

UNIVERSITATEA DE STAT DIN MOLDOVA

FACULTATEA „MATEMATICĂ ȘI INFORMATICĂ”

DEPARTAMENTUL „INFORMATICĂ”

CEMÎRTAN CRISTIAN

Lucrul individual

la disciplina „PROGRAMAREA WEB”

PROIECT DE INVESTIGAȚIE

Site pentru a vinde produse de calcul – „Magazin Hardware”

Conducător științific: Pleșca Natalia, lect. univ.

Chișinău – 2023

Cuprins

Descrierea domeniului studiat.....	3
Diagrama Entitate-Relație a bazei de date	4
Scenariu SQL	5
Proiectarea arhitecturii site-ului	7
Structura fișierelor	7
Machetare, stilizare și scriptare.....	12
Pagina principală.....	12
Pagina cu produse	14
Pagina de finalizare unei comenzii	16
Designul în general	18
Compartimentul „Contul meu”	20
Pagina „Sign-up”	20
Pagina principală.....	22
Pagina „Utilizatori”	24
Pagina „Produse”	26
Pagina „Tranzacții”	28
Concluzii	30
Bibliografie	31

Descrierea domeniului studiat

În acest lucru individual, am elaborat un site specializat vinderii a produselor de calcul. Site-ul este bazat pe un magazin online fictiv, denumit „Magazin Hardware”. Istoric, acest proiect a fost demarat în anul 2020, când învățam anul II la CEITI, ca o lucrare individuală pentru disciplina „Sisteme de gestiune a bazelor de date”. Lucrarea era bazată numai pe limbajul SQL, și la nivel de schițare a diagramei entitate-relație a bazei de date.

În anul III, proiectul s-a avansat într-un program vizual scris în limbajul de programare C#. Acest program simula un SGBD destinat simulării unei administrări realiste/practice a bazei de date schițate anul precedent.

Pe data de 30 martie 2022, versiunea compactă a proiectului „Magazin Hardware” a fost publicată la conferința tehnico-științifică (ediția 2022), în cadrul UTM. Întitulat **„INTEROPERABILITATEA DINTRE JDBC ȘI HIBERNATE PRIN INTERMEDIUL VAADIN”**, această versiune a fost realizată sub formă de serviciu web, în loc de program vizual. Aplicația web, scrisă în Java și prin intermediul componentelor Vaadin, compara nivelele de administrare a BD executate utilizând Hibernate și JDBC, concluzionând că framework-ul Hibernate este învingătorul în acest domeniu de administrare.

Ca ediție recentă a proiectului, această lucrare individuală reprezintă starea matură a site-ului întreprinderii „Magazin Hardware”. Pentru realizare, am utilizat următoarele tehnologii: PHP, MySQL, HTML, CSS și JavaScript. În deosebire de site-ul scris în Java, această ediție a fost realizată de la zero.

