**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ І ЗВ’ЯЗКУ**

**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ**

**«ФАХОВИЙ КОЛЕДЖ ЗВ’ЯЗКУ ТА ІНФОРМАТИЗАЦІЇ**

**ДЕРЖАВНОГО УНІВЕРСИТЕТУ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ І ЗВ’ЯЗКУ»**

Циклова комісія інформаційних технологій

**ЗВІТ**

**з преддипломної практики**

на підприємстві:

"АРДО"

Виконав: студент 4 курсу, групи З-41 спеціальності

121 “Інженерія програмного забезпечення”

Дар’я ІВАНОВА Керівник Сергій ПУСТОВІТ

Нормоконтролер Сергій ПУСТОВІТ

Одеса – 2022 р.

**ЗМІСТ**

[ВСТУП 4](#_Toc190101691)

[2 МЕТА ТА ЗАВДАННЯ ПРАКТИКИ 5](#_Toc190101692)

[2.1 Мета практики 5](#_Toc190101693)

[2.2 Завдання практики 5](#_Toc190101694)

[3 ТЕОРЕТИЧНА ЧАСТИНА 6](#_Toc190101695)

[3.1 Основи розробки вебсайта 6](#_Toc190101696)

[3.2 Адаптивний дизайн і користувацький досвід 6](#_Toc190101697)

[3.3 Аналіз вимог 7](#_Toc190101698)

[4 ПРАКТИЧНА ЧАСТИНА 8](#_Toc190101702)

[4.1 Проєктування структури та інтерфейсу сайту 8](#_Toc190101703)

[4.2 Реалізація функціоналу вебсайту 9](#_Toc190101706)

[ВИСНОВОК 11](#_Toc190101715)

[Додаток А 13](#_Toc190101716)

[Додаток Б 13](#_Toc190101717)

[Додаток В 14](#_Toc190101718)

[Додаток Д 15](#_Toc190101719)

[Додаток Е 15](#_Toc190101720)

[Додаток Ж 16](#_Toc190101721)

[Додаток Л 16](#_Toc190101722)

# **ВСТУП**

Переддипломну практику я проходив у приватному підприємстві "АРДО" в місті Одеса, Україна, з 13 січня 2025 року по 7 лютого 2025 року. Основною метою моєї роботи під час практики було створення власного вебсайту для представлення та просування особистої діяльності в інтернеті.

ПП "АРДО" — провідна компанія, яка спеціалізується на проєктування, монтажі та обслуговуванні електронних систем. Основними напрямками діяльності підприємства є розробка рішень у сфері систем безпеки, відеоспостереження, телекомунікацій та інженерних мереж. Завдяки багаторічному досвіду роботи й професійній команді, компанія займає провідні позиції на ринку та активно співпрацює з комерційними, державними та приватними замовниками.

Практика в ПП "АРДО" надала мені унікальну можливість реалізувати навчальний проєкт, який відповідав моїм професійним цілям та значно сприяв його подальшому вдосконаленню. Проєкт полягав у розробці сучасного вебсайту, який включав теоретичну інформацію про діяльність, інтеграцію невеликого інтернет-магазину, створення бази даних, а також розробку та програмування функціональних елементів.

Особливу увагу під час роботи я приділив створенню зрозумілої структури сайту, розробці адаптивного дизайну та забезпеченню його зручності для користувачів. Робота над проєктом вимагала комплексного підходу, який включав дизайн, програмування, організацію баз даних і тестування всіх функцій для забезпечення стабільної роботи сайту.

# **2 МЕТА ТА ЗАВДАННЯ ПРАКТИКИ**

## **Мета практики**

Метою переддипломної практики є розробка вебсайту. Вебсайт повинен містити сторінку для продажу товарів із каталогом та функціоналом оформлення замовлень та адмін-панель. Практика дозволяє поєднати теоретичні знання із реальними особистими потребами та розвинути навички веброзробки, розробки бази данніх та ботів для інших програмніх забезпечень що є важливим для моєї майбутньої професії.

