному обсязі.

З іншої сторони завідувачам кафедри потрібно контролювати роботу своїх підлеглих. Кожен викладач в кінці тижня повинен надіслати звіт, у якому потрібно описати детально, що він зробив по кожному виду діяльності протягом тижня. Найпростішим варіантом такого звіту може бути таблиця в Excel, у якій для кожного виду діяльності буде відведена своя комірка. Викладач заповнює її й надсилає відповідальному за звіти на кафедрі. Потім відповідальний повинен зробити зведену таблицю для всієї кафедри і надіслати завідувачу кафедри на перевірку. Ця рутинна робота займає немало часу у всіх її виконавців.

Тому автоматизація таких процесів на кафедрі буде чудовим варіантом для полегшення роботи викладачів і відповідальних за звіти.

Метою створення програмного продукту є розробка інформаційної системи «Щотижневий звіт викладача», яка допоможе викладачам вести облік своїх результатів роботи, а завідувачу кафедри контролювати роботу науково-педагогічних працівників.

Робота складається з п'яти розділів, у яких послідовно описано мету розробки даної інформаційної системи, існуючі системи для створення різноманітних звітів, засоби розробки, архітектура системи та робота користувача із самою системою.

1 ПОСТАНОВКА ЗАДАЧІ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ «ЩОТИЖНЕВИЙ ЗВІТ ВИКЛАДАЧА»

Викладачі закладів вищої освіти займаються різними видами науково-педагогічної діяльності, а саме: навчальна, методична, організаційна, виховна, наукова, громадсько-педагогічна, тощо. Кожен викладач в кінці тижня повинен надіслати звіт, у якому потрібно описати детально, що він зробив по кожному виду діяльності протягом визначеного періоду.

Метою кваліфікаційної роботи є розробка зручного та простого вебсайту для подання щотижневих звітів науково-педагогічними працівниками. Застосування інформаційної системи дозволить зменшити витрати часу на роботу зі звітами, автоматизувавши цей процес та інших видів робіт за рахунок зниження трудомісткості роботи викладача

Отже, основне завдання системи – автоматизувати процес звітної діяльності викладачів. Така система, з високим рівнем впровадження, повинна полегшити повсякденну роботу науково-педагогічних працівників закладів вищої освіти.

1.1 Користувачі системи

Інформаційна система повинна передбачати поділ прав доступу і бути реалізована для двох категорій користувачів:

- адміністратор;
- викладачі кафедри.

Система для адміністратора сайту повинна надавати можливість:

- редагувати інформацію по користувачам;
- редагувати доступ до системи;
- редагувати документи;

- реєструватися у системі.

Система для викладачів повинна надавати можливість:

- додавати документи;
- редагувати документи;
- реєструватися у системі;
- переглядати інформацію;
- спілкуватися в чаті.

1.2 Функціонал системи

Функціонал системи, призначений для викладача, у свою чергу повинен забезпечувати наступні функції:

- аутентифікація користувача в системі;
- перегляд списку звітів;
- додавання звіту;
- перегляд та редагування власного профілю;
- перегляд історії роботи зі звітами, зроблених користувачем;
- видалення звіту;
- зміна пароля користувачем;
- додавання коментарів.

Система для адміністрації сайту повинна забезпечувати наступні можливості:

- блокування доступу до системи для інших користувачів;
- можливість додавати, видаляти, змінювати інформацію про користувачів;
- можливість додавати, видаляти, змінювати інформацію про звіт.

При введенні та редагуванні даних система повинна контролювати правильність введеної інформації та по можливості виключати ситуації, які можуть призвести до помилкових дій з боку користувачів.

1.3 Інструменти розробки системи

З метою полегшення підтримки та обслуговування системи, а також її подальшого розширення, вона повинна бути розроблена та створена на основі трирівневої архітектури (клієнт, сервер додатків і база даних) для побудови програмних систем та мати клієнтську і серверну частини. Серверна частина повинна відповідати за доступ до даних і містити бізнес-логіку програми. Клієнтська частина, у свою чергу, повинна реалізувати користувацький інтерфейс програми.

Необхідно використовувати MySQL [1] як рівень доступу до даних, який забезпечує централізоване структуроване зберігання всіх системних даних, гарантуючи їх цілісність і узгодженість, а також надає безліч низькорівневих сервісів для: читання даних зі сховища, збереження даних, зміни їх структура і т.д. На цьому рівні має бути створена база даних, яка буде зберігати всі дані системи. Реалізація команд отримання даних, контроль цілісності та узгодженості даних повинні здійснюватися за допомогою відповідних збережених процедур, тригерів та інших об'єктів, наданих сервером.

Рівень бізнес-логіки буде розгорнутий на сервері додатків і представлятиме ядро системи. Більша частина бізнес-логіки системи повинна бути зосереджена на цьому рівні:

- алгоритми авторизації користувачів системи, перевірки прав доступу;
- правила обробки даних, такі як: перевірка правильності заповнення даних користувачем, перевірка та організація взаємозв'язків даних;
- клас для підключення до бази даних;
- класи та алгоритми роботи з таблицями бази даних та запуску виконання відповідних збережених процедур і функцій на сервері.

Система буде розроблена на платформі в інтегрованому середовищі розробки Sublime Text.

Частина інтерфейсу має бути реалізована з використанням мови гіпертекстової розмітки HTML версії 5, а також технології каскадної таблиці стилів CSS FlexBox та мови сценаріїв браузера JavaScript.

Висновки до розділу 1

У першому розділі описана предметна область, мета даної роботи, основні завдання. Описані категорії користувачів та їхній поділ прав доступу і як саме вони реалізовані.

Описаний функціонал системи, призначений для різних користувачів (адміністратора та викладачів).

Описані інструменти розробки системи. З метою полегшення підтримки та обслуговування системи, а також її подальшого розширення, вона повинна бути розроблена та створена на основі трирівневої архітектури (клієнт, сервер додатків і база даних) для побудови програмних систем та мати клієнтську і серверну частини. Використана MySQL як рівень доступу до даних, що забезпечує централізоване структуроване зберігання всіх системних даних, гарантуючи їх цілісність і узгодженість, а також надає безліч низькорівневих сервісів. Рівень бізнес-логіки буде розгорнутий на сервері додатків і представлятиме ядро системи. Більша частина бізнес-логіки системи повинна бути зосереджена на цьому рівні.

2 АНАЛІЗ ІСНУЮЧИХ НА РИНКУ ІНФОРМАЦІЙНИХ СИСТЕМ ЩОТИЖНЕВИХ ЗВІТІВ

Початковий аналіз показав, що типовою та необхідною інформацією для розробки вебсайту інформаційної системи «Щотижневий звіт викладача», крім програмної частини, є архітектура створеного вебсервісу, принцип його роботи, оформлення навігаційного меню тощо. Також важливо знати інформацію про подані звіти, розташування, експозицію, новини, пов'язані роботою сайту, присвячену певній тематиці, наявність певних онлайн-сервісів, посилання на контактну інформацію, інформацію про ЗВО та методи його роботи. Крім усього перерахованого вище, у вебсервісах відображаються статті та новини, пов'язані зі звітами. Розглянемо детальніше кілька існуючих великих вебсервісів.

2.1 Система електронного документообігу Directum

Directum – система електронного документообігу та управління взаємодією, призначена для підвищення ефективності роботи всіх співробітників організації у всіх сферах їх спільної діяльності.

Система Directum – це зріла система ECM, яка підтримує повний життєвий цикл документообігу, тоді як традиційна «паперова» офісна робота органічно вписується в електронний документообіг. Directum забезпечує ефективну організацію та контроль бізнес-процесів на основі технології workflow: узгодження документів, комплексну обробку замовлень, підготовку та проведення зустрічей, підтримку циклу продажів та інші інтерактивні процеси.

Завдяки впровадженню системи електронного документообігу вирішено наступні завдання:

- обробка документів та їх узгодження всередині організації здійснюється тільки в електронному вигляді з використанням ЕЦП;
- використання документів у системі вищим керівництвом;

- контроль дисципліни виконання базується на системних даних;
- у системі працюють усі структурні підрозділи, включаючи головний апарат;
- створено єдине сховище електронних документів і корпоративних знань, усі співробітники можуть працювати за власними повноваженнями.

Основний функціонал системи визначається наступними характеристиками:

- існування інструментів для спрощення ідентифікації та пошуку паперових документів;
- можливість створювати та керувати сховищем документів практично не обмеженої ємності для різних типів вмісту протягом життєвого циклу документа в одній системі;
- розширений функціонал системи за рахунок модулів управління зустрічами, управління контрактами та управління взаємовідносинами з клієнтами;
- розширене впровадження механізмів електронного цифрового підпису (ЕЦП) для переходу на електронний документообіг;
- загальний механізм інтеграції, що спрощує роботу SAP, 1С та інших ERP систем в Directum;
- можливість організації територіально розподіленої роботи в автономному режимі, підтримка ієрархій серверів;
- існування двох способів для користувачів використовувати систему: через настільний клієнт і через веб-клієнт (за допомогою веб-браузера);
- можливість обміну електронними документами між різними організаціями за допомогою ЕЦП, навіть якщо вони використовують різні системи документообігу або не використовують їх зовсім;
- висока масштабованість системи забезпечує роботу тисяч користувачів і зберігання мільйонів різноманітних документів;

— наявність комплекту розробки (IS-BUILDER), що дозволяє швидко адаптувати систему, інтегрувати її з іншими системами, а також модифікувати самостійно з урахуванням розвитку організації.

Система Directum складається з кількох модулів (рис. 2.1).



Рисунок 2.1 – Модульна композиція СЕД «Directum»

Електронний документообіг. Створення та зберігання неструктурованих документів всіх видів (текст Microsoft Word, електронні таблиці Microsoft Excel, креслення Microsoft Visio, CorelDraw, відео тощо); підтримує версії документів і цифрові підписи; систематизація файлів у папки, призначення прав доступу до документів, робота з історією файлів; повнотекстовий та атрибутивний пошук документів.

Управління бізнес-процесами. Потік затвердження та обробки (docflow), який підтримує всі етапи життєвого циклу документа; видавати електронні завдання та контролювати їх виконання; взаємодії між працівниками під час діяльності; підтримка безкоштовного маршруту та жорсткого маршруту (робочий процес).

Ведення номенклатури справ з гнучкою схемою нумерації; контроль розподілу та розміщення паперових документів; організація ЕЦП для обміну електронними документами з іншими організаціями.

2.2 Система DocsVision

Платформа DocsVision є основою для електронного документообігу – створення системи управління документами та бізнес-процесами, що складається лише з основних елементів, призначених для зберігання інформації, доступу до неї та надання користувачам знайомого та зручного інтерфейсу.

Основними процесами, які автоматизуються за допомогою системи електронного документообігу DocsVision, є реєстрація вхідних/вихідних документів та протоколів нарад, підготовка, затвердження та оформлення внутрішніх розпорядчих документів, контроль виконання доручень.

Платформа складається з клієнтської та серверної частин. Серверний компонент дозволяє створювати всі необхідні структури зберігання даних (картки облікового запису та файли електронних документів) і підтримувати їх цілісність у майбутньому, забезпечуючи отримання, оновлення та видалення даних клієнтами відповідно до прав доступу.

Клієнтський компонент надає кінцевим користувачам інтерфейс для роботи з інформацією, що міститься в базі даних, включаючи її пошук, перегляд і модифікацію. Основним клієнтським компонентом платформи DocsVision є Navigator (працює як додаток у середовищі Microsoft Internet Explorer), який забезпечує доступ до всіх об'єктів DocsVision:

- перегляд дерева папок і карток, розташованих у папках;
- перегляд вмісту карток і їх групування за певними критеріями;
- відшукування картки за вказаними критеріями;
- створення нових документів та редагування існуючих.