Diagrama Entitate-Relație a bazei de date

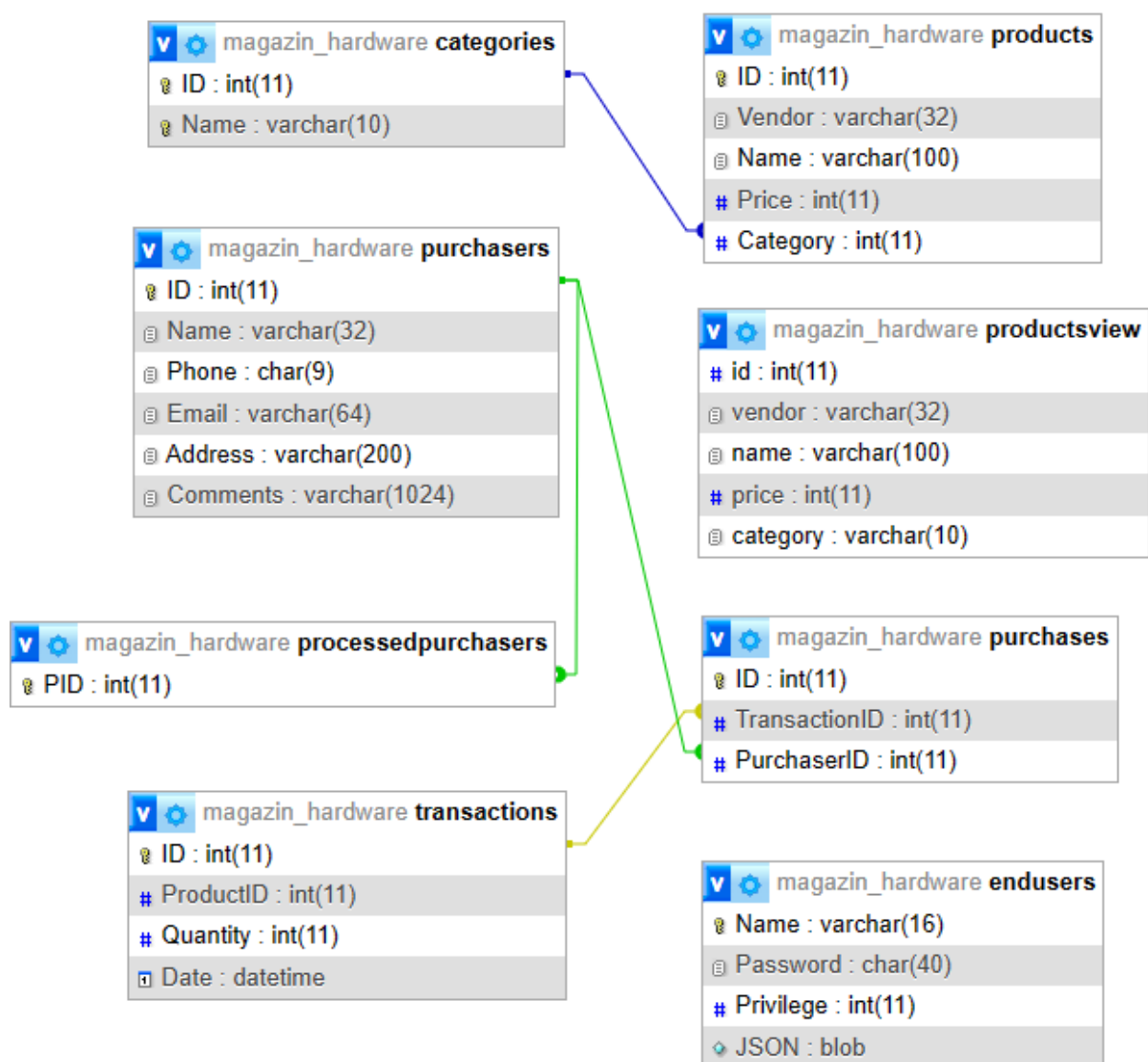


Figura 1

Scenariu SQL

```
CREATE DATABASE magazin_hardware;  
USE magazin_hardware;
```

```
CREATE TABLE Categories (  
    ID INT PRIMARY KEY,  
    Name VARCHAR(10) UNIQUE NOT NULL  
);
```

```
INSERT INTO Categories VALUES  
    (0, 'cpu'),  
    (1, 'gpu'),  
    (2, 'ram'),  
    (3, 'hdd'),  
    (4, 'mb'),  
    (5, 'mouse'),  
    (6, 'kb'),  
    (7, 'cam'),  
    (8, 'pc'),  
    (9, 'laptop');
```

```
CREATE TABLE Products (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    Vendor VARCHAR(32) NOT NULL,  
    Name VARCHAR(100) NOT NULL,  
    Price INT NOT NULL,  
    Category INT NOT NULL,  
    FOREIGN KEY (Category) REFERENCES Categories(ID)  
);
```

```
CREATE TABLE Transactions (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    ProductID INT,  
    Quantity INT NOT NULL,  
    Date DATETIME NOT NULL,  
    FOREIGN KEY (ProductID) REFERENCES Products(ID) ON DELETE CASCADE  
);
```

```
CREATE TABLE Purchasers (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    Name VARCHAR(32) NOT NULL,  
    Phone CHAR(9) NOT NULL,  
    Email VARCHAR(64) NOT NULL,  
    Address VARCHAR(200) NOT NULL,  
    Comments VARCHAR(1024)  
);
```

```
CREATE TABLE ProcessedPurchasers (  
    PID INT PRIMARY KEY,  
    FOREIGN KEY (PID) REFERENCES Purchasers(ID)  
);
```

```
CREATE TABLE Purchases (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    TransactionID INT,  
    PurchaserID INT,  
    FOREIGN KEY (TransactionID) REFERENCES Transactions(ID) ON DELETE CASCADE,  
    FOREIGN KEY (PurchaserID) REFERENCES Purchasers(ID) ON DELETE CASCADE
```

```

);

CREATE TABLE EndUsers (
    Name VARCHAR(16) PRIMARY KEY,
    Password CHAR(40) NOT NULL,
    Privilege INT NOT NULL,
    JSON BLOB
);

DELIMITER $$
CREATE PROCEDURE add_update_enduser(user VARCHAR(32), pwd CHAR(40), priv INT)
BEGIN
    DECLARE n INT;
    SELECT COUNT(*) INTO n FROM EndUsers WHERE Name = user;

    IF n > 0 THEN
        UPDATE EndUsers SET Password = COALESCE(pwd, Password), Privilege =
priv WHERE Name = user;
    ELSEIF pwd IS NOT NULL THEN
        INSERT INTO EndUsers VALUES (user, pwd, priv, NULL);
    END IF;
END$$
DELIMITER ;

CREATE VIEW ProductsView AS
    SELECT A.ID id, Vendor vendor, A.Name name, Price price, B.Name category
    FROM Products A INNER JOIN Categories B ON Category = B.ID
    ORDER BY id;






```

Proiectarea arhitecturii site-ului

Structura fișierelor

Acest site este structurat conform listei arborescente:

- Site
 - css
 - ✓ Include foile de stil pentru a stiliza site-ul:
 - about.css
 - Stilizarea paginii despre autor.
 - style.css
 - Stilizarea generală a elementelor HTML.
 - cart.css
 - Stilizarea meniului ce reprezintă coșul de cumpărături
 - carousel.css
 - Stilizarea „caruselului” ce reprezintă panoul produselor recomandate (se auto-actualizează).
 - products.css
 - Stilizarea listei de produse.
 - products_page.css
 - Stilizarea paginii unde este plasată lista cu produsele.
 - checkout.css
 - Stilizarea formularului ce permite finalizarea coșului de cumpărături.
 - contact.css
 - Stilizarea paginii cu datele de contact a autorului.
 - js
 - ✓ Include scripturile ce permite site-ului să interacționeze cu utilizatorul:
 - cart.js
 - Permite adăugarea sau ștergerea produselor din coșul de cumpărături.
 - carousel.js
 - Popularea „caruselurilor” cu datele cerute din attributele lor.
Fiecare „carusel” are un număr specificat de slide-uri.
 - products.js

- Popularea paginii cu produse conform interogărilor date de client. Răspunsul la interogare se obține de la server sub formă JSON.
- checkout.js
 - Ameliorează utilizatorului ca să-și analizeze coșul de cumpărături.
- realtimeid.js
 - Se completează automat câmpurile formularului de actualizare a unui produs, confirm ID-ului selectat. Utilizat de pagina /panel/products.php.
- img
 - ✓ Imagini:
 - about
 - Utilizate de pagina about.php:
 -  ...
 - carousel
 - Utilizate de „carouselul” de pe pagina principală:
 -  ...
 - products
 - În acest folder se memorizează imaginile .jpg pentru fiecare produs:
 -  cpu (categoria unui produs)
 - 123.jpg (123 indică ID-ul unui produs)
 -  gpu
 -  ...
 - logo.png
 - Insigna „Magazin Hardware”, ilustrată în figura 1.
 - favicon.ico
 - Pictograma site-ului.
 - index.php
 - Răspunde cu un conținut de tip „image/jpeg” după câmpurile de intrare GET, care ușurează procesul de obținere a imaginii unui produs.

