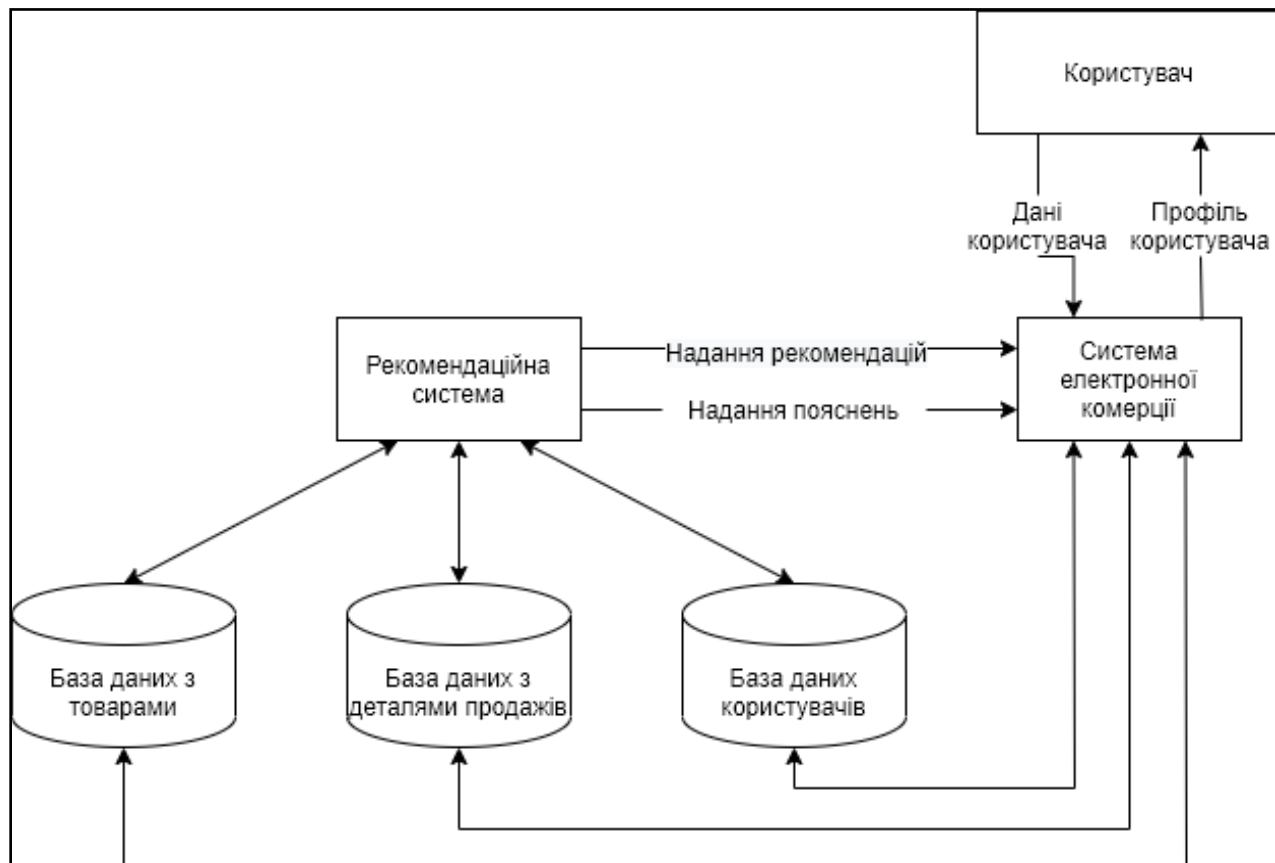


Додаток А
Графічні матеріали

ГЮИК.506100.008 С0

СХЕМА РОБОТИ СИСТЕМИ Е-КОМЕРЦІЇ З РС



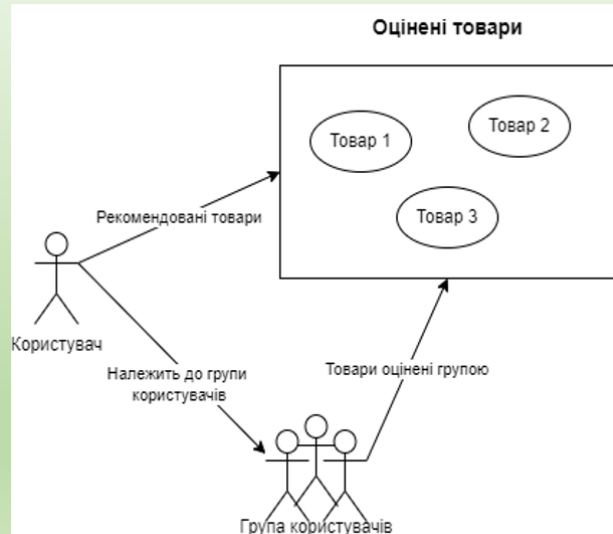
					<i>ГЮИК.506100.008 С0</i>		
					<i>Схема роботи системи е-комерції з РС</i>		
<i>Змін.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Розроб.</i>		<i>Москалюк Ю.О.</i>					
<i>Перевір.</i>		<i>Решетнік В.М.</i>					
<i>Т. Контр.</i>					<i>Лист 1</i>		<i>Листів 1</i>
<i>Реценз.</i>					<i>ХНУРЕ Кафедра СТ</i>		
<i>Н. Контр.</i>							
<i>Затверд.</i>		<i>Гребеннік І.В.</i>					

ГЮИК.506100.008 С0

ДОСЛІДЖУВАНИЙ МЕТОД ФІЛЬТРАЦІЇ

Колаборативна фільтрація (КФ) на основі сусідства (user- based)

