

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра Програмної інженерії

Звіт
з лабораторної роботи №3
з дисципліни: «Скриптові мови програмування»
з теми: «Створення WEB-застосунків за допомогою PHP»

Виконав:
ст. гр. ПЗПІ-23-2
Ситник Є. С.

Перевірів:
Старший викладач кафедри ПІ
Сокорчук І. П.

3 СТВОРЕННЯ WEB-ЗАСТОСУНКІВ ЗА ДОПОМОГОЮ PHP

3.1 Історія змін

№	Дата	Версія звіту	Опис змін та виправлень
1	03.06.2025	0.1	Створено звіт

3.2 Мета роботи

Лабораторна робота полягає у розробці веб-застосунку «Продовольчий магазин Весна» засобами мови програмування PHP з використанням HTML, CSS та JavaScript для створення інтерактивного інтерфейсу користувача.

3.3 Хід роботи

3.3.1 Архітектура веб-застосунку

Для реалізації веб-застосунку було обрано модульний підхід з розділенням логіки на окремі файли:

3.3.1.1 Структура файлів:

- **index.php** – головна сторінка застосунку з логотипом магазину;
- **items.php** – сторінка каталогу товарів з можливістю додавання до кошика;
- **cart.php** – сторінка кошика з переглядом обраних товарів та управлінням;
- **handle_cart.php** – обробник AJAX-запитів для роботи з кошиком;
- **DB.php** – клас для роботи з базою даних SQLite;
- **style.css** – стилі для оформлення веб-сторінок.

3.3.1.2 Основні компоненти:

- **Клас DB** – забезпечує взаємодію з базою даних SQLite, включаючи методи для роботи з товарами, кошиком та налаштуваннями користувача;
- **Клас DbException** – спеціалізований виняток для обробки помилок бази даних;
- **Сесійна система** – для збереження стану користувача між сторінками.

3.3.2 Структура бази даних

Веб-застосунок використовує базу даних SQLite з трьома основними таблицями:

3.3.2.1 Таблиця settings:

- **name** (TEXT) – ім'я користувача;
- **age** (TEXT) – вік користувача.

3.3.2.2 Таблиця items:

- **id** (INTEGER PRIMARY KEY) – унікальний ідентифікатор товару;
- **name** (TEXT NOT NULL) – назва товару;
- **price** (REAL NOT NULL) – ціна товару.

3.3.2.3 Таблиця cart:

- **id** (INTEGER NOT NULL UNIQUE) – ідентифікатор товару;
- **count** (INTEGER NOT NULL) – кількість товару в кошику;
- **FOREIGN KEY** – зв'язок з таблицею items.

3.3.3 Функціональні можливості

3.3.3.1 Головна сторінка (index.php):

- Відображення логотипу магазину;
- Навігаційне меню з кількістю товарів у кошику;
- Привітання користувача за ім'ям;
- Підвал з інформацією про магазин.

3.3.3.2 Каталог товарів (items.php):

- Відображення списку доступних товарів з цінами;
- Форма для вибору кількості товару;
- Можливість додавання товарів до кошика;
- Валідація кількості товару (від 0 до 100 одиниць);

- Автоматичне оновлення лічильника кошика.

3.3.3.3 Кошик покупок (cart.php):

- Перегляд обраних товарів з детальною інформацією;
- Відображення кількості, ціни за одиницю та загальної вартості;
- Можливість видалення окремих товарів з кошика;
- Функція повного очищення кошика;
- Підрахунок загальної суми покупки;
- Кнопка «Сплатити» для завершення покупки.

3.3.3.4 Обробка запитів (handle_cart.php):

- Обробка POST-запитів для додавання товарів до кошика;
- Обробка DELETE-запитів для видалення товарів;
- Підтримка AJAX для асинхронного оновлення кошика;
- Валідація вхідних даних та обробка помилок.

3.3.4 Технічні особливості

3.3.4.1 Безпека:

- Використання підготовлених запитів (prepared statements) для запобігання SQL-ін'єкціям;
- Функція «htmlspecialchars()» для запобігання XSS-атакам;
- Валідація та фільтрація користувацьких даних через «filter_input()»;
- Обробка винятків для коректної роботи з помилками.