- Răspunde cu o listă JSON, lista imaginilor care vor fi utilizate de „caruselul” de pe pagina principală.
- frag (componentele cerute de paginile PHP)
 - head.php
 - Tag-ul <head> completat cu titlul, descripție, căile stilurilor și a scripturilor date ca „parametrii”. Începe tag-urile <html> și <body>.
 - header.php
 - Tag-ul <header> completat cu: logo-ul proiectului și coșului de cumpărături. Coșul de cumpărător poate fi imutabil în dependență după variabila logică \$readOnly.
 - nav.html
 - Tag-ul <nav> ce reprezintă meniul de navigare.
 - credentials.php
 - Definește constantele db_host, db_username, db_password și db_name cu valorile corespunzătoare: denumirea gazdei unde se află baza de date „Magazin Hardware”, denumirea utilizatorului ce are acces la această bază de date, parola, și denumirea bazei de date. Este declarată funcția db_connect(), ce returnează o conexiune (instanța mysqli) la baza de date.
 - connect.php
 - Declară o variabilă globală \$conn, cu valoarea returnată de funcția db_connect().
 - footer.html
 - Tag-ul <footer> completat (antetul paginii). Încheie tag-urile <body> și <html>.
 - global.php
 - Include componentele head.php, header.php și nav.html.
 - fetchProducts.php
 - Răspunde cu o listă JSON, unde se stochează rezultatul unei interogări compuse din câmpurile de intrare GET.
- panel
 - css
 - products.css – stilizează pagina products.php.

- index.php:
 - Meniul de gestiune:
 - Lista utilizatorilor;
 - Lista produselor;
 - Lista tranzacțiilor;
 - Recuperarea coșului de cumpărături;
 - Salvarea coșului de cumpărături.
 - Dacă utilizatorul este neautentificat, atunci se afișează pagina de autentificare.
- products.php
 - Ștergerea sau modificarea unui produs;
 - Restricționată pentru utilizatorii cu rolul de vânzător.
- signup.php
 - Crearea unui utilizator nou;
- transactions.php
 - Vizualizarea și finalizarea tranzacțiilor;
 - Restricționată pentru utilizatorii cu rolul de secretar.
- users.php
 - Adăugarea, modificare și ștergerea utilizatorilor;
 - Restricționată pentru utilizatorii cu rolul de administrator.
- index.php
 - Pagina principală a site-ului, care include „carouselul”.
- products.php
 - Lista produselor. Conținutul paginii este completat utilizând JavaScript, prin intermediul fetchProducts.php.
- contact.php
 - Datele de contact despre autor.
- about.php
 - Informații și curiozități despre proiect.
- checkout.php
 - Finalizarea coșului de cumpărături. Datele completate în formular, împreună cu conținutul coșului de cumpărături, devin introduse în baza de date.
 - Validarea datelor din formular are loc pe client și pe server.

- La introducerea lor cu succes în baza de date, se afișează o notificare corespunzătoare.



Figura 2. Insigna „Magazin Hardware”

Machetare, stilizare și scriptare

Pagina principală



Figura 3. Pagina principală (1920x1080)

Pagina principală cuprinde un „carusel” sau slideshow ce reprezintă slide-urile cu produsele pe panou (recomandate). În machetare, „caruselul” este un element div gol cu attributele: `class="carousel-container" data-idx="0" data-auto="right"`. Attributele prefixate cu `data` se comportă ca datele de intrare ce determină starea curentă a slideshow-ului.

În JavaScript, fiecare element cu clasa `carousel-container` se completează cu elementele div ce au clasele: `carousel`, `carousel-bullets`, `left carousel-arrow` și `right carousel-arrow`:

- `carousel` – conține elementele `img` ca slide-uri, numărul de slide-uri, slide-ul curent și direcția în care pagina se deplasează automat sunt determinate de attributele `idx` și `auto`, toate prefixate cu `data`. Atributul `src` pentru fiecare imagine este completat prin expedierea parametrilor scriptului `/img/index.php`, cu parametrul `GET`, denumit `carousel`.
- `carousel-bullets` – conține elementele `span` cu clasa `carousel-bullet` ce permit selectarea în mod manual poziția slide-ului.
- `carousel-arrow` – asemănător cu cel precedent, dar permite poziționarea pe dreapta sau stânga.

La completarea caruselului, se apelează, în JavaScript, un lambda asincron (de fundal) ce schimbă slide-ul automat cu un pas, la fiecare 7.5 de secunde.

În machetare, clasa `carousel-container` este poziționată relativ cu raportul de aspect 2 / 1, ce înseamnă că înălțimea sa va fi jumătate din lățime și ascunde conținuturile ce se revarsă din aria sa. Clasa `carousel` are înălțimea de 100% (din părinte), plasarea forțată pe linie (`white-space: nowrap`) și transformarea pe axa X unde valoarea este reprezentată de variabila CSS `--idx` care se înmulțește cu -100%.

Pagina cu produse



Figura 4. Pagina cu produse (1920x1080)

Înainte să fie încărcată pagina, se verifică (are loc validarea) parametrii GET, prin intermediul scriptului /frag/fetchProducts.php, cu scopul de a filtra produsele afișate pe ecran:

- **id** – selectează un produs cu ID-ul corespunzător;
- **search** – filtrează lista produselor după modelul dat în rubrica de căutare. Se implică expresii regulate pentru a asigura că toate cuvintele introduse de la utilizator sunt incluse în modelele produselor;
- **vendor** – filtrează după producător;
- **view** – filtrează după tip de produs;
- **limit** – afișează un număr dat de produse pe pagină;
- **page** – pagina curentă a listei;
- **sort** – sortarea ascendentă sau descendentă după criteriu (istoric sau după preț).

Componentul/scriptul, fetchProducts.php, generează vederea interogării SQL alcătuit conform parametrilor GET expediate de client (în cazul nostru – scriptul JavaScript realizează expedierea).

În machetare, am utilizat elementul template cu id `itemEntryTemplate`, ce reprezintă un produs de pe pagină sub formă de șablon. Meniul „căutarea avansată” este un formular ce permite expedierea parametrilor la pagină, și este în elementul `aside`, în interiorul `body`.