Для наочності та зручності роботи з електронними документами картки згруповані в папки, подібні до папок Провідника, які забезпечують логічно впорядковане зберігання документів за правилами, встановленими організацією. Форми реєстрації документів – картотеки – можуть відрізнятися для кожної

організації та розробляються нею самостійно, тому є не частиною платформи DocsVision, а частиною вбудованого додатку DocsVision «Діловодство».

DocsVision об'єднує всі програми, створені на платформі, в єдиний простір для керування документами та процесами в організації. Основні можливості DocsVision:

- автоматизація управління підприємством;
- підтримка процесного підходу в організаційному менеджменті;
- автоматизація системи управління якістю;
- автоматизація процесів узгодження та затвердження документів;
- методи контролю за виконанням документів і завдань;
- налаштування доставки документи на робочі місця користувачів;
- управління опрацюванням документів;
- організація взаємодії з контрагентами та замовниками.

2.3 «1С: Документообіг 8»

«1С: Документообіг 8» не має галузевої специфіки і може ефективно використовуватися як в державному секторі, так і на комерційних підприємствах, будь то розподілена холдингова структура з великою кількістю користувачів або малий бізнес. Завдяки своїй універсальності програму можна легко налаштувати та адаптувати до специфіки конкретної організації.

Програмний продукт «1С: Документообіг 8» — програмний продукт, розроблений на новій технологічній платформі «1С: Підприємство 8.2», є наступником програмного продукту «1С: Архів 3» і використовується в сотнях організацій, підприємствах та установах.

«1С: Документообіг 8» дозволяє досягти наступних функцій:

- спростити роботу співробітників над документами, виключивши можливість втрати версій або перехресних фрагментів під час роботи;

- скоротити час пошуку необхідної інформації та загальний час централізованої обробки документів (проект, документація).

Програма працює на платформі «1С:Підприємство 8» і забезпечує операції:

- зберігання будь-яких файлів;
- обмеження доступу до файлів;
- колективне опрацювання документів;
- зберігання та контроль версій;
- видача та контроль виконання наказів;
- звітність;
- користування системою через веб-браузер.

Система дозволяє працювати з документами будь-якого типу. Програма, встановлена на локальному комп'ютері для перегляду або редагування документів. Текстові документи редагуються безпосередньо в 1С:Підприємство без виклику зовнішніх додатків.

Висновки до розділу 2

Початковий аналіз показав, що типовою та необхідною інформацією для розробки вебсайту інформаційної системи «Щотижневий звіт викладача», крім програмної частини, є архітектура створеного вебсервісу, принцип його роботи, оформлення навігаційного меню тощо. Було проаналізовано кілька існуючих великих вебсервісів. Directum, DocsVision, 1С: Документообіг 8 – системи управління документами та бізнес-процесами, що призначені для зберігання інформації, доступу до неї та надання користувачам знайомого та зручного інтерфейсу. Дані системи призначені для підвищення ефективності роботи всіх співробітників організації у всіх сферах їх спільної діяльності.

3 ЗАСОБИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

«ЩОТИЖНЕВИЙ ЗВІТ ВИКЛАДАЧА»

Вибір засобів розробки є важливим етапом при створенні нового програмного продукту. Правильний вибір наявних технологій полегшує розробнику етапи програмування, взаємодію майбутнього користувача із системою.

3.1 Вибір інструментальних засобів розробки

Інформаційна система буде представлена у вигляді вебсайту.

Front-end – це клієнт, тобто все про клієнта. Інтерфейс включає такі веб-технології – CSS, HTML, JavaScript, які використовуються для створення інтерфейсу користувача.

Back-end – це сторона сервера, де знаходиться ядро веб-сайту, і це ядро забезпечує основні функції та підтримку інтерфейсу. На серверній частині використовуються такі мови програмування сервера, як PHP, ASP.NET, Python, Ruby, Perl тощо. Серверні мови програмування призначені для роботи з такими базами даних, як MySQL, MongoDB та ін.

Опишемо веб технологію, на якій працюватиме сайт.

Back-end:

– PHP – мова програмування на стороні сервера, розроблена спеціально для написання веб-додатків. Спочатку він був створений для відстеження відвідувачів веб-сайту. З часом PHP перетворився з набору інструментів у повноцінну ООП (об’єктно-орієнтовану) мову програмування, код якої також можна вбудовувати в HTML. Коли користувач відвідує веб-сервер, PHP створює сторінку HTML і надсилає її назад у браузер користувача.

– MySQL – це система керування базами даних (СУБД). База даних – це структурований набір даних. Він може містити будь-яку інформацію – від простих списків співробітників до великих обсягів даних, які використовуються в корпоративних мережах та державних установах.

Front-end:

– HTML – це мова розмітки гіпертекстових документів. HTML описує структуру веб-сторінки за допомогою тегів (елементів HTML), які визначають розміщення тексту, графіки, таблиць тощо.

– CSS – це правила, які описують стиль (вигляд) документа HTML. За допомогою CSS веб-сторінки отримують візуально зрозумілий інтерфейс, тобто CSS описує правила тегів HTML, тексту та зображень, тим самим змінюючи відносне розташування, розмір, колір, шрифт та інші візуальні зміни елементів.

– JavaScript – це мова програмування на стороні клієнта. Він призначений для виконання будь-яких програмних операцій у браузері, забезпечуючи тим самим інтерактивність веб-сторінок. JavaScript дозволяє керувати елементами HTML, стилями CSS для веб-сторінок тощо.

Обмін даними між інтерфейсом і сервером використовує протокол HTTP. Це протокол для обміну інформацією в Інтернеті (рис. 3.1).

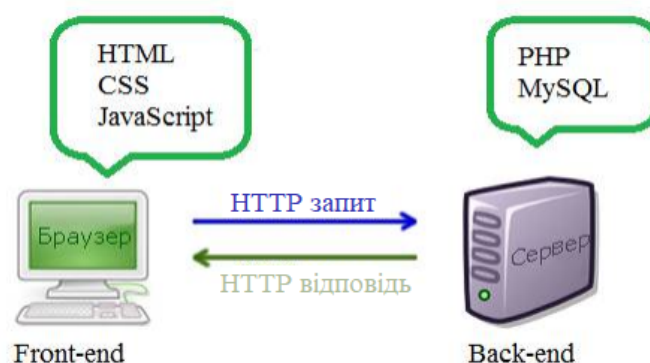


Рисунок 3.1 – Приклад взаємодії Front-end та Back-end.

Якщо користувач відвідує веб-ресурс, йому у відповідь приходить веб-сторінка (Front-end), яка була сформована сервером (Back-end).

У роботі було використано для реалізації Front-end частини CSS та JavaScript, а для Back-end частини використовується PHP та MySQL.

У роботі також було використано технологію CSS FlexBox.

Flexbox – це відносно нова інтерфейсна функція, яка робить створення макета вебсайту (і робить його адаптивним) набагато простішим, ніж це було раніше.

3.2 Опис середовища та системи

Для розробки вебсайту використовується середовище розробки Sublime Text.

Sublime Text – це кросплатформний власний текстовий редактор. Плагіни, які підтримують мову програмування Python. Розробники дозволяють ознайомитися з продуктом безкоштовно і без обмежень, але програма повідомляє про необхідність придбання ліцензії (рис. 3.2).

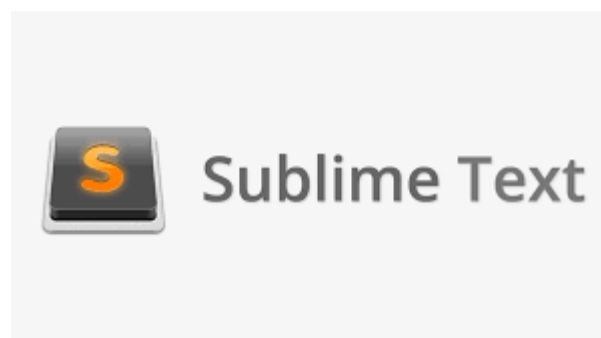


Рисунок 3.2 – Sublime Text

Sublime Text підтримує велику кількість мов програмування та забезпечує підтримку C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java,

JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL і XML [18].

На додаток до тих мов програмування, які включені за замовчуванням, користувачі можуть вибрати завантаження плагінів для підтримки інших мов [25].

Sublime Text може бути оснащений менеджером пакетів, який дозволяє користувачам знаходити, встановлювати, оновлювати та видаляти пакети без перезапуску програми. Менеджер підтримує встановлені пакети в актуальному стані, завантажуючи нові версії зі сховища. Крім того, він надає команди для активації та деактивації встановлених пакетів.

Редактор містить різноманітні візуальні теми, а додаткові теми доступні для завантаження.

Користувачі можуть бачити всі свої коди у вигляді міні-карти в правій частині екрана, на якій вони можуть переміщатися, натискаючи на неї.

Є кілька режимів екрану. Одна з них включає від 1 до 4 панелей, що дозволяє відображати до чотирьох файлів одночасно.

Повний режим (вільний режим) показує лише один файл без будь-яких інших меню навколо нього.

Якщо виділити цілі стовпці або розмістити кілька вказівників на тексті, то можна миттєво його редагувати. Індокси поводяться так, ніби кожен з них є унікальним індексом у тексті.

Типи команд:

- перехід до символу;
- перехід до рядка;
- виділення тексту;
- перехід до слова або його частини (CamelCase, розділені дефісом або підкресленням);
- перехід до початку/кінця рядка тощо.

Впливає на всі вказівники незалежно й одночасно, дозволяючи швидко редагувати складний текст без використання макросів чи регулярних виразів.

Коли користувач вводить код, Sublime Text надає різні варіанти завершення введення залежно від мови, яка використовується. Редактор також автоматично заповнює створені користувачем змінні.

Темний фон Sublime Text призначений для збільшення контрастності тексту. Основні елементи синтаксису виділені різними кольорами, які краще поєднуються з темним фоном, ніж зі світлим.

Sublime Text дозволяє користувачам компілювати програми та запускати їх без переходу до командного рядка. Користувачі також можуть налаштувати свою систему збірки та ввімкнути автоматичне збирання програми кожного разу, коли зберігається код. Інструмент навігації, який дозволяє користувачам переміщатися між файлами та всередині них за допомогою нечіткого пошуку. Присутня функція автозбереження, щоб допомогти користувачам не втратити виконану роботу.

Настроювані клавіатурні скорочення та інструменти навігації дозволяють призначати клавіатурні скорочення для меню та панелей інструментів (тільки для першої версії, у другій і третій – командна палітра).

Функція пошуку під час введення використовується для пошуку документів.

Засіб перевірки граматики працює подібним чином, перевіряючи правильність під час введення. Останню дію можна автоматизувати і повторити за допомогою макросів.

Для роботи з серверною частиною було обрано локальний сервер MAMP. MAMP означає Macintosh, Apache, MySQL і PHP (рис. 3.3).

Після встановлення MAMP все, що потрібно користувачу, є: Apache, MySQL і PHP.

Існує дві версії програми: платна MAMP PRO і безкоштовна MAMP, які відрізняються функціональністю та доступом до контенту та керування сервером.



Рисунок 3.3 – MAMP

Стандартний дистрибутив включає дві версії. Якщо використовувати безкоштовну версію, досить скопіювати її в папку з програмою і почати працювати. Якщо купити ліцензію на платний продукт, щоб використовувати MAMP PRO, то потрібно буде інсталювати обидві версії, оскільки MAMP і MAMP PRO складаються разом. В основному списку включених програм доступні всі ярлики для обох версій, але не рекомендується запускати їх одночасно.

Для роботи з таблицями даних використовується phpMyAdmin.

phpMyAdmin – це безкоштовний програмний інструмент, призначений для адміністрування СУБД MySQL через браузер (рис. 3.4). phpMyAdmin підтримує широкий спектр операцій на MySQL та MariaDB.

phpMyAdmin підтримує широкий спектр операцій на MySQL та MariaDB. phpMyAdmin дозволяє управляти:

- базами даних;
- таблицями;
- відносинами;
- індексами;
- користувачами;
- дозволами тощо.