3.3.4.2 Користувацький інтерфейс:

- Адаптивний дизайн з використанням CSS;
- Інтуїтивна навігація між сторінками;
- Зручні форми для взаємодії з користувачем.

3.3.4.3 База даних:

- Автоматичне створення таблиць при першому запуску;
- Заповнення початковими даними;

3.3.5 Методи класу DB

Клас DB містить наступні основні методи:

- **retrieve_user()** – отримання інформації про користувача;
- **update_user()** – оновлення профілю користувача;
- **get_items()** – отримання списку всіх товарів;
- **get_item_by_id()** – отримання конкретного товару за ID;
- **get_cart()** – отримання товарів у кошику з детальною інформацією;
- **get_cart_count()** – підрахунок загальної кількості товарів у кошику;
- **get_cart_total()** – підрахунок загальної вартості кошика;
- **add_to_cart()** – додавання товару до кошика або оновлення кількості;
- **remove_from_cart()** – видалення товару з кошика;
- **empty_cart()** – повне очищення кошика.

3.4 Висновки

Під час виконання даної лабораторної роботи було успішно розроблено веб-застосунок інтернет-магазину з використанням сучасних веб-технологій. Зокрема, було освоєно:

- створення багатосторінкових веб-застосунків на PHP;
- роботу з базами даних SQLite через PDO з дотриманням принципів безпеки;
- реалізацію сесійної системи для збереження стану користувача;
- створення інтерактивного користувацького інтерфейсу з використанням HTML та CSS;
- валідацію та фільтрацію користувацьких даних.

Застосунок демонструє практичне застосування технологій веб-розробки для створення реальних рішень.

ДОДАТОК А

Відеозапис

Відеозапис презентації результатів лабораторної роботи: <https://youtu.be/Gils7poMkgk>

Хронологічний опис відеозапису:

00:00 – Вступ та загальний опис роботи

00:30 – Структура веб-застосунку

01:14 – Робота з базою даних та сесіями

01:43 – Обробка запитів користувача

03:56 – Демонстрація роботи веб-застосунку

ДОДАТОК Б

Програмний код

Б.1 Головна сторінка (index.php)

GitHub репозиторій: <https://github.com/NureSytnykYehor/smp-pzpi-23-2-sytnyk-yehor/blob/main/Lab3/smp-pzpi-23-2-sytnyk-yehor-lab3/index.php>

```

1  <?php
2  session_start();
3  require_once 'DB.php';
4
5  $db = new DB('shop.db');
6
7  if (!isset($_SESSION['user'])) {
8      $_SESSION['user'] = $db->retrieve_user();
9  }
10
11 $cart_count = $db->get_cart_count();
12 ?>
13
14 <!DOCTYPE html>
15 <html lang="uk">
16 <head>
17     <meta charset="UTF-8">
18     <title>Головна сторінка</title>
19     <link rel="stylesheet" href="style.css">
20 </head>
21 <body>
22     <header>
23         <h1>Продовольчий магазин "Весна"</h1>
24         <h3> Добрий день <?php echo $_SESSION['user']['name'] ?> </h3>
25         <nav>
26             <a href="index.php">Головна</a>
27             |
28             <a href="items.php">Товари</a>
29             |
30             <a href="cart.php">Кошик (<?php echo $cart_count ?? 0; ?
31 >)</a>
32         </nav>
33     </header>
34
35     <div class="container" style="display: flex; flex-direction:
36 column; align-items: center;">
37         
38     </div>
39
40     <footer>
41         <nav>
42             <a href="index.php">Головна</a>
43             |
44             <a href="items.php">Товари</a>
45             |

```

```

44         <a href="cart.php">Кошик (<?php echo $cart_count ?? 0; ?
    >)</a>
45     </nav>
46     <p>&copy; <?php echo date("Y"); ?> ТОВ "Весна". Усі права
    захищені.</p>
47 </footer>
48 </body>
49 </html>

```

Б.2 Каталог товарів (items.php)