- Основна ідея методу — пропонувати нові товари на основі близькості в поведінці подібних користувачів.
- Вважають, що якщо користувачі однаково оцінили деякі об'єкти в минулому, найімовірніше, однаково оцінять інші об'єкти в майбутньому.



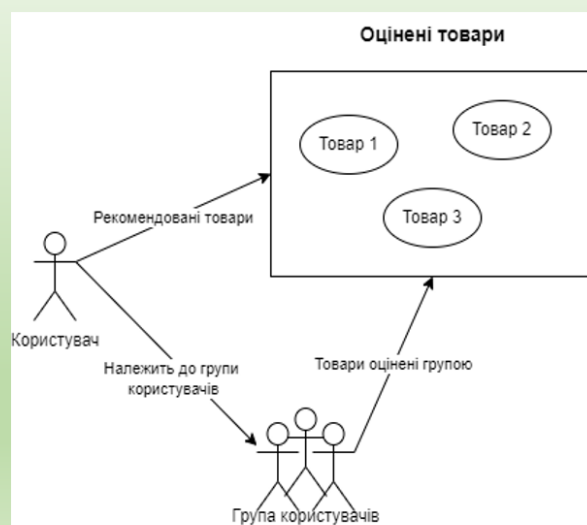
					<i>ГЮИК.506100.008 С0</i>		
					<i>Досліджуваний метод фільтрації</i>		
<i>Змін.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Розроб.</i>		<i>Москалюк Ю.О.</i>					
<i>Перевір.</i>		<i>Решетнік В.М.</i>					
<i>Т. Контр.</i>					<i>Лист 1</i>		<i>Листів 1</i>
<i>Реценз.</i>					<i>ХНУРЕ Кафедра СТ</i>		
<i>Н. Контр.</i>							
<i>Затверд.</i>		<i>Гребеннік І.В.</i>					

ГЮИК.506100.008 С0

ДОСЛІДЖУВАНИЙ МЕТОД ФІЛЬТРАЦІЇ

Колаборативна фільтрація на основі сусідства (user-based)