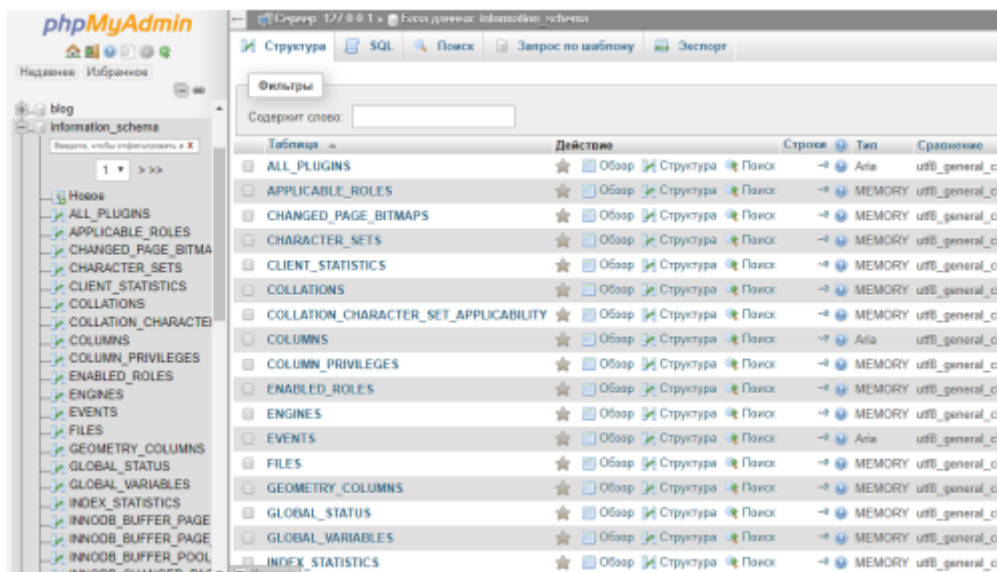


Рисунок 3.4 – phpMyAdmin

phpMyAdmin, дозволяє управляти базами даних, таблицями, відносинами, індексами, користувачами, дозволами тощо.

3.3 Основні вимоги до запуску та роботи з вебсайтом

Для інтерактивної обробки інформації розроблена система повинна містити такі інтерфейси:

- форма введення даних;
- форма для введення запиту для формування звітів та їх друку;
- форма вибору режимів друку вихідних документів;
- форма для редагування довідників.

3.3.1 Вимоги до інтерфейсів інформаційної системи

Витримувати колірну палітру інтерфейсів у стилі Windows. Роздільна здатність екрану: 1280*1024 – великий шрифт.

Інтерфейс повинен бути захищений від некоректних дій користувача та забезпечений системою підказок. Передбачити можливість виклику форм редагування довідників у разі потреби з форми введення та коригування даних.

3.3.2 Вимоги до технічного забезпечення

Для введення та корекції даних онлайн пропонується використовувати декілька комп'ютерів, підключених до локальної мережі.

Вимоги до сервера – персональний комп'ютер стандартної конфігурації, вільне місце на жорсткому диску 1 Гб. Вимоги до клієнта – відповідно до рекомендованих вимог операційної системи.

3.3.3 Вимоги до програмного забезпечення

На стороні сервера – операційна система Windows або Linux, СУБД MySQL, веб-сервер MAMP. Вимоги до клієнта – операційна система Windows XP, браузері – Firefox 3.6, Opera 10.

Висновки до розділу 3

У третьому розділі описані засоби розробки інформаційної системи «Щотижневий звіт викладача». Система представлена у вигляді вебсайту. Обмін даними між інтерфейсом і сервером використовує протокол HTTP. У роботі було використано для реалізації Front-end частини CSS та JavaScript, а для Back-end частини використовується PHP та MySQL. Також було використано технологію CSS FlexBox. Для розробки вебсайту використовується середовище розробки Sublime Text. Для роботи з серверною частиною було обрано локальний сервер MAMP. Для роботи з таблицями даних використовується phpMyAdmin.

Описані основні вимоги до запуску та роботи з вебсайтом.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ «ЩОТИЖНЕВИЙ ЗВІТ ВИКЛАДАЧА»

Розроблена інформаційна система дозволяє користувачам зручно та швидко викладачам створювати щотижневі звіти, а відповідальному їх перевіряти.

4.1 Структура вебсайту

У роботі була створена деревоподібна структура вебсайту. Початкова сторінка була розбита на частини:

- header;
- aside;
- main;
- footer.

Для підключення до бази даних був використаний код, наведений у додатку А.

У header є логотип, посилання на інші сторінки та іконка профілю. Код header винесено в окремий php файл для більш зручного використання багатосторінкового сайту. Код наведено у додатку А.

У aside зберігатися список документація, комунікація, фото, який завантажується через PHP з бази даних. Він також винесено в окремий php файл, що дає можливість швидкої зміни всього вмісту відразу у всіх файлах. Код наведено у додатку А.

В main відображається обрана категорія. Дані завантажуються з бази даних.

У footer відображається логотип сайту, можливі контакти і зв'язки. Footer також винесено в окремий php файл (Додаток А).

З головної сторінки можна перейти на наступні розділи:

- документація;
- комунікація;

- фото;
- відгуки;
- про нас;
- профіль.

Присутня можливість перейти на соціальні мережі. З кожною з них можна повернутися назад на головну натиснувши на логотип вебсайту.

4.2 Проектування та розробка БД

Для зберігання інформації про користувачів, документів, відгуків, комунікації, доцільно використовувати базу даних. Згідно з вибраним інструментарієм, база даних керується СУБД MySQL. У системі керування контентом застосовується СУБД MySQL. Спроектowana база даних було складається з шести таблиць:

- користувачі;
- відгуки;
- звіти;
- повідомлення;
- активні користувачі;
- зразки документів.

У таблиці «Користувачі» (табл. 4.1) зберігається інформація про користувачів, що зареєструвалися, як клієнти магазину. Таблиця містить 8 полів.

Таблиця 4.1 – Таблиця «Користувачі»

№	Атрибут	Семантика	Тип
1	2	3	
1	id	Унікальний ідентифікатор	int(5)
2	login	Логін для авторизації	varchar(50)

Продовження таблиці 4.1.

1	2	3	4
3	password	Пароль для авторизації	varchar(50)
4	sername	Прізвище користувача	varchar(50)
5	name	Ім'я користувача	varchar(50)
6	fotourl	Посилання на фото	varchar(50)
7	address	Адреса користувача	varchar(100)
8	phone	Телефон користувача	varchar(20)

У таблиці «Відгуки» (табл. 4.2) зберігаються відгуки клієнтів. Таблиця складається з 4 полів.

Таблиця 4.2 – Таблиця «Відгуки»

№	Атрибут	Семантика	Тип
1	id	Унікальний ідентифікатор товару	int(5)
2	textotziv	Текст відгуку	varchar(500)
3	username	Ім'я користувача	varchar(50)
4	fotourl	Фото користувача	varchar(50)

У таблиці «Звіт» (табл. 4.3) зберігається інформація про замовлення користувачів, що зареєструвалися.

Таблиця 4.3 – Таблиця «Звіт»

№	Атрибут	Семантика	Тип
1	2	3	4
1	id	Унікальний ідентифікатор товару	int(5)
2	name	Фото	varchar(100)
3	disciplina	Дисципліна	varchar(100)

Продовження таблиці 4.3.

1	2	3	4
4	communication	Засоби комунікації	varchar(100)
5	complate	Що зроблено	varchar(100)

Аналогічна структура і для інших таблиць.

Всі дані зберігаються на локальному сервері в окремих базах даних (рис. 4.1).

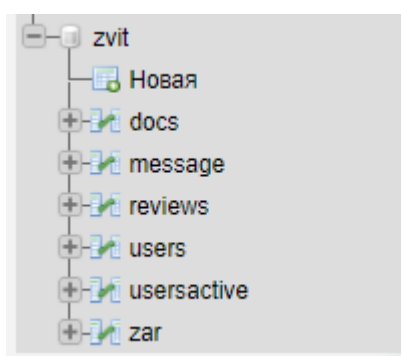


Рисунок 4.1 – Загальна структура бази даних

На рисунках 4.2-4.7 описані всі структури бази даних, які були створені для вебсайту.

+ Параметры					
		id	name	disciplina	communication
☐	Изменить	Копировать	Удалить	1	Иванов Иван Иванович
				Дипломна робота	Viber
					Перевірено роботу

Рисунок 4.2 – Дані в базі даних (Звіт)

				id	message	d	tosend
☐	Изменить	Копировать	Удалить	1001	Hello	2022-11-27 17:42:40	1002
☐	Изменить	Копировать	Удалить	1002	Hi	2022-11-27 17:42:48	1001
☐	Изменить	Копировать	Удалить	2		2022-12-03 15:28:39	

Рисунок 4.3 – Дані в базі даних (Повідомлення)

+ Параметры						
				id	textotziv	username fotourl
☐		Изменить		Копировать		Удалить
				1	Це дуже добрий сайт для роботи з документами	@svitlana_m src="img/p2.jpg"
☐		Изменить		Копировать		Удалить
				2	Nice	@alex_v src="img/p1.jpg"
☐		Изменить		Копировать		Удалить
				3	Я обираю цей сайт для контролю документів	@kira2012 src="img/p3.jpg"
☐		Изменить		Копировать		Удалить
				4	Я задоволений цим сайтом	@vlad01 src="img/p4.jpg"
☐		Изменить		Копировать		Удалить
				5	Very well!!!	a89307 src="img/p3.jpg"

Рисунок 4.4 – Дані бази даних (Відгуки)

+ Параметры								
				id	login	password	sername	name fotourl
☐		Изменить		Копировать		Удалить		
				1	admin@gmail.com	admin	admin	admin src="uploads/p1.jpg"
								admin +1234567890

Рисунок 4.5 – Дані бази даних (Користувачі)

+ Параметры				id	d	ready
☐		Изменить		Копировать		Удалить
				1	1682192335	0

Рисунок 4.6 – Дані бази даних (Активні користувачі)

+ Параметры						
				id	fotoUrl	name description
☐		Изменить		Копировать		Удалить
				11	src="img/zar1.jpg"	Протокол Протокол про навчання
☐		Изменить		Копировать		Удалить
				12	src="img/zar2.jpg"	Протокол Протокол педагогічної ради
☐		Изменить		Копировать		Удалить
				13	src="img/zar3.jpg"	Заява Заява на підвищення кваліфікації
☐		Изменить		Копировать		Удалить
				14	src="img/zar4.jpg"	Заява Заява на звільнення
☐		Изменить		Копировать		Удалить
				15	src="img/zar5.jpg"	Наказ Премія

Рисунок 4.7 – Дані бази даних (Зразки)

Підключення до бази даних відбувається за допомогою коду, наведеного у додатку А.

4.3 Концептуальна та логічна моделі бази даних

База даних спроектована з урахуванням таких вимог:

1. **Мінімальна надмірність.** Дані, що зберігаються в БД, можуть містити як «корисну», так і «шкідливу» надмірність. Остання завжди має місце за відсутності концептуального представлення даних, коли кожен користувач створює окремий набір даних для своїх додатків. У цьому випадку, якщо кільком користувачам потрібні ті самі дані, то вони повинні бути повторені в кожному наборі. Така надмірність є неконтрольованою, оскільки про її існування користувачі можуть і не підозрювати. Інтеграція уявлень користувача в єдине концептуальне уявлення, як правило, усуває цю надмірність даних. До «корисної» надмірності можна віднести періодичні копії даних, що зберігаються в БД. Ця надмірність легко контролюється. Більше того, вона є необхідною, наприклад, для відновлення даних, зруйнованих за випадкових збоїв і в катастрофічних ситуаціях. Таким чином, вимогу мінімальної надмірності слід розуміти як усунення «шкідливої» (неконтрольованої) і зведення до мінімуму «корисної» (контрольованої) надмірності даних.