GitHub репозиторій: <https://github.com/NureSytnykYehor/smp-pzpi-23-2-sytnyk-yehor/blob/main/Lab3/smp-pzpi-23-2-sytnyk-yehor-lab3/items.php>

```

1  <?php
2  session_start();
3  require_once 'DB.php';
4
5  $db = new DB('shop.db');
6
7  if (!isset($_SESSION['user'])) {
8      $_SESSION['user'] = $db->retrieve_user();
9  }
10
11 $items = $db->get_items();
12 $cart_count = $db->get_cart_count();
13 ?>
14
15 <!DOCTYPE html>
16 <html lang="uk">
17 <head>
18     <meta charset="UTF-8">
19     <title>Сторінка товарів</title>
20     <link rel="stylesheet" href="style.css">
21 </head>
22 <body>
23     <header>
24         <h1>Продовольчий магазин "Весна"</h1>
25         <h3> Добрий день <?php echo $_SESSION['user']['name'] ?> </h3>
26         <nav>
27             <a href="index.php">Головна</a>
28             |
29             <a href="items.php">Товари</a>
30             |
31             <a href="cart.php">Кошик (<?php echo $cart_count ?? 0; ?
    >)</a>
32     </nav>
33 </header>
34
35 <div class="container">
36     <h2>Доступні товари</h2>
37     <div class="product-list">
38         <?php foreach ($items as $item): ?>
39         <div>

```



```

40         <h2><?php echo htmlspecialchars($item['name']); ?
></h2>
41         <h3>Ціна: <?php echo number_format($item['price'],
2); ?> грн</h3>
42
43         <form action="handle_cart.php" method="POST">
44             <input type="hidden" name="product_id"
value="<?php echo htmlspecialchars($item['id']); ?>">
45             <label for="quantity_<?php echo
htmlspecialchars($item['id']); ?>">Кількість:</label>
46             <input type="number" id="quantity_<?php echo
htmlspecialchars($item['id']); ?>" name="quantity" value="0" min="0"
max="100">
47             <button type="submit">Купити</button>
48         </form>
49     </div>
50     <?php endforeach; ?>
51 </div>
52 </div>
53
54 <footer>
55     <nav>
56         <a href="index.php">Головна</a>
57         |
58         <a href="items.php">Товари</a>
59         |
60         <a href="cart.php">Кошик (<?php echo $cart_count ?? 0; ?
>)</a>
61     </nav>
62     <p>&copy; <?php echo date("Y"); ?> ТОВ "Весна". Усі права
захищені.</p>
63 </footer>
64 </body>
65 </html>

```

Б.3 Кошик покупок (cart.php)

GitHub репозиторій: <https://github.com/NureSytnykYehor/smp-pzpi-23-2-sytnyk-yehor/blob/main/Lab3/smp-pzpi-23-2-sytnyk-yehor-lab3/cart.php>

```

1 <?php
2 session_start();
3 require_once 'DB.php';
4
5 $db = new DB("shop.db");
6
7 if (!isset($_SESSION['user'])) {
8     $_SESSION['user'] = $db->retrieve_user();
9 }
10
11 $cart_items = $db->get_cart();
12 $cart_total = $db->get_cart_total();
13 $cart_count = $db->get_cart_count();
14 ?>