## **Завдання практики**

Для досягнення поставленої мети були сформульовані такі завдання:

* вивчення основ проєктування та створення вебсайтів.
* розробка структури та інтерфейсу вебсайту.
* створення інтерактивних елементів із використанням сучасних технологій (HTML, CSS, JavaScript, php).
* створення дизайну для вебсайту.
* створення бази данних для збереження товарів та запитів.
* створення та розробка бота для Telegram.

# **3 ТЕОРЕТИЧНА ЧАСТИНА**

## **Основи розробки вебсайта**

Розробка вебсайту базується на трьох основних технологіях: HTML, CSS і JavaScript. HTML використовується для створення структури сторінки, визначаючи елементи, такі як заголовки, абзаци, зображення та посилання. CSS відповідає за стилізацію: задає кольори, шрифти, розміри елементів і забезпечує адаптивність сайту для різних пристроїв. JavaScript додає інтерактивність, наприклад, обробляє кліки на кнопки, перевіряє форми та динамічно оновлює контент без перезавантаження сторінки.

У процесі розробки важливим є використання принципів адаптивного дизайну, які забезпечують зручне відображення сайту на смартфонах, планшетах і комп’ютерах. Для спрощення роботи можна використовувати фреймворки, такі як Bootstrap, які прискорюють створення дизайну. Також необхідно тестувати сайт у різних браузерах, щоб забезпечити його кросбраузерність. Після завершення розробки сайт розміщується на сервері для доступу користувачів в Інтернеті.

## **Адаптивний дизайн і користувацький досвід**

Адаптивний дизайн дозволяє сайтам коректно відображатися на різних пристроях, таких як смартфони, планшети та комп’ютери. Для цього використовуються медіазапити в CSS, які задають стилі залежно від розмірів екрану. Основна мета адаптивного дизайну – забезпечити зручність користування сайтом без необхідності масштабування чи прокрутки вбік.

Користувацький досвід (UX) фокусується на тому, як зручно користувачеві взаємодіяти із сайтом. Інтуїтивна навігація є ключовим принципом UX: меню повинно бути зрозумілим, а переходи між розділами – легкими. Увага приділяється кольоровій гамі, яка не лише підкреслює стиль бренду, а й полегшує сприйняття інформації.

Користувацький інтерфейс (UI) відповідає за візуальну привабливість сайту. Використання сучасних трендів, таких як мінімалізм, плаский дизайн та інтерактивні елементи, робить сайт естетично привабливим і функціональним. Також важливо правильно вибрати шрифти – вони мають бути читабельними й гармонійно поєднуватися між собою. У підсумку, адаптивний дизайн і грамотний UX/UI-дизайн забезпечують комфортний досвід користувачів та сприяють залученню більшої аудиторії.

## **Аналіз вимог**

### Функціональні вимоги:

* перегляд товарів: сайт має показувати список товарів із їхніми зображеннями, назвами та цінами. Інформація про товари витягується з бази даних.
* оформлення замовлення: користувач може натискати кнопку "замовити", щоб відкрити форму для замовлення.
* форма замовлення: користувач заповнює форму з даними (ім'я, телефон, e-mail, коментар, адреса, тип доставки, товар), після чого дані відправляються на сервер через ajax-запит.
* збереження замовлення в базі даних: після отримання форми замовлення, дані зберігаються у базі даних.
* відправка повідомлення в telegram**:** після оформлення замовлення, інформація про замовлення відправляється у особисті повідомлення власника сайтув telegram.
* повідомлення: після успішного оформлення замовлення на сайті з'являється повідомлення.

### Невідкладні вимоги:

* підключення до бази даних: Сайт використовує базу даних для зберігання замовлень. Треба переконатися, що база даних налаштована та доступна.
* інтеграція з Telegram API: Для відправлення повідомлень необхідно правильно налаштувати Telegram Bot API, отримати токен бота та встановити доступи.
* базова валідація полів: Усі поля форми мають бути правильно перевірені на коректність вводу (особливо телефон та email).