Sortarea: ▼

După furnizor: ▼


După model:

Itemi/pagină: ☒ 16 ☐ 32 ☐ 64

Calculatoare



Dell Vostro 3471

 Adaugă în coș - 7632 lei



Dell Vostro 3681

 Adaugă în coș - 8597 lei

Figura 5

Pagina de finalizare unei comenzii

Produs	Cantitate	Preț
1. AMD Ryzen 7 3700X	1	1091.00 lei
2. Mother B450M	1	1294.00 lei
Total:	2	8334.00 lei

Plasează comanda

Completați formularul

Comanda vă costă: **8334.00 lei**

Date personale:

Prenume:

Număr de contact:

Posta electronică:

Date de livrare:

Adresă de livrare:

Comentariu:

☐ Am verificat conținutul comenzii și doresc să plasez comanda

Expediește

Imaginile produselor

AMD Ryzen 7 3700X

Mother B450M

Despre „Magazin Hardware”

Conținutul bazei de date
Măsu 5020
Lucrarea s-a realizat în

Contact
Site-ul personal
Blog-ul personal

Despre autor

Copyright © Cristian Cretian 2020-2023

Figura 6. Pagina de finalizare unei comenzii (1920x1080)

În stilizare, orice element din formular sunt marcate cu roșu dacă nu-și satisfac cerințele de validare.

Când lățimea ecranului este mai mare decât 1207px, tabela cu produsele în coș este poziționată în mod fix, și această permite la o ușurare ce prevede verificarea coșului de cumpărături.

Datele din formular sunt validate de pe partea clientului și de pe server, utilizând expresiile regulate și filtre (pentru câmpul email). Câmpurile invalide devin colorate cu roșu.

În cazul când utilizatorul ocolește forțat validarea locală, sau dorește să expedieze formularul completat incorect, serverul refuză să-l proceseze mai departe, obligând utilizatorului să completeze încă o dată formularul.

Prenume:

Număr de contact:

Posta electronică:

Figura 7. Câmpurile invalide marcate

Succes

Comanda a fost preluată cu succes. Puteți reveni la pagina principală realizând un click [aici](#).

Figura 8. Comanda finalizată

La finalizarea comenzii, datele din formular se introduc în baza de date: tabelele Purchasers, Transactions și Purchases.

Logica validării datelor urmează în felul următor:

```
define('nameRegexI', '/^((?![_\d])\w){1,32}$/u'); // cel mult 32 litere (inclusiv diacritice)
define('phoneRegex', '0[0-9]{8}'); // 9 cifre, unde primul este 0

if ($_SERVER['REQUEST_METHOD'] == 'POST' &&
    preg_match(nameRegexI, @$_POST['prenume']) &&
    preg_match('/^' . phoneRegex . '$/', @$_POST['tel']) &&
    filter_var(@$_POST['email'], FILTER_VALIDATE_EMAIL) !== false &&
    !empty(@$_POST['cart'])) { // nu are coșul de cumpărături gol
// Se verifică dacă câmpurile sunt introduse corect
    require 'frag/connect.php'; // conexiune la BD

    try {
        $conn->begin_transaction(); // se începe unitatea de lucru
indivizibilă

        $stmt = $conn->prepare('INSERT INTO Purchasers VALUES (NULL, ?, ?,
?, ?, ?)');
        $stmt->execute([$_POST['prenume'], $_POST['tel'], $_POST['email'],
            htmlspecialchars($_POST['adr']),
            htmlspecialchars($_POST['comment'])]);
// Utilizăm interogările preparate ca să „sterilizăm” datele de intrare
        $pID = $conn->insert_id; // ID-ul cumpărătorului recent introdus

        $stmt = $conn->prepare('INSERT INTO Transactions VALUES (NULL, ?, ?,
SYSDATE())');
        $stmt1 = $conn->prepare('INSERT INTO Purchases VALUES (NULL, ?,
?)');

        foreach (explode(',', $_POST['cart']) as $itemQ) {
            $arr = explode(' ', $itemQ);

            $stmt->execute([$arr[1], $arr[0]]);

            $insertId = $conn->insert_id;
            $stmt1->execute([$insertId, $pID]);
        }

        $conn->commit(); // Persistarea schimbărilor asupra BD
        header('Location: ?success');
    }
    catch (mysqli_sql_exception) {
        $conn->rollback(); // A avut loc o eroare, se anulează schimbările
        header('Refresh: 0');
    }

    exit;
}
```

Designul în general

Pentru a face un site responsiv, am specificat că elementul `body` să fie de tip `flex`, cu direcția pe coloană și să nu aibă nici o margină. Fontul este de sistem (`system-ui`) cu dimensiunea de 12 puncte. Fiecare pagină are următoarea structură:

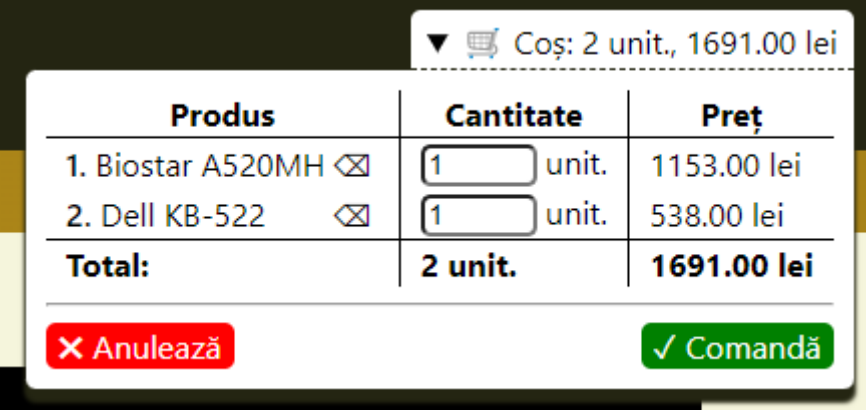
- `body`
 - `header` – imboldul companiei și elementul `details` ce conține coșul de cumpărături.
 - `nav` – conține meniuri de navigare. În pagina cu produse, este încă un element ce permite afișarea sau ascunderea opțiunii de căutare avansată.
 - `aside` – prezent în pagina cu produse. Conține un formular ce permite filtrarea produselor după cerințele clientului.
 - `main` – corpul principal a paginii, unicul element `flex` cu multiplicatorul de creștere 1.
 - `article` - articol
 - `section` – secțiune, unde sunt plasate conținuturile
 - ...
 - ...
 - `footer` – subsolul cu ancore suplimentare de navigare și dreptul de autor.