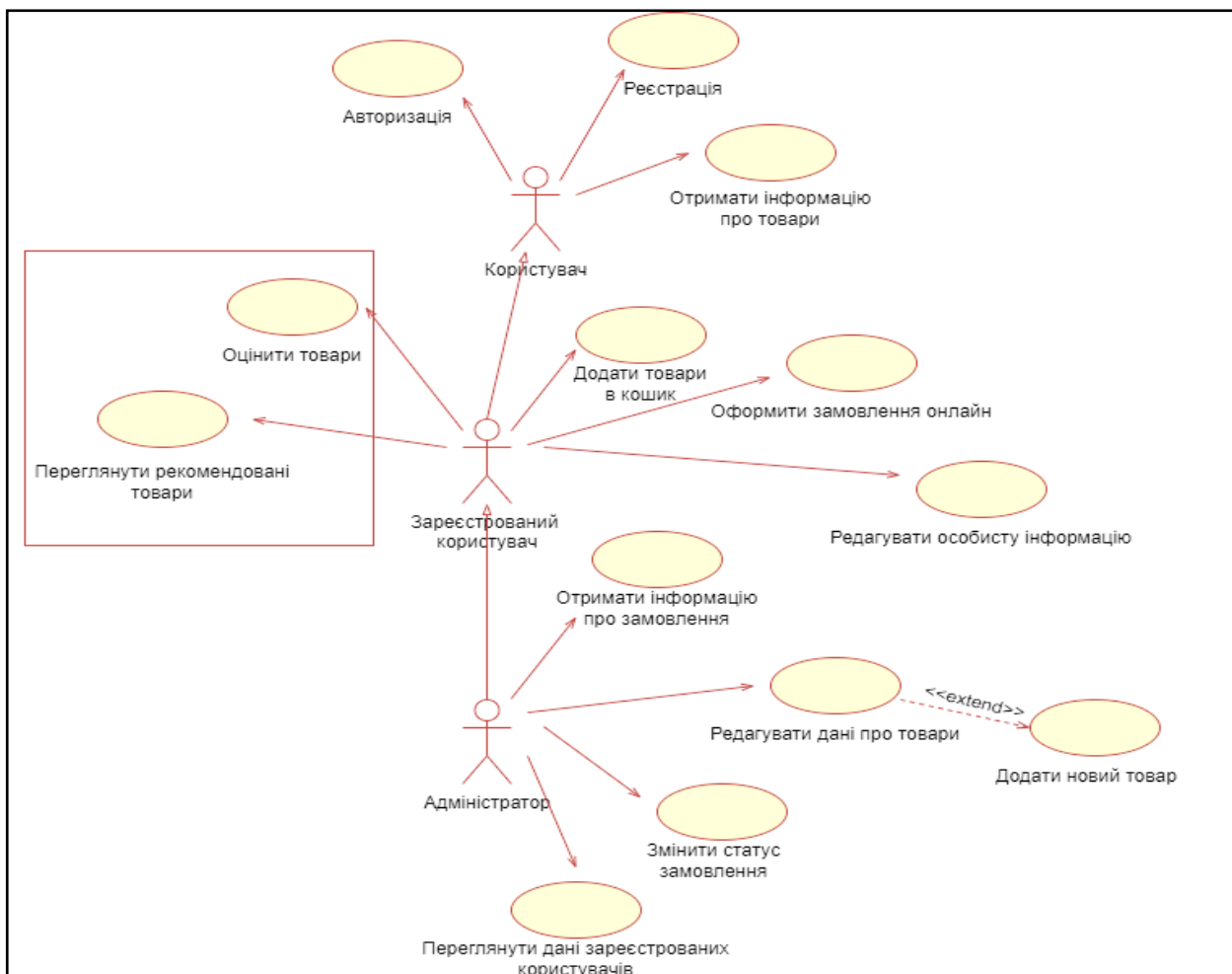
- Основна ідея методу — пропонувати нові товари на основі близькості в поведінці подібних користувачів.



					<i>ГЮИК.506100.008 С0</i>			
					<i>Досліджуваний метод фільтрації</i>	<i>Лім.</i>	<i>Маса</i>	<i>Масштаб</i>
<i>Змін.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>	<i>Москалюк Ю.О.</i>							
<i>Перевір.</i>	<i>Решетнік В.М.</i>							
<i>Т. Контр.</i>						<i>Лист 1</i>		<i>Лист 1</i>
<i>Реценз.</i>						<i>ХНУРЕ</i>		
<i>Н. Контр.</i>					<i>Кафедра СТ</i>			
<i>Затверд.</i>	<i>Гребеннік І.В.</i>							