```

```

15
16 <!DOCTYPE html>
17 <html lang="uk">
18 <head>
19     <meta charset="UTF-8">
20     <title>Кошик</title>
21     <link rel="stylesheet" href="style.css">
22 </head>
23 <body>
24     <header>
25         <h1>Продовольчий магазин "Весна"</h1>
26         <h3> Добрий день <?php echo $_SESSION['user']['name'] ?> </h3>
27         <nav>
28             <a href="index.php">Головна</a>
29             |
30             <a href="items.php">Товари</a>
31             |
32             <a href="cart.php">Кошик (<?php echo $cart_count ?? 0; ?
33 >)</a>
34         </nav>
35     </header>
36     <div class="container">
37         <?php if (empty($cart_items)): ?>
38             <div style="display: flex; align-items: center; justify-
39 content: space-evenly;">
40                 <h3>Ваш кошик порожній <a href="items.php">Перейти до
41 покупок</a> </h3>
42             </div>
43         <?php else: ?>
44             <div style="display: flex; align-items: center; justify-
45 content: space-evenly;">
46                 <h3>Ваш кошик</h3>
47                 <h3 class="cart-summary">
48                     Загальна сума: <?php echo
49 number_format($cart_total, 2); ?> грн
50                 </h3>
51                 <button type="submit">Сплатити</button>
52                 <button onclick="fetch('handle_cart.php', {'method':
53 'DELETE'}).then(_ => { location.reload(); });">Очистити</button>
54             </div>
55             <div class="product-list">
56                 <?php foreach ($cart_items as $item): ?>
57                     <div>
58                         <h2>
59                             <?php echo
60 htmlspecialchars($item['name']); ?>
61                             <br>
62                             <?php echo
63 htmlspecialchars($item['count']); ?> шт.
64                         </h2>

```

```

62         <span>Ціна за одиницю: <?php echo
number_format($item['price'], 2); ?> грн</span>
63         <br>
64         <span>Загальна ціна: <?php echo
number_format($item['total_price'], 2); ?> грн</span>
65
66         <br><br>
67
68         <button
69             style="width: 100%;"
70             onclick="
71             fetch(
72                 'handle_cart.php?product_id=<?php echo
htmlspecialchars($item['id']); ?>',
73                 { 'method': 'DELETE' }
74                 ).then(_ => { location.reload(); });">
75             Видалити
76         </button>
77     </div>
78     <?php endforeach; ?>
79 </div>
80 <?php endif; ?>
81 </div>
82
83 <footer>
84     <nav>
85         <a href="index.php">Головна</a>
86         |
87         <a href="items.php">Товари</a>
88         |
89         <a href="cart.php">Кошик (<?php echo $cart_count ?? 0; ?
>)</a>
90     </nav>
91     <p>&copy; <?php echo date("Y"); ?> ТОВ "Весна". Усі права
захищені.</p>
92 </footer>
93 </body>
94 </html>

```

Б.4 Обробник кошика (handle_cart.php)

GitHub репозиторій: https://github.com/NureSytnykYehor/smp-pzpi-23-2-sytnyk-yehor/blob/main/Lab3/smp-pzpi-23-2-sytnyk-yehor-lab3/handle_cart.php

```

1 <?php
2 session_start();
3 require_once 'DB.php';
4
5 $db_path = 'shop.db';
6 $db = new DB($db_path);
7
8 switch ($_SERVER['REQUEST_METHOD']) {
9     case 'POST':

```

```

10     $product_id = filter_input(INPUT_POST, 'product_id',
FILTER_VALIDATE_INT);
11     $quantity = filter_input(INPUT_POST, 'quantity',
FILTER_VALIDATE_INT);
12
13     if ($product_id !== false && $product_id !== null &&
$quantity !== false && $quantity !== null) {
14         try {
15             $db->add_to_cart($product_id, $quantity);
16         } catch (DbException $e) {
17             error_log("Cart handling error: " . $e->getMessage());
18         }
19     }
20     header('Location: items.php');
21     break;
22
23     case 'DELETE':
24         $product_id = filter_input(INPUT_GET, 'product_id',
FILTER_VALIDATE_INT);
25
26         if ($product_id === null) {
27             try {
28                 $db->empty_cart();
29             } catch (DbException $e) {
30                 error_log("Cart handling error: " . $e->getMessage());
31             }
32         } else if ($product_id !== false) {
33             try {
34                 $db->remove_from_cart($product_id);
35             } catch (DbException $e) {
36                 error_log("Cart handling error: " . $e->getMessage());
37             }
38         }
39
40         header('Location: cart.php');
41         break;
42
43     default:
44         break;
45 }
46
47 exit();

```

Б.5 Клас для роботи з базою даних (DB.php)

GitHub репозиторій: <https://github.com/NureSytnykYehor/smp-pzpi-23-2-sytnyk-yehor/blob/main/Lab3/smp-pzpi-23-2-sytnyk-yehor-lab3/DB.php>

```

1 <?php
2
3 class DbException extends Exception {}
4
5 class DB
6 {