2. **Цілісність даних.** Означає, що в базі даних встановлені та коректно підтримуються взаємозв'язки між записами різних таблиць при завантаженні, додаванні та видаленні записів у зв'язаних таблицях, а також при зміні значень ключових полів.

3. **Безпека та секретність.** Забезпечує захист даних від апаратних та програмних збоїв, від катастрофічних та кримінальних ситуацій, а також від некомпетентного доступу до них.

4. **Незалежність даних.** Має на увазі можливість зміни структури БД без зміни прикладних програм користувачів. Розуміється у двох аспектах, а саме, як логічна та фізична незалежність. Логічна незалежність передбачає можливість зміни концептуальної схеми БД без зміни прикладних програм користувачів. Фізична незалежність передбачає можливість зміни способу розміщення даних на

фізичних носіях та (або) методів доступу до даних без зміни прикладних програм користувачів.

5. Продуктивність. Характеризується часом відповіді запити користувачів.

6. Гнучкість та здатність до розширення. Розуміється як здатність БД до нарощування даних, а також збільшення кількості можливих додатків та розширення функцій у межах кожної програми.

Процес проектування бази даних складається з наступних окремих кроків: концептуального проектування, проектування логічної структури бази даних, яка підтримується СУБД та описує уявлення користувача про дані, та вибору фізичної структури, що включає уявлення даних, методи доступу та фізичне групування даних.

Основним компонентом системи є модель даних (концептуальна модель). Безліч структур даних, що підтримуються концептуальної моделі, визначає набір операцій у мові маніпулювання даними.

Основою на вищесказаному, були створені концептуальна модель (рис. 4.8) та логічна модель (рис. 4.9)

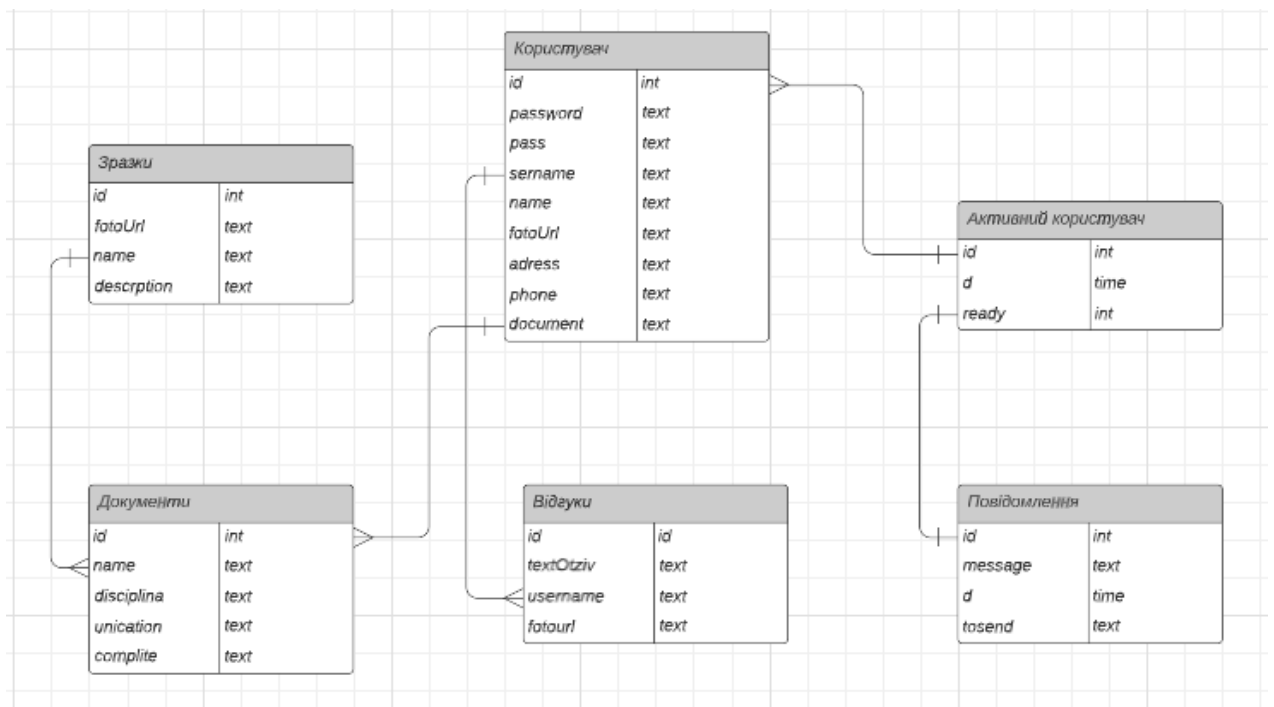


Рисунок 4.8 – Концептуальна модель

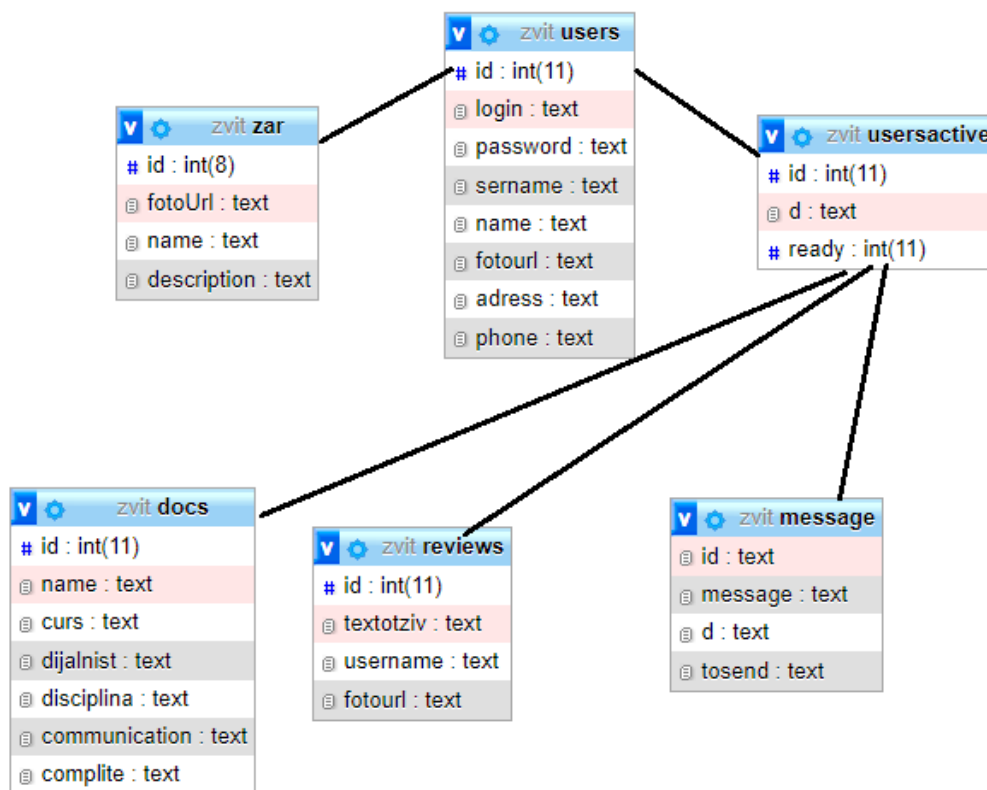


Рисунок 4.9 – Логічна модель

У процесі розробки автоматизованої системи управління дуже небезпечно пропустити будь-які вимоги. Вимоги до інформаційних систем можуть впливати з характеру предметної галузі. Задоволення вимог знижує помилки користувачів і, отже, підвищує надійність системи.

4.4 Аналіз можливих помилок

Дубльовані дані. Дублювання даних неминуче, коли користувачі часто використовують декілька систем. Ті самі дані можна ввести з різних систем. Ці системи потрібно поєднати, щоб не залишались дублікати записів. Необхідно запровадити процес перевірки даних із використанням інструментів дедуплікації

для аналізу даних і виявлення повторюваних записів, навіть якщо записи чи імена не ідентичні, але мають певну схожість. Оскільки кожен постачальник джерел даних має різні методи запису тієї самої інформації, інструмент дедуплікації розпізнає подібні точки даних і може позначати їх для дедуплікації.

Незаповнені поля. Незаповнені поля є типовою небезпекою, коли дані збираються вручну шляхом введення людиною. Користувач, який поспішає, іноді не завжди надає повну інформацію в звіті. Цю проблему можна вирішити за допомогою системи, яка не дозволяє подавати звіти, якщо всі поля не заповнені. Потрібно налаштувати систему керування даними, яка автоматично виключає неповні записи, групуючи їх для подальшого аналізу. Якщо використовувати стороннє рішення для керування даними, необхідно переконатися, що воно має стосунки з постачальником даних і може відстежувати втрачені дані

Несумісний формат. Різні формати даних завжди є проблемою для будь-якого аналітика. Наприклад, щось таке просте, як дата, можна ввести різними способами. Це може спричинити проблеми, особливо коли дані переміщуються з однієї системи в іншу, де неправильно витлумачені дати можуть призвести до великої кількості значних помилок даних. Пізніші розробки почали визначати формати, але застарілі протоколи не завжди їм відповідали.

При використанні джерел і постачальників даних необхідно вказати бажаний формат даних. Це все ще залишає проблеми. Потрібно переконатися, що існує автоматизований процес перевірки даних штучного інтелекту, щоб уникнути цих проблем на ранній стадії, щоб їх можна було виправити.

Різні мови та одиниці вимірювання. Як і у випадку з різними форматами, кожна країна має свою власну одиницю вимірювання. Додавання даних з двох різних одиниць або валют може легко призвести до помилкових цифр. З точки зору мови, вони можуть завдати шкоди, якщо система не налаштована на обробку спеціальних символів, таких як вирази та наголоси. Дублювати можна пропускати, а той самий готель можна надсилати кілька разів. Дані про подорожі, очевидно,

охоплюють різні мови та показники. Валюти також слід конвертувати одна в одну перед виконанням будь-яких розрахунків.

Людська помилка. Найбільшою перешкодою для якості даних є люди. Співробітники можуть робити помилки, що призводить до проблем із якістю даних, помилок і неправильних наборів даних. Єдиний спосіб обмежити це – максимально скоротити робочу силу.

Ми живемо у світі, де штучний інтелект щодня робить можливою автоматизацію. Замість того, щоб змушувати співробітників заповнювати звіти у документі Word чи Excel, можна створити сайт, який автоматично буде фіксувати всі види діяльності. Крім того, використання систем на основі штучного інтелекту та вдосконалених алгоритмів забезпечить мінімізацію людських помилок під час аналізу та передачі даних між системами.

Щоб підтримувати рішення, якість даних має бути надійною. Якщо виникне проблема з даними, у користувача виникнуть проблеми, та дані стануть ненадійними.

Одним із важливих компонентів роботи з даними є процес ETL (Extract, Transform, Load). ETL – це процес інтеграції даних, який складається з трьох окремих, але взаємопов'язаних кроків (вилучення, перетворення та завантаження) для ітеративного синтезу даних із багатьох джерел для створення сховища даних, концентратора даних або озера даних.

Три етапи складають процес ETL і дозволяють інтегрувати дані від джерела до цільового [7].

Крок 1: Витяг. Більшість керує даними з різних джерел і використовує різноманітні інструменти аналізу даних для отримання аналітики. Щоб створити таку складну стратегію даних, як ця вправа, дані повинні мати можливість вільно переміщатися між системами та програмами.