### Технічні вимоги:

* Веб-сервер
* PHP
* База даних
* Telegram Bot API

# **4 ПРАКТИЧНА ЧАСТИНА**

## **Проєктування структури та інтерфейсу сайту**

### Структура сайту

Структура сайту визначається основними блоками, необхідними для забезпечення функціональності онлайн-магазину картин. Вона повинна бути логічною, зручною та зрозумілою для користувачів.

Приклад структури сайту:

* Сторінка продажу товарів
* Форма замовлення
* Telegram-сповіщення
* Адмін-панель

### Інтерфейс сайту

Інтерфейс сайту має бути простим, інтуїтивно зрозумілим і привабливим для користувача. Основні компоненти інтерфейсу включають:

а) головна сторінка:

1) заголовок

2) блок товарів

3) вікно повідомлень

б) форма замовлення:

1) поле для введення імені: Текстове поле для введення імені та прізвища покупця.

2) поле для введення телефону: Текстове поле з валідацією для телефонного номера (формат: +380XXXXXXXXX).

3) поле для введення e-mail: Поле для введення адреси електронної пошти з валідацією.

4) поле для введення коментаря: Текстова область для додаткових коментарів покупця.

5) поле для введення адреси доставки: Текстове поле для введення адреси доставки.

6) вибір типу доставки: Выпадаючий список з варіантами доставки (Нова пошта, Укрпочта).

7) кнопка для відправлення замовлення: Кнопка для відправлення форми. Після натискання формується AJAX-запит для обробки замовлення на сервері.

8) кнопка "Відміна": Кнопка для закриття форми замовлення.

в) адмін-панель:

1) загрузка товару

2) редагування товару

3) видаленяя товару

4) відображення усіх товарів

5) відображення усіх замовлень

6) видалення усіх замовлень

## **Реалізація функціоналу вебсайту**

### Підключення до бази даних (Додаток А)

Для роботи сайту використовується база даних MySQL, до якої підключається за допомогою mysqli. У коді є встановлення з’єднання через параметри servername, username, password, і dbname. Якщо з'єднання не вдається, виводиться повідомлення про помилку.

### Виведення списку товарів (Додаток Б)

Запит до бази даних отримує список товарів, включаючи назву, ціну та шлях до зображення. Товари виводяться в HTML через цикл PHP, що перевіряє наявність записів у таблиці items та показує кожен товар у вигляді картки з кнопкою для оформлення замовлення. Для кожного товару створюється блок із зображенням, назвою, ціною та кнопкою " Змовити ".

### Обробка замовлення (Додаток В)

При натисканні на кнопку "Змовити" відкривається форма замовлення. Форма містить поля для введення імені, телефону, електронної пошти, коментарів, адреси доставки та типу доставки. Дані з форми передаються через AJAX до сервера, не перезавантажуючи сторінку.

### Обробка форми замовлення через AJAX (Додаток Г)

Після заповнення та відправлення форми дані обробляються сервером через AJAX-запит. Форма перевіряє введення даних на відповідність, наприклад, номер телефону має бути у форматі +380XXXXXXXXX. За допомогою XMLHttpRequest дані передаються на сервер

### Збереження замовлення в базі даних (Додаток Д)

Після успішного отримання даних, замовлення зберігається в базі даних у таблиці orders. Для цього використовується підготовлений SQL-запит з параметрами, що запобігає SQL-ін'єкціям.

### Відправка сповіщень в Telegram (Додаток Е)

Після збереження замовлення в базі даних, адміністратор отримує повідомлення в Telegram через API бота. Сповіщення містить дані про замовлення: ім’я покупця, телефон, email, коментар, адресу доставки та тип доставки.

### Повідомлення про успішне оформлення замовлення (Додаток Ж)

Після успішної відправки замовлення користувачеві показується спливаюче повідомлення про те, що замовлення було успішно прийнято. Сповіщення зникає через кілька секунд після показу.