ГЮИК.506100.008 С10

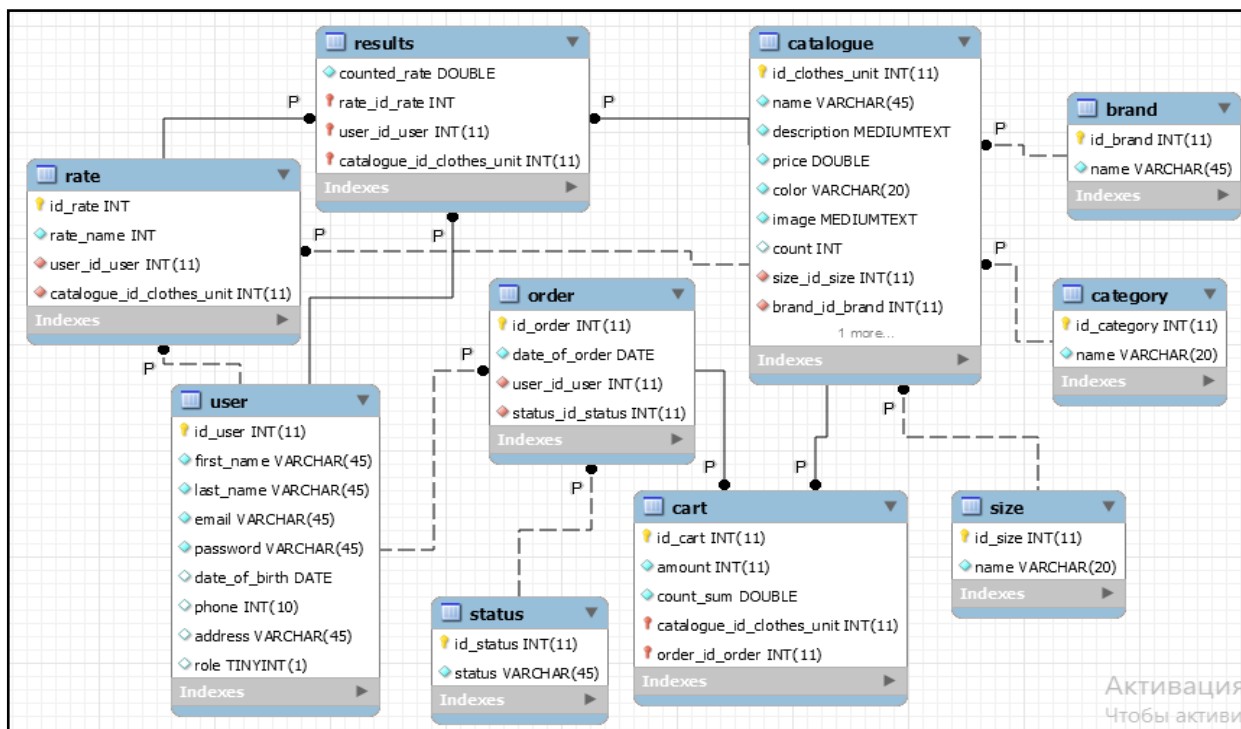
ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



					ГЮИК.506100.008 С10		
					Діаграма варіантів використання		
Змін.	Лист	№ докум.	Підпис	Дата			
Розроб.		Москалюк Ю.О.					
Перевір.		Решетнік В.М.					
Т. Контр.					Лист 1		Лист 1
Реценз.					ХНУРЕ Кафедра СТ		
Н. Контр.							
Затверд.		Гребеннік І.В.					

ГЮИК.506100.008 С10

ФІЗИЧНА МОДЕЛЬ ДАНИХ

Активация
Чтобы активир

					ГЮИК.506100.008 С10			
Змін.	Лист	№ докум.	Підпис	Дата	Фізична модель даних	Лім.	Маса	Масштаб
Розроб.		Москалюк Ю.О.						
Перевір.		Решетнік В.М.						
Т. Контр.						Лист 1	Листів 1	
Реценз.						ХНУРЕ Кафедра СТ		
Н. Контр.								
Затверд.		Гребеннік І.В.						

ГЮИК.506100.008 С11

КЛІЄНТСЬКИЙ ІНТЕРФЕЙС СИСТЕМИ (ПЕРЕГЛЯД ТА ОЦІНКА ТОВАРІВ В КАТАЛОЗІ)

					ГЮИК.506100.008 С11			
					<i>Клієнтський інтерфейс системи (перегляд та оцінка товарів в каталозі)</i>	<i>Лім.</i>	<i>Маса</i>	<i>Масштаб</i>
<i>Змін.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Москалюк Ю.О.</i>						
<i>Перевір.</i>		<i>Решетнік В.М.</i>						
<i>Т. Контр.</i>								
<i>Реценз.</i>						<i>Лист 1</i>	<i>Листів 1</i>	
<i>Н. Контр.</i>						<i>ХНУРЕ Кафедра СТ</i>		
<i>Затверд.</i>		<i>Гребеннік І.В.</i>						