Pentru a afișa submeniul când cursorul este deasupra unui item din bara de navigare, a fost implicată pseudo-clasa `hover` și `focus-within` pentru tasta `Tab`. În stilizare, orice submeniu, inclusiv coșul de cumpărături, sunt reprezentate de un element `div` cu clasa `dropdown`.



Figura 9. Submeniu

Coșul de cumpărături, descendentul elementului details cu id-ul cartView, permite excluderea individuală a unei produs sau chiar anularea întregii comenzi. Butonul de comandare ne duce la pagina de finalizare a comenzii. Coșul de asemenea suportă și validarea datelor în privința schimbării cantității de produs. Prin intermediul elementului template cu id-ul cartBodyRowTemplate, se ușurează scriptarea ce prevede adăugarea produselor.





▼ Coș: 2 unit., 1691.00 lei		
Produs	Cantitate	Preț
1. Biostar A520MH 	<input type="text" value="1"/> unit.	1153.00 lei
2. Dell KB-522 	<input type="text" value="1"/> unit.	538.00 lei
Total:	2 unit.	1691.00 lei
<div> ✖ Anulează ✓ Comandă </div>		

Figura 10. Coșul de cumpărături

Elementul părinte details are atributul data-has-items când coșul de cumpărături nu este gol. Acest atribut se exclude când nu este nici un produs în coș. Determinarea conținutului pentru coșul plin sau gol este implementat doar în stilizare, cu ajutorul selectorului de attribute.

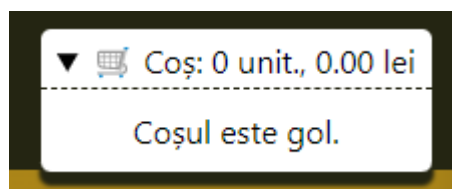


Figura 11. Coșul gol

Compartimentul „Contul meu”

Pagina „Sign-up”

Dacă utilizatorul accesează compartimentul „Contul meu” și nu este autentificat, se afișează această pagină. Câmpurile din acest formular sunt validate în felul următor: denumirea contului să fie unică și că parola să conțină cel puțin 8 simboluri.

Figura 12

Captcha este reprezentat ca al patrulea câmp din formularul de înregistrare (figura 14).

Contul nou se introduce în baza de date, cu privilegiul de client.

Logica validării paginii urmează astfel:

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {  
    // Verifică dacă câmpurile sunt completate  
    if (!isset($_POST['user'], $_POST['password'], $_POST['captcha']) ||  
        strlen($_POST['user']) == 0 || strlen($_POST['password']) == 0) {  
        header('Location: ?info=0');  
        exit;  
    }  
  
    // Verifică dacă parola este confirmată corect  
    if ($_POST['password'] !== @$_POST['password2']) {  
        header('Location: ?info=1');  
        exit;  
    }  
  
    // Verifică dacă valoarea p/u CAPTCHA este introdusă corect  
    if ($_POST['captcha'] !== $_SESSION['captcha']->Vendor) {  
        header('Location: ?info=2');  
        exit;  
    }  
}
```

```

unset($_SESSION['captcha']);

// Lungimea parolei - cel puțin 8 simboluri
if (!preg_match('/.{8}/', $_POST['password'])) {
    header('Location: ?info=4');
    exit;
}

require '../frag/connect.php';

// Verifică dacă username-ul nu există
if ($conn->query(sprintf('SELECT NULL FROM EndUsers WHERE Name = \'%s\'',
$conn->real_escape_string($_POST['user'])))->num_rows > 0) {
    header('Location: ?info=3');
    exit;
}

try {
    // Introducerea datelor despre utilizatorul nou în BD
    // Funcția hash este SHA1
    $conn->prepare('INSERT INTO EndUsers VALUES (?, SHA1(?), -1,
NULL)')->execute([$_POST['user'], salted($_POST['password'], $_POST['user'])]);
}
catch (mysqli_sql_exception) {
    // În caz de eroare
    header('Location: ?info=5');
    exit;
}

// Este autentificat
$_SESSION['user'] = $_POST['user'];
$_SESSION['privilege'] = -1;
header('Location: .');
exit;
}

```

Funcția ce adaugă sare pentru parola este definită în /frag/credentials.php:

```
function salted($a, $b) { return $a . md5(strrev($b)); }
```

Pagina principală

Pagina principală afișează utilizatorului înregistrat, un meniu de administrare (figura 16). Utilizatorii de orice privilegiu pot să-și salveze/recupereze coșul de cumpărături.

Dacă utilizatorul nu este autentificat, atunci se afișează următorul dialog:

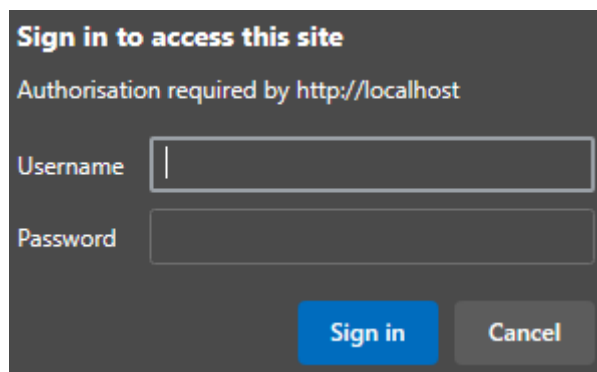


Figura 13. Dialog

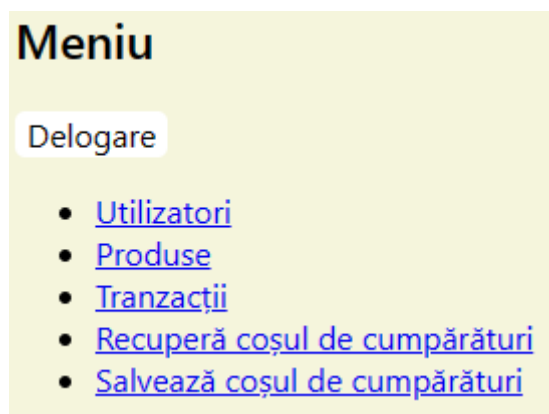


Figura 14. Meniul afișat în cazul administratorului

Aducem la cunoștință că toate paginile de administrare sunt protejate împotriva ocolirii autentificării, utilizând sesiunile.

Logica validării paginii urmează astfel:

```
session_start();

// Deautentificare dacă există parametrul GET logoff
if (isset($_GET['logoff'])) {
    session_destroy();
    header('Location: /');
    exit;
}

// Dacă este neautentificat
if (!isset($_SESSION['user'])) {
    // Redirecționare spre pagina de înregistrare dacă nu este explicit login
    if (!isset($_GET['login'])) {
        header('Location: signup.php');
        exit;
    }
}
```

```

    }

    // Verifică dacă utilizatorul a expediat username și parola
    if (isset($_SERVER['PHP_AUTH_USER'], $_SERVER['PHP_AUTH_PW'])) {
        require '../frag/connect.php';
        $stmt = $conn->prepare('SELECT Privilege FROM EndUsers WHERE Name =
? AND Password = SHA1(?)');
        $stmt->execute([$_SERVER['PHP_AUTH_USER'],
salted($_SERVER['PHP_AUTH_PW'], $_SERVER['PHP_AUTH_USER'])]);
        $row = $stmt->get_result()->fetch_row();

        if (isset($row)) {
            $_SESSION['user'] = $_SERVER['PHP_AUTH_USER'];
            $_SESSION['privilege'] = $row[0];
            http_response_code(401);
            header('Refresh: 0');
            echo ' ';
            exit;
        }
    }

    // Afișarea dialogului
    header('WWW-Authenticate: basic realm="Acces intern"');
    header('Refresh: 0;url=signup.php');
    echo ' ';
    exit;
}

// Obținerea coșului de cumpărături salvat (realizat de un script JavaScript, prin
Fetch API)
if (isset($_GET['get'])) {
    require '../frag/connect.php';
    header('Content-Type: application/json');
    echo $conn->query(sprintf('SELECT JSON FROM EndUsers WHERE Name = \'%s\'',
$conn->real_escape_string($_SESSION['user'])))->fetch_row()[0];
    exit;
}

// Actualizarea coșului de cumpărături salvat (realizat de un script JavaScript,
prin Fetch API)
if ($_SERVER['REQUEST_METHOD'] == 'POST' && $_SERVER['CONTENT_TYPE'] ==
'application/json') {
    require '../frag/connect.php';
    $json = file_get_contents('php://input');
    $conn->prepare('UPDATE EndUsers SET JSON = ? WHERE Name = ?')-
>execute([$json == '{}' ? null : $json, $_SESSION['user']]);
    exit;
}

```

Pagina „Utilizatori”

Această pagină permite administratorului să adauge/modifice/șteargă un cont. Administratorul poate să seteze unui utilizator, unul din cele 4 privilegii: client, secretar, vânzător și administrator. Administratorul nu poate să-și modifice privilegiul.

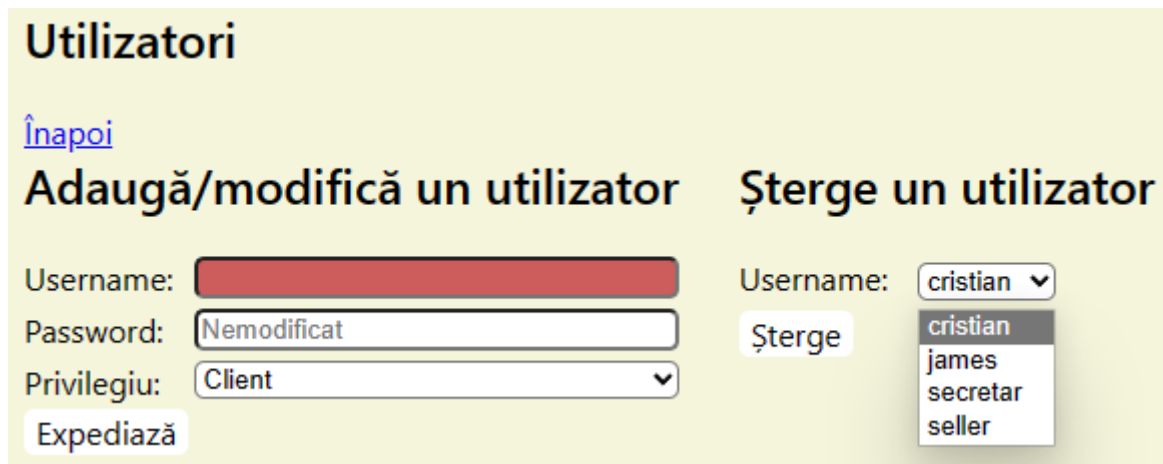


Figura 15

Dacă utilizatorul curent nu este un administrator, atunci poate numai să-și schimbe parola și să-și șteargă contul. În acest caz, câmpul „Username” este completat automat, nemodificabil și nu este asociat formularului.

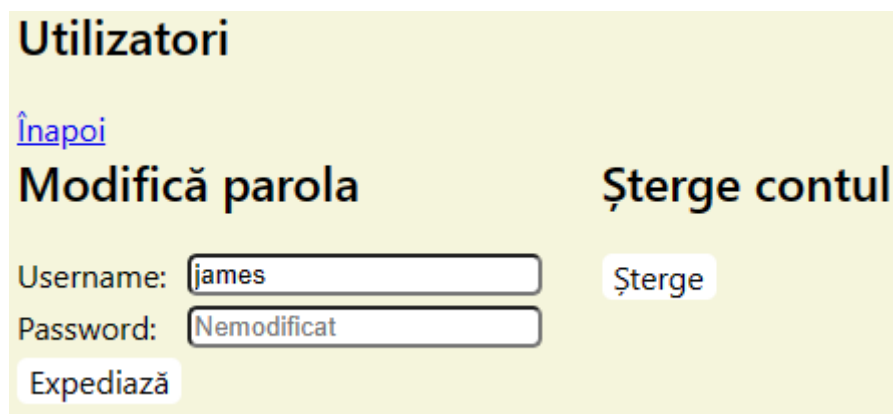


Figura 16

Logica formularului urmează în felul următor:

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {  
    // Adaugă/modifică cont  
    if (isset($_POST['createupdate'])) {  
        // Este administrator  
        if (isAdmin) {  
            // Sunt completate câmpurile + administratorul nu-și poate  
            degrada privilegiul  
            if (@strlen($_POST['user']) > 0) {  
                require '../frag/connect.php';  
            }  
        }  
    }  
}
```



```

        $conn->prepare('CALL add_update_enduser(?, SHA1(?),
?)')->execute([$POST['user'], strlen($POST['password']) > 0 ?
salted($POST['password'], $POST['user']) : null, $_SESSION['user'] !==
$POST['user'] ? $POST['privilege'] : $_SESSION['privilege']]);
// Dacă parola de intrare nu e introdusă, atunci nu se modifică parola contului
    }
}
else {
    require '../frag/connect.php';
    // Schimbă parola pentru contul curent
    $conn->prepare('UPDATE EndUsers SET Password =
COALESCE(SHA1(?), Password) WHERE Name = ?')->execute([
        strlen($POST['password']) > 0 ?
salted($POST['password'], $_SESSION['user']) : null, $_SESSION['user']]);
}
}
// Șterge un cont
elseif (isset($POST['delete'])) {
    if (isAdmin) {
        $delSelf = $POST['user'] === $_SESSION['user'];
        // Se verifică dacă se șterge contul curent
    }
    else {
        $POST['user'] = $_SESSION['user'];
        $delSelf = true;
        // Non-administratorii numai pe sine pot șterge
    }
}

require '../frag/connect.php';
// Se execută ștergerea contului, dacă nu este unicul
if ($conn->query('SELECT COUNT(*) FROM EndUsers')->fetch_row()[0] >
1) {
    $conn->prepare('DELETE FROM EndUsers WHERE Name = ?')->
execute([$POST['user']]);

    if ($delSelf) {
        header('Location: .?logoff');
        exit;
    }
}

header('Refresh: 0');
exit;
}