Перш ніж дані можна буде перемістити в нове місце призначення, їх потрібно витягти з джерела. На першому етапі процесу ETL структуровані та неструктуровані дані імпортуються та об'єднуються в єдине сховище.

Необроблені дані можуть надходити з різноманітних джерел, таких як існуючі бази даних і застарілі системи, хмарні, гібридні та локальні середовища, програми для продажів і маркетингу, системи CRM, інструменти аналітики тощо.

Інструменти ETL автоматизують процес майнінгу та створюють більш ефективні та надійні робочі процеси.

Крок 2: Перетворення. На цій стадії процесу ETL можна застосовувати правила та норми для забезпечення якості та доступності даних. Процес перетворення даних складається з кількох підпроцесів:

- очищення — видаляються невідповідності та відсутні значення в даних;
- нормалізація — правила форматування застосовуються до формату даних;
- дедуплікація даних — усуває або видаляє зайві дані;
- перевірка — видаляє недійсні дані та позначає винятки;
- сортувати — дані сортуються за типом;
- інші завдання — можна застосувати будь-які додаткові/необов'язкові правила для покращення якості даних.

Трансформація даних покращує цілісність даних і допомагає гарантувати, що дані є повністю сумісними та готовими до використання, коли вони досягають свого нового призначення.

Крок 3: Завантаження. Останнім кроком у процесі ETL є завантаження щойно перетворених даних у нове місце призначення. Усі дані можна завантажити одночасно (повне завантаження) або через заплановані проміжки часу (поступове завантаження).

Повне завантаження. У повністю завантаженому сценарії ETL усе з конвеєра перетворення переходить у нові унікальні записи в сховищі даних. Навантажуйте поступово. Менш складним, але більш керованим підходом є поступове завантаження. Поступове завантаження порівнює вхідні дані з існуючими та створює додаткові записи лише тоді, коли знайдено нову унікальну інформацію.

Ця архітектура дозволяє створити менші, менш дорогі сховища даних для підтримки та керування звітними процесами.

4.5 Опис основних форм інтерфейсу веб-сервісу

Основним компонентом будь-якого веб-сервісу є логотип (рис. 4.10).



Рисунок 4.10 – Логотип

Усі сторінки містять в собі такі елементи:

- Header;
- Aside (рис. 4.11);
- Footer.

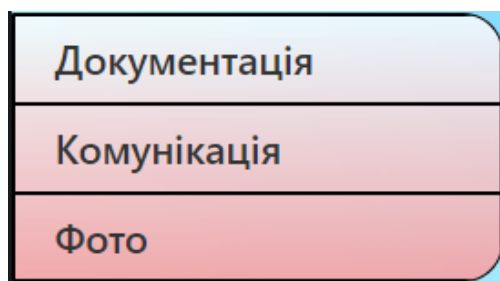


Рисунок 4.11 – Aside

Вміст даних тегів може змінювати в залежності від наповнення вебсервісу.

Висновки до розділу 4

У четвертому розділі описана програмна реалізація розробленої інформаційної системи. Була створена деревоподібна структура вебсайту. Початкова сторінка була розбита на частини: header; aside; main; footer.

Описана база даних, для проектування якої було зроблено 6 таблиць. Згідно з вибраним інструментарієм база даних керується СУБД MySQL. База даних спроектована з урахуванням певних вимог: мінімальна надмірність, цілісність даних, безпека та секретність, незалежність даних, продуктивність, гнучкість та здатність до розширення.

Основним компонентом системи є модель даних (концептуальна модель). Безліч структур даних, що підтримуються концептуальної моделі, визначає набір операцій у мові маніпулювання даними. Були створені концептуальна модель та логічна модель.

Проведено аналіз можливих помилок.

5 РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ

Розроблений веб-додаток потребує підключення до мережі Інтернет.

5.1 Опис функціоналу

Розглянемо роботу даної інформаційної системи.

При введенні адреси сайту відвідувач потрапляє на його головну сторінку (рис. 5.1).

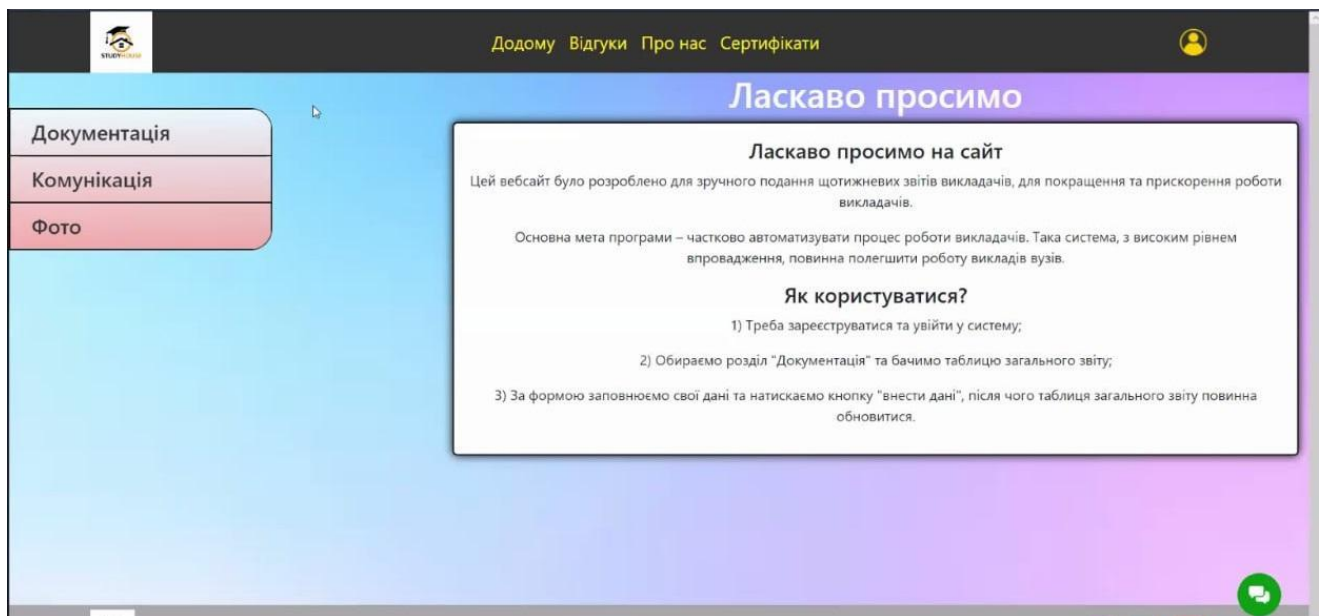


Рисунок 5.1 – Головна сторінка

Зовнішній вигляд товарів цієї категорії визначає адміністратор системи. На ній може бути подана інструкція про те, як користуватися цією системою.

З головної сторінки можна перейти на сторінку з фото, на сторінку з комунікацією або на сторінку з документаціями.

Вигляд сторінки «Фото», «Комунікація» та «Документація» представлені на рисунках 5.2, 5.3 та 5.4 відповідно.

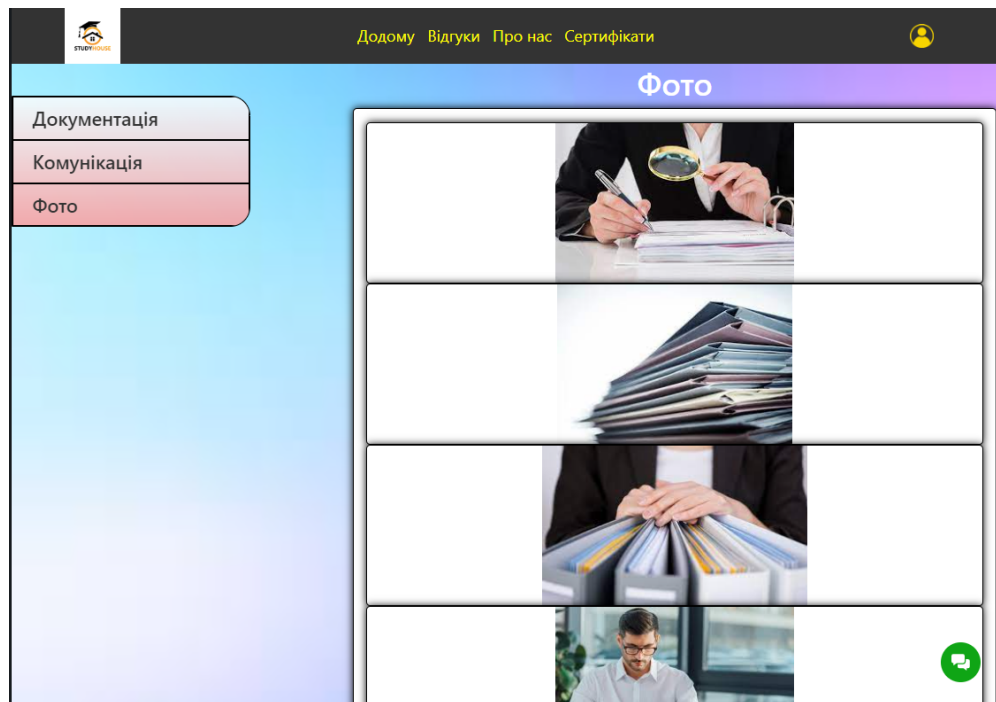


Рисунок 5.2 – Сторінка «Фото»

На сторінці «Комунікація» користувач може ввести повідомлення і відправити його іншому активному користувачу.

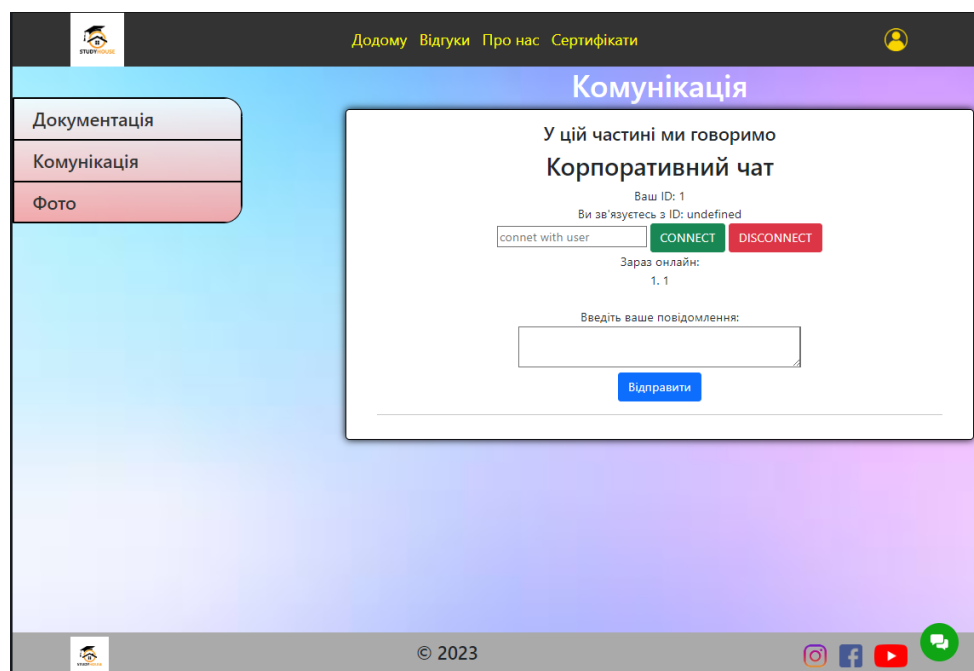


Рисунок 5.3 – Сторінка «Комунікація»

На сторінці «Документація» можна подати звіт, якщо ви увійшли у систему як зареєстрований користувач, заповнивши відповідні поля.

#	ПІБ Викладача	Курс	Діяльність	Назва дисципліни	Комунікація	Що зроблено
1	Іванов Іван Іванович	5	Магістерська дисертація	Дипломна робота	Viber	Перевірено роботу
2	Петров Петр Петрович	2	Наукова робота	Курсова робота	Telegram	Проведена лекція

Рисунок 5.4 – Сторінка «Документація»

Також є вкладка «Зразки документів» з переліком документів (рис. 5.5).