ГЮИК.506100.008 С11

КЛІЄНТСЬКИЙ ІНТЕРФЕЙС СИСТЕМИ (НАБІР РЕКОМЕНДОВАНИХ ТОВАРІВ)

RECOMMENDED

JORGE Archer Rouched Mini Dress Print	\$79	JORGE Archer Rouched Midi Skirt Print	\$69	THE PEOPLE VS Dreamland Helena Tank Vintage Black	\$59	THE PEOPLE VS Eagle Prey Ss Tee Dress Vintage Black	\$79	THE PEOPLE VS Thunder Crop Tee Vintage Black	\$59	TOMMY HILFIGER Tjw Crop Tiny Tommy 1 Tank White	\$59
---	------	---	------	---	------	---	------	--	------	---	------

					ГЮИК.506100.008 С11			
					<i>Клієнтський інтерфейс системи (набір рекомендованих товарів)</i>	<i>Лім.</i>	<i>Маса</i>	<i>Масштаб</i>
<i>Змін.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>	<i>Москалюк Ю.О.</i>							
<i>Перевір.</i>	<i>Решетнік В.М.</i>							
<i>Т. Контр.</i>						<i>Лист 1</i>		<i>Лист 1</i>
<i>Реценз.</i>						<i>ХНУРЕ</i>		
<i>Н. Контр.</i>					<i>Кафедра СТ</i>			
<i>Затверд.</i>	<i>Гребеннік І.В.</i>							

ГЮИК.506100.008 ДЗ

ДОПОВІДЬ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Проведена оцінка ефективності роботи алгоритму, шляхом вимірювання якості рекомендацій, використовуючи метрику root mean square error (RMSE).

$$RMSE = \sqrt{\frac{1}{|n|} \sum_{(u,o) \in n} (r'_{u,o} - r_{u,o})^2}$$

Згідно з результатами експериментальних досліджень, середньоквадратична помилка (кількість відсутніх значень в матриці) дорівнює 8.7%.

Проведений аналіз кількості продажів в системі електронної комерції з продажу жіночого одягу, здійснених користувачами, що відбулися в період з лютого 2021 року по листопад 2021 року.

Графік відображає приріст продажів після інтеграції алгоритму КФ фільтрації, **який склав**.



					ГЮИК.506100.008 ДЗ			
Змін.	Лист	№ докум.	Підпис	Дата	Аналіз ефективності функціонування алгоритму	Лім.	Маса	Масштаб
Розроб.		Москалюк Ю.О.						
Перевір.		Решетнік В.М.						
Т. Контр.								
Реценз.								
Н. Контр.								
Затверд.		Гребеннік І.В.				Лист 1	Лист 1	
						ХНУРЕ Кафедра СТ		

ГЮИК.506100.008 С11

ДОПОВІДЬ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ



СЕРТИФІКАТ
учасника конференції

IN 2063
від 29.10.2021

МОСКАЛЮК ЮЛІЯ ОЛЕКСАНДРІВНА

ВЗЯВ(ЛА) УЧАСТЬ У І МІЖНАРОДНІЙ СТУДЕНТСЬКІЙ НАУКОВІЙ КОНФЕРЕНЦІЇ
ФОРМУВАННЯ СУЧАСНОЇ НАУКИ:
МЕТОДИКА ТА ПРАКТИКА
29 ЖОВТНЯ 2021 РІК • М. КАМ'ЯНЕЦЬ-ПОДІЛЬСЬКИЙ, УКРАЇНА

В рамках участі було опубліковано тези доповіді учасника на тему:
ДОСЛІДЖЕННЯ СПОСОБІВ УРАХУВАННЯ УПОДОБАНЬ КЛІЄНТІВ ПРИ РОЗРОБЦІ СИСТЕМ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

ДИРЕКТОР МОЛОДІЖНОЇ НАУКОВОЇ ЛІГИ
ГОЛОВА ОРГКОМІТЕТУ КОНФЕРЕНЦІЇ
ІГОР КОРОНИК

29 жовтня 2021 рік • Кам'янець-Подільський, Україна • Молодіжна наукова ліга

**СЕКЦІЯ 17.
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ**

Москалюк Ю.О., здобувач вищої освіти
Харківський національний університет радіоелектроніки, Україна
Науковий керівник: Решетнік В.М., канд. техн. наук
Харківський національний університет радіоелектроніки, Україна

ДОСЛІДЖЕННЯ СПОСОБІВ УРАХУВАННЯ УПОДОБАНЬ КЛІЄНТІВ ПРИ РОЗРОБЦІ СИСТЕМ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

Протягом останніх років сфера торгівлі стрімко переміщується в Інтернет в галузь електронної комерції (е-комерції). Інтернет-магазини охоплюють буд-які напрямки торгівлі та зменю спрощують взаємодію між продавцями та клієнтами, дозволяють прикрити вихід певної одиниці бізнесу на ринок, скоротити витрати на рекламу тощо.