```

Pagina „Produse”

Aceasta pagină permite vânzătorului să adauge/modifice/șteargă un produs. Sub-compartimentul „Produsele reclamate”, care este lista imaginilor pentru „carousel”, este vizibil doar administratorului.

The screenshot shows a web interface for managing products. It has a yellow background and a title 'Produse' at the top left. Below the title is a link 'Înapoi'. The interface is divided into three main sections:

- Adaugă/modifică un produs:** Contains form fields for 'Cod serial:', 'Furnizor:', 'Model:', 'Pret:', and 'Categorie:' (with a dropdown menu showing 'cam'). There is also an 'Imagine:' section with a 'Choose File' button and the text 'No file chosen'. At the bottom of this section is a button 'Expediază'.
- Șterge un produs:** Contains a 'Cod serial:' field and a 'Șterge' button.
- Produsele reclamate:** A table with two columns: 'Denumire' and 'Acțiune'. It lists four items with IDs 1, 2, 3, and 4. Each item has a 'Șterge' button in the 'Acțiune' column. Below the table is a 'Choose File' button and the text 'No file chosen'. At the bottom right of this section is a button 'Adaugă'.

Figura 17

Logica paginii:

```
require '../frag/connect.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Se sterilizează fiecare parametru
    foreach ($_POST as &$field) {
        $field = $conn->real_escape_string(htmlspecialchars($field));
    }

    // Adaugă/modifică un produs
    if (isset($_POST['insertupdate'])) {
        // Dacă ID-ul e specificat, atunci se modifică produsul
        if (@is_numeric($_POST['id'])) {
            $o = $conn->query('SELECT * FROM ProductsView WHERE ID = ' .
$_POST['id'])->fetch_object();
            // Se execută dacă există
            if (isset($o)) {
                // Se acceptă doar imaginile de tip JPEG
                if ($_FILES['image']['type'] == 'image/jpeg') {
                    // Se mută fișierul din directorul temporar în mapa sa
                    move_uploaded_file($_FILES['image']['tmp_name'],
sprintf('../img/products/%s/%s.jpg', $o->category, $o->id));
                }
                // Dacă nu s-a încărcat imaginea, atunci se actualizează
                // locația imaginii existente
                else {
                    rename(
                        sprintf('../img/products/%s/%s.jpg', $o-
>category, $o->id),
                        sprintf('../img/products/%s/%s.jpg',
                            $conn->query('SELECT Name FROM Categories
WHERE ID = ' . $_POST['category'])->fetch_row()[0],
                            $o->id)
                    );
                }
            }
        }
    }
}
```

```

    }

    // Se actualizează datele produsului
    $conn->prepare('UPDATE Products SET Vendor = ?, Name = ?,
Price = ?, Category = ? WHERE ID = ' . $o->id->execute([
        $_POST['vendor'], $_POST['model'], $_POST['price'],
$_POST['category']
    ]));
    }
    }
    // Dacă ID-ul nu e specificat, atunci se verifică dacă o imagine a
    fost încărcat
    elseif ($_FILES['image']['type'] == 'image/jpeg') {
        // Se inserează produsul
        $conn->prepare('INSERT INTO Products VALUES (NULL, ?, ?, ?,
?)')->execute([
            $_POST['vendor'], $_POST['model'], $_POST['price'],
$_POST['category']]);

        // Se mută fișierul încărcat...
        move_uploaded_file($_FILES['image']['tmp_name'],
sprintf('../img/products/%2$s/%1$s.jpg',
            $conn->insert_id, // ID-ul produsului inserat
            $conn->query('SELECT category FROM ProductsView WHERE
ID = ' . $conn->insert_id)->fetch_row()[0])
        );
    }
}
// Dacă se șterge produsul
elseif (isset($_POST['delete'])) {
    // Se obține categorie produsului ce va fi șters
    $o = $conn->query('SELECT * FROM ProductsView WHERE ID = ' .
+$_POST['delete']->fetch_object());
    // Se șterge imaginea produsului
    unlink(sprintf('../img/products/%s/%s.jpg', $o->category, $o->id));
    // Se execută ștergerea produsului
    $conn->query('DELETE FROM Products WHERE ID = ' .
+$_POST['delete']);
}
// Se verifică dacă utilizatorul este administrator
elseif (isAdmin) {
    // Se adaugă o imagine la carusel
    if (isset($_POST['carouselAdd'])) {
        if ($_FILES['image']['type'] == 'image/jpeg') {
            move_uploaded_file($_FILES['image']['tmp_name'],
'../img/carousel/' . basename($_FILES['image']['name']));
        }
    }
    // Se șterge o imagine din carusel
    elseif (isset($_POST['carouselDel'])) {
        unlink(sprintf('../img/carousel/%s.jpg',
basename($_POST['carouselDel'])));
    }
}

header('Refresh: 0');
exit;
}