Рисунок 5.5 – Сторінка «Зразки документів»

Відповідальний за звіти може також додавати різноманітні зразки документів, які будуть корисними для викладачів кафедри:

- зразки протоколів засідання кафедри;
- різноманітні зразки заяв для викладачів;
- різноманітні зразки заяви для студентів, які куратор зможе скачати і передати їм;
- зразки службового подання для викладачів;
- зразки документів, які потрібно заповнити після календарних контролів та семестрового контролю як викладачам, так і студентам тощо.

Всі ці зразки документи наповнюються відповідальним протягом семестру та є доступними для зареєстрованого користувача. Якщо якийсь документ втрачає свою цінність, то його буде видалено із сайту.

Також з головної сторінки можна додати та переглянути відгуки користувачів сайту (рис. 5.6).

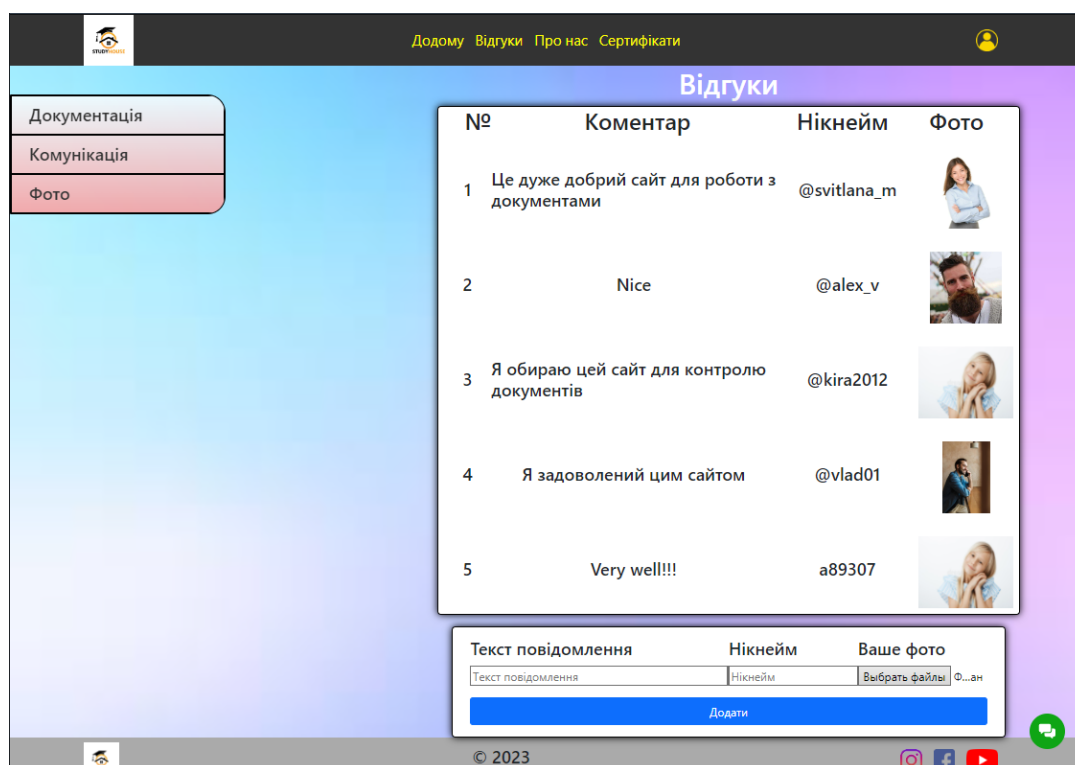


Рисунок 5.6 – Сторінка «Відгуки»

Ці всі відгуки будуть враховуватися при наповненні сайту адміністратором в майбутньому.

Розроблена інформаційна система є дуже простою і зручною у користуванні. Викладачі кафедри зможуть швидко заповнювати свої звіти і відправляти їх відповідальному.

5.2 Тестування розробленої інформаційної системи

Під час тестування системи усі виникаючі помилки одразу ж виправлялися. Нижче наведено декілька прикладів тестування роботи програми.

Проведений тест на реєстрацію користувача, який повинен заповнити відповідні поля необхідними даними (табл. 5.1):

- електронна пошта;
- пароль;
- ПІБ;
- адреса проживання;
- контактний телефон;
- фото.

Таблиця 5.1. Тест на реєстрацію користувача

Ідентифікатор тест-варіанту	Користувач реєструється в системі
Набір вхідних даних	введення власних даних
Очікувані результати	Користувач успішно зареєструвався у системі
Виконувані дії	Користувач заповнює відповідні поля та натискає на кнопку Реєстрація

Результат тесту на реєстрацію користувача наведено на рисунку 5.7.

Рисунок 5.7 – Результати тестування

Тест на реєстрацію користувача пройдено успішно.

Проведений тест на користувача в систему, який повинен ввести у відповідні поля свій логін та пароль (табл. 5.2).

Таблиця 5.2. Тест на вхід у систему

Ідентифікатор тест-варіанту	Вхід в систему
Набір вхідних даних	Логін та пароль
Очікувані результати	Користувач входить у систему
Виконувані дії	Користувач вводить логін та пароль

Якщо даний користувач введе невірні дані, то буде виведена помилка.

Якщо користувач забув свій пароль, то його можна буде відносити через пошту, яка була вказана при реєстрації.

Результат тесту на вхід в систему наведено на рисунках 5.8 – 5.9.

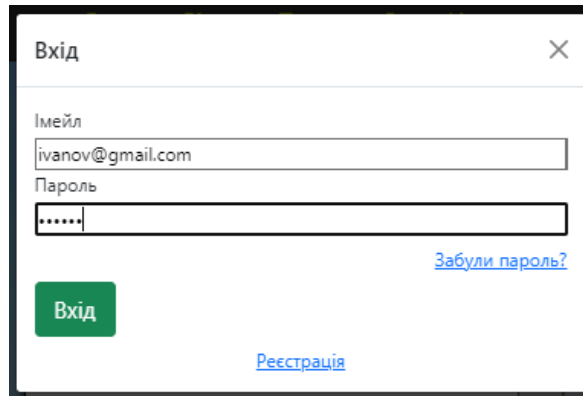


Рисунок 5.8 – Результати тестування на вхід в систему

На сторінці «Мій профіль» відображено фото та загальна інформація.

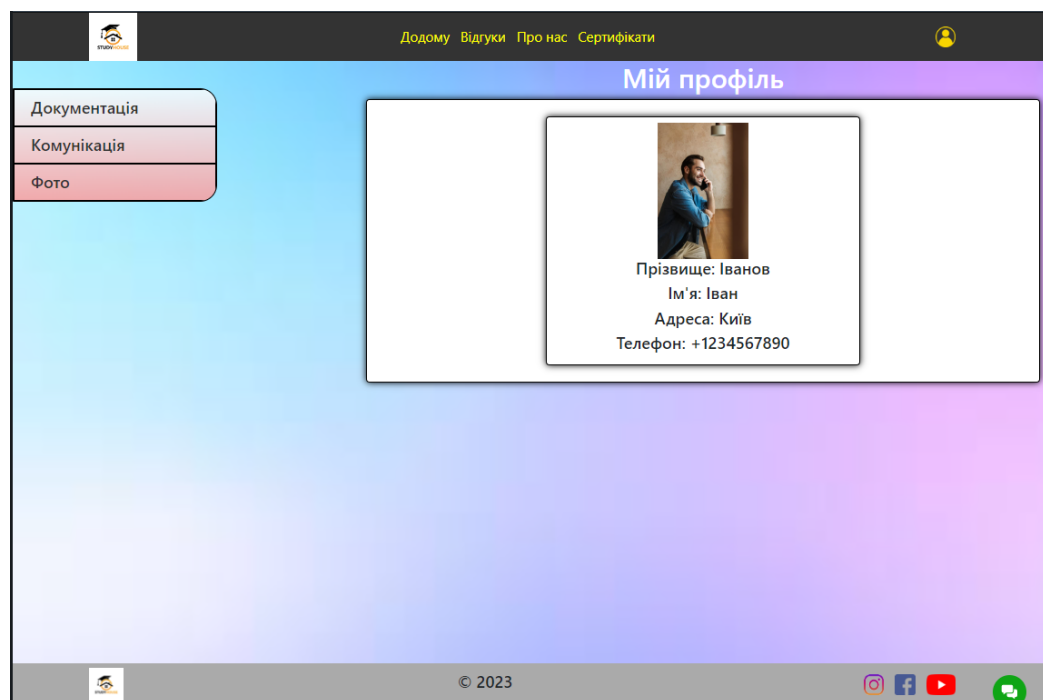


Рисунок 5.9 – Сторінка «Мій профіль»

Тест на вхід в систему пройдено успішно.

Проведений тест на додавання звіту користувачем, який повинен ввести у відповідні поля вхідні дані (табл. 5.3):

- ПІБ викладача;
- курс, на якому викладається навчальна дисципліна чи проводиться певна діяльність;

- вибрати зі списку діяльність;
- ввести назву дисципліни;
- обрати засоби електронної комунікації (e-mail, Viber, WhatsApp, Skype, Telegram, тощо);
- описати, що було зроблено.

Викладач натискає кнопку «Внести дані» і заповнює відповідні поля (рис. 5.10).

Звіти

Документація | Комунікація | Фото

Додавання файлу | Зразки документів

ПІБ: Сергієнко Дмит | Курс: 3 | Діяльність: Магістерська | Назва дисципліни: Курсова робота | Засоби комунікації: Viber | Що зроблено: Проведено коні

Внести дані

Номер

Очистити таблицю | Очистити за номером

#	ПІБ Викладача	Курс	Діяльність	Назва дисципліни	Комунікація	Що зроблено
1	Іванов Іван Іванович	5	Магістерська дисертація	Дипломна робота	Viber	Перевірено роботу
2	Петров Петр Петрович	2	Наукова робота	Курсова робота	Telegram	Проведена лекція

Зберегти

© 2023

Рисунок 5.10 – Заповнення полів звіту

Після того, як викладач заповнив один вид діяльності, нижче в таблиці він може переглянути свої записи.

Видів діяльності може бути декілька (рис. 5.11):

- навчальна дисципліна;
- дипломна/магістерська робота;
- науково-інноваційна робота;
- методична робота.

Звіти

Додавання файлу Зразки документів

ПІБ: Курс: Діяльність: Назва дисципліни: Засоби комунікації: Що зроблено:

Очистити таблицю Очистити за номером

#	ПІБ Викладача	Курс	Діяльність	Назва дисципліни	Комунікація	Що зроблено
1	Іванов Іван Іванович	5	Магістерська дисертація	Дипломна робота	Viber	Перевірено роботу
2	Петров Петр Петрович	2	Наукова робота	Курсова робота	Telegram	Проведена лекція
3	Сергієнко Дмитро	3	Магістерська дисертація	Курсова робота	Viber	Проведено консультацію

Зберегти

Рисунок 5.11 – Види діяльності у звіті

Якщо видів діяльності декілька, то потрібно знову проробити ту саму процедуру заповнення звіту. Якщо викладач допустив помилку при заповненні відповідної діяльності, то може внести зміни в певне поле і зберегти свій звіт, натиснувши на кнопку «Зберегти».

Таблиця 5.3. Тест на додавання звіту викладачем

Ідентифікатор тест-варіанту	Користувач додає звіт до сайту
Набір вхідних даних	ПІБ, Курс, Діяльність, Дисципліна, Комунікація, Що зроблено
Очікувані результати	Користувач успішно додає звіт
Виконувані дії	Користувач заповнює відповідні поля, та натискає «Внести дані»

Результат тесту наведено на рисунку 5.12.