Для успішного розвитку торговельного бізнесу та підвищення прибутку необхідно вдосконалити процес торгівлі з урахуванням уподобань клієнтів. Користувачі стає все складніше робити вибір серед нескінченної кількості товарів, здійснювати точні пошуки. Тому, існує необхідність обрати товари з влучними «пошуковими» якими може запропонувати те, що може поглиблює сподобався користувачу. Основні способи урахування уподобань клієнтів базуються на використанні рекомендаційних систем, адже вони спрямовані на взаємодію з користувачем, якому надається певна кількість об'єктів, серед яких він і має зробити свій вибір.

Рекомендаційні системи – це підмножина систем для фільтрації контенту, які допомагають вирішити проблеми холодного старту, шкрябіття, масштабованості, проблеми користувачів щодо складності пошуку товарів або чимого вибору. Рекомендації підбираються на основі переважень та поведінки користувача. Система рекомендацій має спрогнозувати реакцію користувача на той чи інший елемент і запропонувати інші елементи, які також можуть стати до вподоби після того, як-система інтернет-магазину має показати користувачу схожі товари [1].

					ГЮИК.506100.008 С11			
					Доповідь результатів дослідження	Літ.	Маса	Масштаб
Змін.	Лист	№ докум.	Підпис	Дата				
<i>Розроб.</i>		<i>Москалюк Ю.О.</i>						
<i>Перевір.</i>		<i>Решетнік В.М.</i>						
<i>Т. Контр.</i>						Лист 1	Лист 1	
<i>Реценз.</i>						ХНУРЕ Кафедра СТ		

Додаток Б
Текст програми

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

«ЗАТВЕРДЖУЮ»

Керівник кваліфікаційної роботи

_____ доц. Решетнік В.М.
(підпис)

ДОСЛІДЖЕННЯ СПОСОБІВ ВРАХУВАННЯ УПОДОБАНЬ КЛІЄНТІВ ПРИ
РОЗРОБЦІ СИСТЕМ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ
ГЮОИК.506100.008 34 01 – 01 12 01 – ЛУ

РОЗРОБИЛА

ст. гр. ІТПм-20-1

Москалюк Ю.О.

2021

ЗАТВЕРДЖЕНО
ГЮИК.506100.008 34 01 – 01 12 01 – ЛУ

ДОСЛІДЖЕННЯ СПОСОБІВ ВРАХУВАННЯ УПОДОБАНЬ КЛІЄНТІВ ПРИ
РОЗРОБЦІ СИСТЕМ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

ГЮИК.506100.008 34 01 – 01 12 01

Сторінок 16

RegisterController.php

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\User;
use Illuminate\Foundation\Auth\RegistersUsers;
use Illuminate\Support\Facades\Validator;

class RegisterController extends Controller
{
    /*RegisterController Обробляє реєстрацію нових користувачів, а також
    перевірку та створення*/

    use RegistersUsers;
    /**
     * Перенаправлення користувача після реєстрації.
     *
     * @var string
     */
    protected $redirectTo = '/home';
    /**
     * Створення нового екземпляра контролера.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('guest');
    }
    /**
     * Валідація на запит реєстрації.
     *
     * @param array $data
     * @return \Illuminate\Contracts\Validation\Validator
     */
    protected function validator(array $data)
    {
        return Validator::make($data, [
            'name' => ['required', 'string', 'max:255'],
            'email' => ['required', 'string', 'email', 'max:255',
'unique:users'],
            'password' => ['required', 'string', 'min:8', 'confirmed'],
        ]);
    }

    /**
     * Create a new user instance after a valid registration.
     */
}
```

```
* @param array $data
* @return \App\User
*/
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => Hash::make($data['password']),
    ]);
}
}