```

Pagina „Tranzacții”

Această pagină permite secretarului să vizualizeze și să confirme tranzacțiile. Pentru client este afișat lista produselor achiziționate.

Tranzacții

Meniu

ID	Prenume	Telefon	Email	Adresa de livrare	Comentarii	
<input checked="" type="checkbox"/>	1	Cristian	076757900	cemirtan.cristian@outlook.com	str. Paris	Sa fie testate.
Data		Produs		Cantitate	Pret/unitate	Total (MDL)
2023-04-19 03:29:09		V50s-07IMB		1	6798	6798
2023-04-19 03:29:09		Core i5 9400F		1	3600	3600
2023-04-19 03:29:09		4GB DDR3-1600MHz		2	500	1000
Total: 11398						
<input type="checkbox"/>	2	Paul	075277743	paul@walker.co.uk	Red Light District 32nd Way	nimic
Data		Produs		Cantitate	Pret/unitate	Total (MDL)
2023-04-19 20:45:44		A68N-5600E		1	1533	1533
2023-04-19 20:45:44		Celeron G5900		1	1164	1164
Total: 2697						
<input checked="" type="checkbox"/>	3	Chril	067889315	kirea251@mail.md	str. Matei Basarab 14/1	Sa fie calitative.
Data		Produs		Cantitate	Pret/unitate	Total (MDL)
2023-04-20 02:59:30		Athlon II X2 240		1	800	800
2023-04-20 02:59:30		2.5" SSD FURY 3D		1	870	870
2023-04-20 02:59:30		B450MH		1	1284	1284
2023-04-20 02:59:30		M570 Trackball		1	1428	1428
Total: 4382						
<input type="checkbox"/>	4	Cristian	076757981	cemirtan.cristian@outlook.com	str. Paris	nimic
Data		Produs		Cantitate	Pret/unitate	Total (MDL)
2023-04-20 03:08:51		Vostro 3681		1	8597	8597
Total: 8597						
<input type="checkbox"/>	5	Diana	067813843	checking.in@mail.ru	Broadbent 1234	Didn't see it working so fast.
Data		Produs		Cantitate	Pret/unitate	Total (MDL)
2023-04-21 00:19:36		V50s-07IMB		1	6798	6798
Total: 6798						
<input type="checkbox"/>	6	Tiberiu	067842698	kok3165@gmail.com	bd. Carpati 12/1	Nu pentru mining.
Data		Produs		Cantitate	Pret/unitate	Total (MDL)
2023-04-29 23:38:20		Ryzen 7 3700X		1	7050	7050
2023-04-29 23:38:21		B450MH		1	1284	1284
Total: 8334						

Procesează

Figura 18

Logica paginii:

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    require '../frag/connect.php';
    // Se inserează în tabela ProcessedPurchasers, rândurile cu ID-urile
    corespunzătoare
    $conn->query('INSERT INTO ProcessedPurchasers VALUES ' . implode(',',
    array_map(function ($x) { return "($x)"; }, $_POST['id'])));
```

```
    header('Refresh: 0');  
    exit;  
}
```

Concluzii

Acest proiect este o recreare în comparație cu edițiile anterioare care erau bazate pe framework-urile Windows Forms și Vaadin. Lucrarea actuală a fost începută în întregime manuală, folosind numai un simplu editor text de a crea fișierele legate de machetare, stilizare și scriptare (de pe client și de pe server).

În machetare, am beneficiat de elementele recent adăugate în HTML5: header, nav, aside, body, section, article, footer, output, details, dialog și template. Aceste elemente contribuie la lizibilitatea accesibilă a machetării și a stilizării.

În scriptarea PHP, am beneficiat de interogările preparate ca să pot realiza tranzacțiile sigure asupra bazei de date „Magazin Hardware”, fiindcă prepararea lor implică sterilizarea implicită a parametrilor de intrare, evitând injecțiile SQL.

În stilizare, am beneficiat și de pseudo-clasele noi: :backdrop, :invalid, :focus-within, :is, și de atributul nou contents a proprietății display. Atributul contents permite ca subelementele a unui element să se comporte cum ar fi copii a superelementului, ceea ce aduce avantaje mari în aplicarea stilurilor de tip flex și grid.

Conținutul coșului de cumpărături este din partea clientului, prin intermediul construcției JS – localStorage. La completarea coșului de cumpărături, în formularul respectiv se adaugă un câmp ascuns ce are valoarea „textuală” a coșului. Valoarea conține un set de perechi cheie-valoare, ID-ul itemului – cantitate. Serverul nu are acces direct la localStorage, de aceea se necesită de creat acel câmp.

Validarea datelor din formular e realizată pe ambele părți. Mesajele din urma validării eșuate sunt afișate din partea clientului, iar din partea serverului – obligă utilizatorul să completeze încă odată, dar corect, formularul.

Bibliografie

1. <https://developer.mozilla.org/en-US/docs/Web/HTML>
2. <https://developer.mozilla.org/en-US/docs/Web/CSS>
3. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
4. <https://www.php.net/manual>
5. Suport de curs „HTML, CSS, JAVASCRIPT”.
<https://moodle.usm.md/course/view.php?id=1699>
6. Suport de curs „Programare Web”.
<https://moodle.usm.md/course/view.php?id=1967>