```

```

7     private $pdo;
8
9     /**
10      * Initializes database
11      *
12      * @param string $db_path
13      * @throws DbException If there's a database error.
14      */
15     public function __construct($db_path)
16     {
17         try {
18             $this->pdo = new PDO("sqlite:" . $db_path);
19             $this->pdo->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
20         } catch (PDOException $e) {
21             throw new DbException("Connection to DB failed.\nCaused
by: " . $e->getMessage());
22         }
23
24         // Ініціалізація таблиць та початкових даних
25         try {
26             $this->pdo->exec("
27                 CREATE TABLE IF NOT EXISTS settings (
28                     name TEXT,
29                     age TEXT
30                 );
31             ");
32             if ($this->pdo->query("SELECT COUNT(*) FROM settings;")-
>fetchColumn() == 0) {
33                 $this->pdo->exec("INSERT INTO settings (name, age)
VALUES ('user', 0);");
34             }
35         } catch (PDOException $e) {
36             throw new DbException("Error initialising settings table.
\nCaused by: " . $e->getMessage());
37         }
38
39         try {
40             $this->pdo->exec("
41                 CREATE TABLE IF NOT EXISTS items (
42                     id INTEGER PRIMARY KEY AUTOINCREMENT,
43                     name TEXT NOT NULL,
44                     price REAL NOT NULL
45                 );
46             ");
47             if ($this->pdo->query("SELECT COUNT(id) FROM items;")-
>fetchColumn() == 0) {
48                 $this->pdo->exec("
49                     INSERT INTO items (name, price) VALUES ('Молоко
пастеризоване', 32.50);
50                     INSERT INTO items (name, price) VALUES ('Хліб
чорний', 18.00);
51                     INSERT INTO items (name, price) VALUES ('Сир
білий', 85.00);
52                     INSERT INTO items (name, price) VALUES ('Сметана
20%', 45.80);

```

```

53             INSERT INTO items (name, price) VALUES ('Кефір
54 1%', 28.50);
55             INSERT INTO items (name, price) VALUES ('Вода
56 газована', 25.00);
57             INSERT INTO items (name, price) VALUES ('Печиво
58 \"Весна\"', 42.30);
59             INSERT INTO items (name, price) VALUES ('Масло
60 вершкове', 125.00);
61             INSERT INTO items (name, price) VALUES ('Йогурт
62 натуральний', 38.90);
63             INSERT INTO items (name, price) VALUES ('Сік
64 апельсиновий', 55.00);
65         });
66     }
67     } catch (PDOException $e) {
68         throw new DbException("Error initialising items table.
69 \nCaused by: " . $e->getMessage());
70     }
71     try {
72         $this->pdo->exec("
73             CREATE TABLE IF NOT EXISTS cart (
74                 id INTEGER NOT NULL UNIQUE,
75                 count INTEGER NOT NULL,
76                 FOREIGN KEY(id) REFERENCES items(id) ON DELETE
77 CASCADE
78             );
79         ");
80     } catch (PDOException $e) {
81         throw new DbException("Error initialising cart table.
82 \nCaused by: " . $e->getMessage());
83     }
84 }
85
86 /**
87  * Retrieve user information from the database
88  *
89  * @return array
90  * @throws DbException If there's a database error.
91  */
92 public function retrieve_user(): array
93 {
94     try {
95         $stmt = $this->pdo->query("SELECT name, age FROM
96 settings;");
97         return $stmt->fetch(PDO::FETCH_ASSOC);
98     } catch (PDOException $e) {
99         throw new DbException("Error retrieving user info.\nCaused
100 by: " . $e->getMessage());
101     }
102 }
103
104 /**
105  * Fetches all items from the database.
106  *
107  * @return array[]