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Order extends Model
{
    protected $table = 'orders';

    protected $fillable = [
        'id_order', 'id_user', 'status', 'first_name', 'last_name',
        'email'];

    public function user()
    {
        return $this->belongsTo(User::class, 'id_user');
    }

    public function cart()
    {
        return $this->hasMany(Cart::class);
    }
}
```

OrderController.php

```
<?php
namespace App\Http\Controllers;
use App\Order;
use App\Cart;
use App\User;
use App\Address;
use Illuminate\Http\Request;
use Session;
use Notification;
use App\Notifications\ShopNotification;
use App\Http\Resources\OrderCollection;
class OrderController extends Controller
```

```
{
/**
 * Відображає список ресурсів.
 *
 * @return \Illuminate\Http\Response
 */
public function index()
{
    $order = Order::first();
    return view('admin.order.index')->with('order', $order);
}
}
public function ordersList()
{
    return new OrderCollection( Order::all() );
    // $orders = Order::orderBy('status', 'desc')->get()->toArray();
    // return datatables()->of($orders)->make(true);
}
public function ordersTrack(Request $request)
{
    $order = Order::where('id_order', $request->id_order)->first();
    if ($order) {
        return back()->with('success', 'Order: '.$order->status);
    }else{
        return back()->withErrors('Invalid order number!.');
    }
}
public function ordersTrackIndex()
{
    return view('clothesshop.track');
}
/**
 * Відображає форму для створення нового ресурсу.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    //
}
/**
 * Зберігає нові створені ресурси.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function clothingshop(Request $request)
{
    $request->validate([
        'first_name' => 'required',
        'last_name' => 'required',
    ]);
}
```

```
        'email'      => 'required',
        // 'transactionId'      => 'string|min:3',
    ]);
    if ( empty(Cart::where('user_id', auth()->user()->id)-
>where('order_id', null)->first()) ) {
        return back()->withErrors('Cart empty!');

        $order = new Order;
        $order->id_order = 'ORD-'.strtoupper(uniqid());
        $order->id_user = auth()->user()->id;
        $order->status = 'pending';      }
        $order->first_name = $request->first_name;
        $order->last_name = $request->last_name;

/**/ Відображає ресурс.
 *
 * @param \App\Order $order
 * @return \Illuminate\Http\Response
 */
public function show(Request $request)
{
    $order = Order::find($request->id);
    return view('admin.order.show')->with('order', $order);
}
/**/
 * Відображає форму редагування.
 *
 * @param \App\Order $order
 * @return \Illuminate\Http\Response
 */
public function edit(Request $request)
{
    $order = Order::find($request->id);
    return view('admin.order.edit')->with('order', $order);
}
/**/
 * Відновлює ресурс в сховищі.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Order $order
 * @return \Illuminate\Http\Response
 */
public function update(Request $request)
{
    $order = Order::find($request->id);
    if($order->status == 'completed') return back();
    $order->status = $request->order_status;
    $order->save();
    if($request->order_status == 'completed'){
        foreach ($order->cart as $cart) {
            $product = $cart->product;
```

```

        $product->quantity -= $cart->quantity;
        $product->save();
    }
}
return redirect()->route('admin.product.order')-
>with('success','You have successfully update an order.');
```

```

}
/**
 * Видаляє ресурс зі сховища.
 *
 * @param \App\Order $order
 * @return \Illuminate\Http\Response
 */
public function destroy(Request $request)
{
    $order = Order::find($request->id);
    if ($order) {
        foreach ($order->cart as $cart) {
            $cart->delete();
        }
        $order->delete();
        return redirect()->route('admin.product.order')-
>with('success','You have successfully delete an order.');
```

```

    }else{
        return redirect()->route('admin.product.order')-
>withErrors('Invalid order!.');
```