### Безпека

Дані з форми обробляються через htmlspecialchars(), що запобігає XSS-атакам. (Додаток К) Функція htmlspecialchars() перетворює спеціальні символи (наприклад, <, >, ", ', &) на їх HTML еквіваленти (наприклад, &lt;, &gt;, &quot;, &amp;). Це захищає від XSS-атак (Cross-Site Scripting), де зловмисник може вставити шкідливий JavaScript-код, що викликає небажану поведінку на сторінці або крадіжку даних користувача.

Всі запити на сервер здійснюються асинхронно за допомогою AJAX, що дозволяє не перезавантажувати сторінку. (Додаток Г) Використання AJAX дозволяє надсилати дані на сервер і отримувати відповідь без перезавантаження сторінки. Це забезпечує кращий користувацький досвід, оскільки сторінка не буде "заморожена", а користувач може продовжити взаємодію з сайтом. Запит до сервера обробляється асинхронно, і у разі успішного оформлення замовлення з'являється сповіщення, яке зникає через кілька секунд.

Telegram-бот: Використовується API Telegram для сповіщення адміністратора про нові замовлення, що робить процес більш зручним для моніторингу. (Додаток Л)

# **ВИСНОВОК**

Під час проходження переддипломної практики в приватному підприємстві "АРДО" я отримав унікальну можливість реалізувати проект, який відповідав моїм професійним цілям. Моя основна задача полягала в створенні вебсайту для представлення особистої діяльності в інтернеті, що включало розробку дизайну, програмування функціональних елементів, інтеграцію інтернет-магазину,створення телеграм-боту, а також створення бази даних. Зокрема, я приділив увагу зручності адміністратора, та забезпечив безпроблемну роботу з базою данних та доступом до замовлень саме в час іх відправлення.

Практика в ПП "АРДО" дозволила мені застосувати теоретичні знання на практиці, значно поглибити розуміння основ веб-розробки та взаємодії з користувачем, а також удосконалити навички роботи з базами даних та тестуванням програмних рішень. Робота в команді та вивчення процесів розробки та підтримки вебсайтів у реальних умовах сприяли розвитку важливих навичок, які я зможу використовувати у майбутній професійній діяльності.

**ПЕРЕЛІК ПОСИЛАНЬ**

1. Документація PHP: htmlspecialchars() — для безпеки при обробці даних з форми [електронний ресурс] Режим доступу: <https://www.php.net/manual/uk/function.htmlspecialchars.php>
2. AJAX: Asynchronous JavaScript and XML — для розуміння асинхронних запитів на сервер [електронний ресурс] Режим доступу: <https://developer.mozilla.org/uk/docs/Web/Guide/AJAX>
3. Розділ про діяльність компанії [електронний ресурс] Режим доступу: <https://www.ua-region.com.ua/24532240>
4. Використання PHP з CSS [електронний ресурс] Режим доступу: <https://www.delftstack.com/howto/php/php-css/>

# Додаток А

$servername = "localhost";

$username = "root";

$password = "\*\*\*\*\*\*\*\*";

$dbname = "store";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect\_error) {

    die("Помилка підключення: " . $conn->connect\_error);

}

# Додаток Б

$sql = "SELECT i.id, i.name, i.price, img.file\_path

        FROM items i

        LEFT JOIN images img ON i.id = img.items\_id";

$result = $conn->query($sql);

.

.