Додому Відгуки Про нас Сертифікати

Звіти

Додавання файлу Зразки документів

Документація
Комунікація
Фото

ПІБ Курс Діяльність Назва дисципліни Засоби комунікації Що зроблено

ПІБ Курс Дисципліна Назва дисципліни Засоби комунікації Що зроблено

Внести дані

Номер

Очистити таблицю Очистити за номером

#	ПІБ Викладача	Курс	Діяльність	Назва дисципліни	Комунікація	Що зроблено
1	Іванов Іван Іванович	5	Магістерська дисертація	Дипломна робота	Viber	Перевірено роботу
2	Петров Петр Петрович	2	Наукова робота	Курсова робота	Telegram	Проведена лекція
3	Сергієнко Дмитро	3	Магістерська дисертація	Курсова робота	Viber	Проведено консультацію

Зберегти

Рисунок 5.12 – Результати тестування на додавання звіту викладачем

Після того, як всі викладачі заповнять свої звіти, відповідальний може скачати загальний звіт собі на робочий комп'ютер у вигляді Ексель-файлу (рис. 5.13).

	A	B	C	D	E	F	G
1	1	Іванов Іван Іванович	4	Бакалаврська робота		Viber	Перевірено звіти дипломних робіт гр.ТВ-391
2	2	Іванова Марина Яковлев	5	Магістерська дисертація		Telegram	Перевірено звіти магістерських робіт гр.ТА-41
3	3	Станіслав Ігоревич	2	Дисципліна	Основи програмування	Telegram , email	Проведено лекції та практики за розкладом засобами Google Meet. Проведено практичні роботи за розкладом в Google Meet.
4	4	Новікова Юлія Володимирівна		Методична робота	Розробка комп. програм	Viber	Написання лекцій
5	5	Педенко Ольга Анатолівна	3	Дисципліна	Організація комп. мереж	Viber, email	Проведені лабораторні заняття зі студентам груп ТФ-01, ТА-02 і ТУ-22.
6	6	Мороз Оксана Григорівна	1	Дисципліна	Безпека програмного забезпечення	Telegram , Viber, email	Проведено заняття через Zoom згідно розкладу; додатково вислано лекційних матеріал та завдання на виконання практичної роботи на e-mail групи. Створено гугл-диск для роботи зі студентами

Рисунок 5.13 – Загальний звіт викладачів

Тест на додавання звіту викладачем в систему пройдено успішно.

Вебсайт вийшов працездатний, та пройшов всі тестування і показав відмінний результат. Він може бути використаний на практиці на кафедрах закладів вищої освіти. В подальшому, вебсайт можна доповнювати різними функціями.

Висновки до розділу 5

В п'ятому розділі було описана робота користувача з веб-застосунком. Наведено вигляд всіх пунктів меню з детальним описом всіх сторінок. Розроблена інформаційна система є дуже простою і зручною у користуванні. Викладачі кафедри зможуть швидко заповнювати свої звіти і відправляти їх відповідальному.

Також проведено тестування розробленої системи «Щотижневий звіт викладача». Усі виникаючі помилки одразу ж виправлялися. Наведено декілька прикладів тестування роботи програми з описом всіх операцій, які можна здійснювати в системі. Усі тести в системі пройдено успішно.

Вебсайт вийшов працездатний, та пройшов всі тестування і показав відмінний результат. Він може бути використаний на практиці на кафедрах закладів вищої освіти. В подальшому, вебсайт можна доповнювати різними функціями.

ВИСНОВКИ

Метою кваліфікаційної роботи була розробка інформаційної системи «Щотижневий звіт викладача». У ході досягнення цієї мети стало можливим вирішити наступні завдання:

- 1) аналіз предметної області
- 2) проаналізовано новітні технології розробки вебсайтів;
- 3) проаналізовано особливості розвитку сучасних вебсайтів з точки зору проектування дизайну вебсайту;
- 4) вивчено сучасні засоби розробки вебсайтів;
- 5) реалізовано необхідний функціонал;
- 6) на локальному хостингу розміщено веб-сервіс.

В результаті кваліфікаційної роботи було розроблено програмний продукт (вебсайт), який функціонально забезпечує вирішення наступних завдань:

- оформлення щотижневих звітів викладача;
- контроль за ходом робіт;
- формування бази знань;
- формування звітів.

Застосування інформаційної системи дозволить зменшити витрати часу на роботу зі звітами, автоматизувавши цей процес та інших видів робіт за рахунок зниження трудомісткості роботи викладача. Інтерфейс користувача розроблено з урахуванням ергономічних вимог за допомогою технологій CSS FlexBox.

У першому розділі описана предметна область, мета даної роботи, основне завдання системи. Описані категорії користувачів та їхній поділ прав доступу і як саме вони реалізовані. Описаний функціонал системи, призначений для різних користувачів (адміністратора та викладачів).

У другому розділі проаналізовано кілька існуючих великих вебсервісів. Аналіз показав, що типовою та необхідною інформацією для розробки вебсайту інформаційної системи «Щотижневий звіт викладача», крім програмної частини, є архітектура створеного вебсервісу, принцип його роботи, оформлення навігаційного меню тощо.

У третьому розділі описані засоби інформаційної системи «Щотижневий звіт викладача». Описані основні вимоги до запуску та роботи з вебсайтом. Даний веб-сайт розроблено з використанням HTML+CSS+JS та мови програмування PHP та СУБД MySQL. Переважно через те, що вони безкоштовні та за рядом інших критеріїв, ці інструменти є найкращим вибором.

У четвертому розділі описана програмна реалізація розробленої інформаційної системи. Описана база даних, для проектування якої було зроблено 6 таблиць. База даних спроектована з урахуванням певних вимог. Проведено аналіз можливих помилок.

В п'ятому розділі було описана робота користувача з веб-застосунком. Наведено вигляд всіх пунктів меню з детальним описом всіх сторінок. Розроблена інформаційна система є дуже простою і зручною у користуванні. Викладачі кафедри зможуть швидко заповнювати свої звіти і відправляти їх відповідальному.

Також проведено тестування розробленої системи «Щотижневий звіт викладача». Усі виникаючі помилки одразу ж виправлялися. Наведено декілька прикладів тестування роботи програми з описом всіх операцій, які можна здійснювати в системі. Усі тести в системі пройдено успішно.

Вебсайт вийшов працездатний, та пройшов всі тестування і показав відмінний результат. Його можна буде використовувати в університеті при здачі різноманітних звітів. У подальшому, вебсайт можна доповнити різними функціями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ніксон Р. Створення динамічних вебсайтів за допомогою PHP, MySQL, JavaScript, CSS та HTML5. Ранок. 2016. 768 с.
2. Томсон В. Розробка веб-додатків за допомогою PHP та MySQL. Дніпро. 2017. 768 с.
3. Гаєвський А.Ю. 100% підручник зі створення веб-сторінок і вебсайтів: HTML і JavaScript. Одеса: Маяк. 2008. 457 с.
4. Макфарланд Д. Велика книга CSS3: навчальний посібник. Київ: Сяйво. 2014. 608 с.
5. Стратегія та основні кроки при розробці вебсайту. URL: <http://ruszura.in.ua/neobdidno-znaty/strategiya-i-osnovni-kroky-rzyrozo>. (дата звернення: 01.06.2023)
6. Горнаков С.Г. Освоюємо популярні системи керування сайтом. Київ: ДМК Прес. 2009. 336 с.
7. Мак-Дональд М. HTML5. Брак керівництва. Харків: Оберіг. 2016. 480 с.
8. Мак-Дональд М. Створення Web-сайту. Брак керівництва. Запоріжжя: Юніверс. 2018. 624 с.
9. Макфарланд Д. Нова велика книга CSS. Львів: Ранок. 2016. 720 с.
10. Мейєр Е., Вейл Е. CSS: повний довідник. Львів: ТОВ «Діалектика». 2019. 1088 с.
11. Ромашов, В. CMS Drupal. Система управління вмістом сайту. Запоріжжя: Юніверс. 2020 254 с.
12. Ши Д. Філософія CSS дизайну. Київ: НТ Прес. 2017. 312 с.
13. Бен Ф. HTML5 та CSS3. Розробка сайтів для будь-яких браузерів та пристроїв. Київ: Пітер. 2017. 304 с.
14. Дронов В.А. Розробка сучасних Web-сайтів. Київ: Либідь. 2019. 414 с.

15. Печніков В.М. Створення Web-сторінок та Web-сайтів. Самовчитель. Харків: Тріумф. 2019. 470 с.
16. Берковський В.В., Радіонов В.О., Бурковський В.С. Зіставлення технологій створення сайтів з їх цільовим призначення. Наука та техніка Повітряних Сил Збройних Сил України. 2017. № 2 (19). С. 122-124.
17. Борисов Р. І. Побудуйте професійний сайт самі. Київ: Техніка. 2019. 304 с.
18. Ажгіхіна С.Г., Марченко М.М. Проектування фірмового стилю. Київ: НТ Прес. 2015. 312 с.
19. Матросов А., Сергєєв А., Чаунін М., «HTML 4.0». Київ: Освіта. 2019. 304 с.
20. Фіелл П. Енциклопедія дизайну. Харків: «Аст-Астрель». 2021. 321 с.
21. Гладкий А.В. Веб-Самоделкін. Як самому створити сайт швидко та професійно. Київ: Ліра. 2018. 414 с.
22. Ваншнайдер М. Основи розробки веб-додатків за допомогою PHP та MySQL. Київ: Світ. 2018 р. 546 с.
23. Орлов З. Технології розробки програмного забезпечення: Підручник. Київ: Світ, 2020. 464 с.

ДОДАТОК А

Інформаційна система «Щотижневий звіт викладача»

Код програми

УКР.НТУУ”КПІ”_НН ІАТЕ_ІПЗЕ_ТВз9112647977_23А

Аркушів 18

Київ – 2023

header.php

```
<header>
  <div class="container">
    <div class="row">
      <div class="col-12 d-flex justify-content-between
align-items-center">
        <a href="index.php"></a>
        <nav class="navbar navbar-expand-lg navbar-light
bg-light">
          <div class="container-fluid
align-items-center">
            <button class="navbar-toggler" type="button"
data-bs-toggle="collapse" data-bs-target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
              <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse"
id="navbarNav">
              <ul class="navbar-nav">
                <li class="hp nav-item">
                  <a class="nav-link"
href="index.php">Додому</a>
                </li>
                <li class="hp nav-item">
                  <a class="nav-link"
href="reviews.php">Відгуки</a>
                </li>
                <li class="hp nav-item">
                  <a class="nav-link"
href="about.php">Про нас</a>
                </li>
                <li class="hp nav-item">
```

```

Сертифікати
    </li>
  </ul>
</div>
</div>
</nav>

<ul class="d-flex m-3 p-0">
  <li class="mx-2"><a data-bs-toggle="modal"
data-bs-target="#exampleModal"></a></li>
</ul>
</div>
</div>
</div>
</header>

<?php include('modal.php')?>

```

aside.php

```

<div class="col-12 aside scrollmenu p-0">
  <ul class="px-0 d-flex flex-row flex-md-column m-0">
    <li class="py-2 px-4"><a href="documentation.php">Документація</a></li>
    <li class="py-2 px-4"><a href="communication.php">Комунікація</a></li>
    <li class="py-2 px-4"><a href="photo.php">Фото</a></li>
  </ul>
</div>