```

```

98      * @throws DbException If there's a database error.
99      */
100     public function get_items(): array
101     {
102         try {
103             $stmt = $this->pdo->query("SELECT id, name, price FROM
items ORDER BY id;");
104             return $stmt->fetchAll(PDO::FETCH_ASSOC);
105         } catch (PDOException $e) {
106             throw new DbException("Error retrieving data from the
items table.\nCaused by: " . $e->getMessage());
107         }
108     }
109
110     /**
111      * Fetches all items in the cart from the database.
112      *
113      * @return array[]
114      * @throws DbException If there's a database error.
115      */
116     public function get_cart(): array
117     {
118         try {
119             $stmt = $this->pdo->query(
120                 "SELECT
121                     cart.id,
122                     items.name,
123                     items.price,
124                     cart.count,
125                     ROUND(items.price * cart.count, 2) as total_price
126                 FROM cart
127                 INNER JOIN items ON cart.id = items.id
128                 ORDER BY cart.id;"
129             );
130             return $stmt->fetchAll(PDO::FETCH_ASSOC);
131         } catch (PDOException $e) {
132             throw new DbException("Error retrieving cart items.
\nCaused by: " . $e->getMessage());
133         }
134     }
135
136     /**
137      * Get total items count in cart
138      *
139      * @return int
140      * @throws DbException If there's a database error.
141      */
142     public function get_cart_count(): int
143     {
144         try {
145             $stmt = $this->pdo->query("SELECT COALESCE(SUM(count), 0)
FROM cart;");
146             return (int)$stmt->fetchColumn();
147         } catch (PDOException $e) {
148             throw new DbException("Error getting cart count.\nCaused
by: " . $e->getMessage());

```

```

149     }
150 }
151
152 /**
153  * Get total price of all items in cart
154  *
155  * @return float
156  * @throws DbException If there's a database error.
157  */
158 public function get_cart_total(): float
159 {
160     try {
161         $stmt = $this->pdo->query(
162             "SELECT COALESCE(SUM(items.price * cart.count), 0.0)
163             FROM cart
164             INNER JOIN items ON cart.id = items.id;"
165         );
166         return (float)$stmt->fetchColumn();
167     } catch (PDOException $e) {
168         throw new DbException("Error calculating cart total.
169 \nCaused by: " . $e->getMessage());
170     }
171
172 /**
173  * Add item to the cart or update its quantity.
174  *
175  * @param int $id
176  * @param int $count
177  *
178  * @return bool
179  * @throws DbException If there's a database error or item doesn't
180 exist.
181 */
182 public function add_to_cart($id, $count): bool
183 {
184     try {
185         // Check if the item exists
186         $item = $this->get_item_by_id($id);
187         if (!$item) {
188             throw new DbException("Item with ID $id does not
189 exist.");
190         }
191
192         // If count is 0 or less, remove the item from the cart
193         if ($count <= 0) {
194             return $this->remove_from_cart($id);
195         }
196
197         // Insert or update the cart item
198         $stmt = $this->pdo->prepare(
199             "INSERT INTO cart (id, count)
200             VALUES (:id, :count)
201             ON CONFLICT(id) DO UPDATE SET
202                 count = excluded.count;"
203         );

```



```

202         return $stmt->execute(['id' => $id, 'count' => $count]);
203     } catch (PDOException $e) {
204         throw new DbException("Error adding/updating item in the
cart.\nCaused by: " . $e->getMessage());
205     }
206 }
207
208 /**
209  * Empty the cart
210  *
211  * @return bool
212  * @throws DbException If there's a database error.
213  */
214 public function empty_cart(): bool
215 {
216     try {
217         $stmt = $this->pdo->prepare("DELETE FROM cart");
218         return $stmt->execute();
219     } catch (PDOException $e) {
220         throw new DbException("Error removing item from the cart.
\nCaused by: " . $e->getMessage());
221     }
222 }
223
224 /**
225  * Remove an item from the cart
226  *
227  * @param int $id
228  *
229  * @return bool
230  * @throws DbException If there's a database error.
231  */
232 public function remove_from_cart($id): bool
233 {
234     try {
235         $stmt = $this->pdo->prepare("DELETE FROM cart WHERE id
= :id");
236         return $stmt->execute(['id' => $id]);
237     } catch (PDOException $e) {
238         throw new DbException("Error removing item from the cart.
\nCaused by: " . $e->getMessage());
239     }
240 }
241 }

```