.

 if ($result->num\_rows > 0) {

            while ($row = $result->fetch\_assoc()) {

                echo "<div class='product'>

                        <img src='" . htmlspecialchars($row['file\_path']) . "' alt='Картина'>

                        <h3>" . htmlspecialchars($row['name']) . "</h3>

                        <p>Цена: " . htmlspecialchars($row['price']) . " грн</p>

                        <button onclick='openOrderForm(\"" . htmlspecialchars($row['name']) . "\")'>Замовити</button>

                      </div>";

            }

        } else {

            echo "<p>Товари не знайдені.</p>";

        }

# Додаток В

if ($\_SERVER['REQUEST\_METHOD'] == 'POST' && isset($\_POST['order'])) {

    $name = htmlspecialchars($\_POST['name']);

    $phone = htmlspecialchars($\_POST['phone']);

    $email = htmlspecialchars($\_POST['email']);

    $comment = htmlspecialchars($\_POST['comment']);

    $address = htmlspecialchars($\_POST['address']);

    $deliveryType = htmlspecialchars($\_POST['delivery\_type']);

    $productName = htmlspecialchars($\_POST['product\_name']);

    $stmt = $conn->prepare("INSERT INTO orders (name, phone, email, comment, address, delivery\_type, product\_name) VALUES (?, ?, ?, ?, ?, ?, ?)");

    $stmt->bind\_param("sssssss", $name, $phone, $email, $comment, $address, $deliveryType, $productName);

Додаток Г

document.getElementById('order-form-data').addEventListener('submit', function(event) {

            event.preventDefault();

            const formData = new FormData(this);

            const xhr = new XMLHttpRequest();

            xhr.open('POST', '', true);

            xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');

            xhr.onload = function() {

                const response = JSON.parse(xhr.responseText);

                if (response.status === 'success') {

                    document.getElementById('order-form').style.display = 'none';

                    document.getElementById('notification').style.display = 'block';

                    setTimeout(function() {

                        document.getElementById('notification').style.display = 'none';

                    }, 5000);

                } else {

                    alert(response.message);

                }

            };

            xhr.send(new URLSearchParams(formData).toString());

        });

# Додаток Д

if ($stmt->execute()) {

        $message = "Нове замовлення:\n\n" .

                   "Ім'я: $name\n" .

                   "Телефон: $phone\n" .

                   "E-mail: $email\n" .

                   "Комментар: $comment\n" .

                   "Адреса: $address\n" .

                   "Тип доставки: $deliveryType\n" .

                   "Товар: $productName";

        $telegramUrl = "https://api.telegram.org/bot$telegramToken/sendMessage";

        file\_get\_contents("$telegramUrl?chat\_id=$chatId&text=" . urlencode($message));

        echo json\_encode(['status' => 'success']);

    } else {

        echo json\_encode(['status' => 'error', 'message' => 'Ошибка при сохранении заказа: ' . $stmt->error]);

    }

# Додаток Е

$message = "Нове замовлення:\n\n" .

                   "Ім'я: $name\n" .

                   "Телефон: $phone\n" .

                   "E-mail: $email\n" .

                   "Комментар: $comment\n" .

                   "Адреса: $address\n" .

                   "Тип доставки: $deliveryType\n" .

                   "Товар: $productName";

        $telegramUrl = "https://api.telegram.org/bot$telegramToken/sendMessage";

        file\_get\_contents("$telegramUrl?chat\_id=$chatId&text=" . urlencode($message));

# Додаток Ж

document.getElementById('notification').style.display = 'block';

                    setTimeout(function() {

                        document.getElementById('notification').style.display = 'none';

                    }, 5000);

Додаток К

$name = htmlspecialchars($\_POST['name']);

    $phone = htmlspecialchars($\_POST['phone']);

    $email = htmlspecialchars($\_POST['email']);

    $comment = htmlspecialchars($\_POST['comment']);

    $address = htmlspecialchars($\_POST['address']);

    $deliveryType = htmlspecialchars($\_POST['delivery\_type']);

    $productName = htmlspecialchars($\_POST['product\_name']);

# Додаток Л

from flask import Flask, request, jsonify

from flask\_cors import CORS

import telegram

import asyncio

app = Flask(\_\_name\_\_)

CORS(app)

TELEGRAM\_TOKEN = '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

CHAT\_ID = '\*\*\*\*\*\*\*\*\*\*'

bot = telegram.Bot(token=TELEGRAM\_TOKEN)