```

footer.php

```
<footer>
  <div class="container">
    <div class="row">
      <div class="col-12 d-flex justify-content-between align-items-center">
        
        <h4>© 2023</h4>
        <ul class="p-0 m-3 d-flex">
          <li class="mx-2"><a target="_blank" href="https://www.instagram.com"></a></li>
          <li class="mx-2"><a target="_blank" href="https://www.facebook.com"></a></li>
          <li class="mx-2"><a target="_blank" href="https://www.youtube.com"></a></li>
        </ul>
      </div>
    </div>
  </div>
</footer>
```

database.php

```
<?php

$link = mysqli_connect('localhost','root','root','zvit');
if (mysqli_connect_errno())
{
    echo 'Mistake ('.mysqli_connect_errno().'): '.mysqli_connect_error();
}
```

```
exit();
}
```

```
?>
```

index.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstra
p.min.css" rel="stylesheet"
integrity="sha384-KyZXEAg3QhqLmP6G8r+8fhAXLRk2vvoC2f3B09zVXn8CA5QIVfZ
OJ3BCsw2P0p/We" crossorigin="anonymous">
  <?php include('css.php') ?>
  <title>Home</title>
</head>
<body>
  <?php include('header.php') ?>
  <main>
    <div class="row m-0">
      <div class="col-12 col-md-4 p-0">
        <?php include('aside.php') ?>
      </div>
      <div class="main_section col-12 col-md-8">
        <div class="col-12 d-flex flex-column
align-items-center">
          <h1>Ласкаво просимо</h1>
          <div class="wrapper p-3 col-12
text-center">
            <h4>Ласкаво просимо на сайт</h4>
```

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Voluptatum officiis eaque quisquam similique debitis fugit laborum nobis doloribus accusamus, dicta numquam itaque, beatae nisi nihil ducimus neque ad, doloremque quo atque. Fugit aut dolor minima ab sapiente. Eligendi soluta commodi provident sed, nulla explicabo, nam earum cum consectetur voluptas necessitatibus aliquid rerum, accusamus optio unde? Atque deserunt a libero explicabo incidunt quae tempore rem rerum earum nam tenetur amet harum similique commodi ducimus eligendi numquam eius, veritatis, sed quaerat cupiditate quia, illo. Voluptas debitis molestiae atque quasi nulla quae magni neque, consequatur veniam laborum repellat omnis nihil vel a, dolorum, sequi officia fugit nesciunt assumenda iure possimus odio. Pariatur facilis, magni eligendi dolorum, deserunt saepe praesentium deleniti aspernatur veniam blanditiis quasi accusantium cum commodi beatae, in at aut veritatis labore. Esse adipisci reiciendis, est ipsam illo dolorem ratione incidunt nulla perferendis minus dignissimos tenetur unde nisi? Dicta at sint recusandae omnis? Quo facilis unde numquam ipsa voluptatem esse architecto labore velit impedit, hic tenetur, animi error necessitatibus voluptatibus maiores assumenda totam atque quas veritatis, quaerat, vero quod modi. Velit at incidunt consequuntur libero unde repellat aut maxime omnis labore possimus voluptate hic veniam, sint ipsam illum, voluptates perferendis, et molestiae.</p>

</div>

</div>

</div>

</div>

</main>

<div class="chatbot__btn">

<div class="chatbot__tooltip d-none">Є питання?</div>

</div>

<?php include('footer.php') ?>

```
<?php include('script.php') ?>
</body>
</html>
```

style.css

```
.main_section{
  min-height: 84vh;
}
header{
  background: #333;
}

li{
  list-style-type: none;
}

.aside{
  z-index: 100;
  position: sticky;
  top: 110px;
  background-image: linear-gradient(rgba(255, 255, 255,
0.8), rgba(255, 151, 151, 0.8) );
  max-width: 300px;
  border: 1px solid black;
  max-height: 80vh;
  border-radius: 0 25px 25px 0;
}
@media (max-width: 768px){
  .aside{
    max-width: 100vh;
  }
}
.aside a{
  text-decoration: none;
```

```

    font-weight: 600;
    font-size: 24px;
    color: #333;
}
.aside li{
    border: 1px solid black;
}
.scrollmenu {
    overflow: auto;
    white-space: nowrap;
}
.scrollmenu::-webkit-scrollbar {
    height: 0px;
    width: 0;
}
.scrollmenu {
    -ms-overflow-style: none;
}
.scrollmenu {
    overflow: -moz-scrollbars-none;
}
.wrapper{
    border: 2px solid black;
    border-radius: 5px;
    box-shadow: 0 0 10px black;
    height: 100%;
    background-color: #fff;
}
.wrapper:hover{
    background-color: #aaa;
    transition-duration:1s;
}

img[alt="comp"]{

```

```

    height: 200px;
}

footer{
    background-color: #aaa;
}

.hp a{
    color: yellow !important;
    font-size: 20px;
    text-decoration: none;
}

.navbar{
    background: transparent!important;
}

.slider{
    background: #FFFFFF;
    box-shadow: 0px 0px 50px rgba(64, 74, 84, 0.06);
    border-radius: 20px;
}

.circle{
    border: 0px!important;
    border-radius: 7px;
    width: 14px!important;
    height: 14px!important;
    background: #E0E0E0!important;
}

.circle.active{
    background: #CFB88C!important;
}

#arrow_left2{
    display: none;
}

```

```

}
.carousel-control-prev:hover #arrow_left{
  display: none;
}
.carousel-control-prev:hover #arrow_left2{
  display: flex;
}
#arrow_right2{
  display: none;
}
.carousel-control-next:hover #arrow_right{
  display: none;
}
.carousel-control-next:hover #arrow_right2{
  display: flex;
}
main{
  background-image:url(../img/BG.jpg);
  background-repeat: no-repeat;
  background-size: cover;
  min-height: 84vh;
}
.menu a{
  text-decoration: none;
  color: orange;
  font-size: 24px;
}
.wrapper2, .wrapper3{
  border: 2px solid black;
  border-radius: 5px;
  box-shadow: 0 0 10px black;

  background-color: #fff;
}

```

```

.wrapper2 img{
  height: 100px;
}
#block-sort{
  display: none;
}
.wrapper3 img{
  height: 500px;
}
h1{
  color: white;
}

```

script.js

```

(function (factory) {
  if (typeof define === 'function' && define.amd) {
    // AMD (Register as an anonymous module)
    define(['jquery'], factory);
  } else if (typeof exports === 'object') {
    // Node/CommonJS
    module.exports = factory(require('jquery'));
  } else {
    // Browser globals
    factory(jQuery);
  }
})(function ($) {

  var pluses = /\+/g;

  function encode(s) {
    return config.raw ? s : encodeURIComponent(s);
  }

  function decode(s) {

```

```

        return config.raw ? s : decodeURIComponent(s);
    }

    function stringifyCookieValue(value) {
        return encode(config.json ? JSON.stringify(value) :
String(value));
    }

    function parseCookieValue(s) {
        if (s.indexOf('"') === 0) {
            // This is a quoted cookie as according to RFC2068,
unescape...
            s = s.slice(1, -1).replace(/\\"/g,
'").replace(/\\\\/g, '\\');
        }

        try {
            // Replace server-side written pluses with spaces.
            // If we can't decode the cookie, ignore it, it's
unusable.

            // If we can't parse the cookie, ignore it, it's
unusable.

            s = decodeURIComponent(s.replace(pluses, ' '));
            return config.json ? JSON.parse(s) : s;
        } catch(e) {}
    }

    function read(s, converter) {
        var value = config.raw ? s : parseCookieValue(s);
        return $.isFunction(converter) ? converter(value) : value;
    }

    var config = $.cookie = function (key, value, options) {

```

```

// Write

if (arguments.length > 1 && !$.isFunction(value)) {
    options = $.extend({}, config.defaults, options);

    if (typeof options.expires === 'number') {
        var days = options.expires, t = options.expires
= new Date();
        t.setMilliseconds(t.getMilliseconds() + days *
864e+5);
    }

    return (document.cookie = [
        encode(key), '=', stringifyCookieValue(value),
        options.expires ? '; expires=' +
options.expires.toUTCString() : '', // use expires attribute,
max-age is not supported by IE
        options.path ? '; path=' + options.path : '',
        options.domain ? '; domain=' + options.domain :
'',
        options.secure ? '; secure' : ''
    ].join(''));
}

// Read

var result = key ? undefined : {},
    // To prevent the for loop in the first place assign
an empty array
    // in case there are no cookies at all. Also prevents
odd result when
    // calling $.cookie().
    cookies = document.cookie ? document.cookie.split(';
') : [],

```

```

        i = 0,
        l = cookies.length;

    for (; i < l; i++) {
        var parts = cookies[i].split('='),
            name = decode(parts.shift()),
            cookie = parts.join('=');

        if (key === name) {
            // If second argument (value) is a function it's
a converter...

            result = read(cookie, value);
            break;
        }

        // Prevent storing a cookie that we couldn't decode.
        if (!key && (cookie = read(cookie)) !== undefined) {
            result[name] = cookie;
        }
    }

    return result;
};

config.defaults = {};

$.removeCookie = function (key, options) {
    // Must not alter options, thus extending a fresh
object...
    $.cookie(key, '', $.extend({}, options, { expires: -1 }));
    return !$.cookie(key);
};

});

```

```

function coockie(){
    var name = $("#login").val();
        var description = $("#pass").val();
        $.cookie('login',name);
        $.cookie('pass',description);
    }

function createCookie(name, value, days) {
    var expires;
    if (days) {
        var date = new Date();
        date.setTime(date.getTime() + (days * 24 * 60 * 60 *
1000));
        expires = "; expires=" + date.toGMTString();
    }
    else {
        expires = "";
    }
    document.cookie = name + "=" + value + expires + "; path=/";
}

function getCookie(c_name) {
    if (document.cookie.length > 0) {
        c_start = document.cookie.indexOf(c_name + "=");
        if (c_start != -1) {
            c_start = c_start + c_name.length + 1;
            c_end = document.cookie.indexOf(";", c_start);
            if (c_end == -1) {
                c_end = document.cookie.length;
            }
            return unescape(document.cookie.substring(c_start,
c_end));
        }
    }
}

```

```

        }
    }
    return "";
}

function exituser(){
    createCookie('user','',-1);
}

function mySortUp(){
    var nav=document.querySelector('.product_list');
    for (var i = 0; i < nav.children.length-1; ++i) {
        for (var j = i+1; j < nav.children.length; ++j) {
            if
(+nav.children[i].getAttribute('data-sort')>+nav.children[j].getAttr
ibute('data-sort')){
                replaceNode = nav.replaceChild(nav.children[j],
nav.children[i]);
                insertAfter(replaceNode, nav.children[i]);
            }
        }
    }
}

function mySortDown(){
    var nav=document.querySelector('.product_list');
    for (var i = 0; i < nav.children.length-1; ++i) {
        for (var j = i+1; j < nav.children.length; ++j) {

            if
(+nav.children[i].getAttribute('data-sort')<+nav.children[j].getAttr
ibute('data-sort')){
                replaceNode = nav.replaceChild(nav.children[j],
nav.children[i]);
                insertAfter(replaceNode, nav.children[i]);
            }
        }
    }
}

```

```

        }
    }
}

function insertAfter(elem, refElem){
    return
refElem.parentNode.insertBefore(elem,refElem.nextSibling);
}

var show=0;
function block_show(){
    var ss=document.getElementById('block-sort');
    if (show==0)
    {
        show=1;
        ss.style.display='grid';
    }else{
        show=0;
        ss.style.display='none';
    }
}

function mySortPrice(){
    var nav=document.querySelector('.product_list');
    var
min=document.querySelector('#minprice').getAttribute('value');
    var
max=document.querySelector('#maxprice').getAttribute('value');

    for (var i = 0; i < nav.children.length; ++i) {
        if      (+nav.children[i].getAttribute('data-sort')<min      ||
+nav.children[i].getAttribute('data-sort')>max){
            nav.children[i].style.display='none';
        }
        else

```

```
        nav.children[i].style.display='flex';
    }
}
function inputMaxVal(a){
    $('#maxprice').attr('value',a.value);
}
function inputMinVal(a){
    $('#minprice').attr('value',a.value);
}
```