```

    }}}}

```

CartController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Cart;
```

```
use App\Catalogue;
```

```
use Illuminate\Http\Request;
```

```
class CartController extends Controller
```

```
{
```

```
    public function index()
```

```
    {
```

```
        $carts = Cart::where('user_id', auth()->user()->id)-
>where('order_id', null)->get();
```

```
        return view('shop.cart')->with('carts', $carts);
```

```
    }
```

```
/**
```

```
 * Додавання в кошик.
```

```
 *
```

```
 */
```

```
public function addTo(Request $request){
```

```
    if (empty($request->slug)) {
```

```
        return back()->withErrors('Invalid product!');
    }

    $product = Product::where('slug', $request->slug)->first();
    if (empty($product)) {
        return back()->withErrors('Invalid product!');
    }
    $already_cart = Cart::where('id_user', auth()->user()->id)-
>where('id_order', null)->where('id_clothes_unit', $product->id)-
>first();
    if($already_cart) {
        $already_cart->amount = $already_cart->quantity + 1;
        $already_cart->price = $product->offer_price + $already_cart-
>price ;
        if ($already_cart->product->amount < $already_cart->amount ||
$already_cart->product->amount <= 0) return back()->withErrors('Stock not
sufficient!.');
        $already_cart->save();

    }else{

        $cart = new Cart;
        $cart->id_user = auth()->user()->id;
        $cart->id_clothes_unit = $clothes_unit->id;
        $cart->price = $product->offer_price;
        $cart->amount = 1;
        if ($cart->catalogue->amount < $cart->amounty || $cart-
>catalogue->amount <= 0) return back()->withErrors('Stock not
sufficient!.');
        $cart->save();
    }
    return back()->with('success', 'Product added to cart.');
```

```
    }
    /** Додавання в кошик зі сторінки товару.*/

    public function singleToAdd(Request $request){
        $request->validate([
            'slug'      => 'required',
            'qty'       => 'required',
        ]);
        $id_clothes_unit = Catalogue::where('slug', $request->slug)-
>first();
        if ( ($request->qty < 1) || empty($catalogue) ) {
            return back()->withErrors('Something wrong. Try again!');
        }
        $already_cart = Cart::where('id_user', auth()->user()->id)-
>where('id_order', null)->where('id_clothes_unit', $'id->clothes_unit')-
>first();
        if($already_cart) {
            $already_cart->amount = $already_cart->amount + $request-
>qty;
```

```

        $already_cart->price = ($product->offer_price * $request-
>qty) + $already_cart->price ;
        if ($already_cart->catalogue->amount < $already_cart->amount
|| $already_cart->product->amount <= 0) return back()->withErrors('Stock
not sufficient!.');
        $already_cart->save();

    }else{

        $cart = new Cart;
        $cart->id_user = auth()->id ->user;
        $cart->id_product = $product->id;
        $cart->price = ($product->offer_price * $request->qty);
        $cart->amount = $request->qty;
        if ($cart->product->quantity < $cart->quantity || $cart-
>product->quantity <= 0) return back()->withErrors('Stock not
sufficient!.');
        $cart->save();
    }
    return back()->with('success','Product added to cart.');
```

```

}
/**Видалення кошика*/
public function addToDelete(Request $request){
    $cart = Cart::find($request->id);
    if ($cart) {
        $cart->delete();
        return back()->with('success','Cart removed!');
    }
    return back()->withErrors('Invalid card');
}
/**
 * Оновлення кошика.*
public function addToUpdate(Request $request){
    if($request->qty){
        $error = array();
        $success = '';
        foreach ($request->qty as $k=>$qty) {
            $id = $request->qty_id[$k];
            $cart = Cart::find($id);
            if ($qty>0 && $cart) {
                $cart->quantity = ($cart->catalogue->amount > $qty) ?
$qty : $cart->catalogue->amount;
                if ($cart->catalogue->amount <=0) continue;
                $cart->price = $cart->catalogue->offer_price * $qty;
                $cart->save();
                $success = 'Cart updated!';
            }else{
                $error[] = 'Cart Invalid!';
            }
        }
    }
    return back()->withErrors($error)->with('success', $success);
}

```

```
    }else{
        return back()->withErrors('Cart Invalid!');}

```

Catalogue.php

```
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class Catalogue extends Model
{
    public function images(){
        return $this->hasMany(Image::class);
    }
    public function category(){
        return $this->belongsTo(Category::class, 'id_category');
    }
    public function brand(){
        return $this->belongsTo(Brand::class, 'id_brand');
    }
    public function size(){
        return $this->belongsTo(Brand::class, 'id_size');
    }

    public function cart(){
        return $this->hasMany(Cart::class)->whereNotNull('id_order');
    }
}
```

AdminController.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\User;
use App\Http\Resources\UserCollection;
class AdminController extends Controller
{
    /**Створити новий контролер*/
    public function __construct()
    {
        $this->middleware('auth');
    }

    public function index(){
        return view('admin.home');
    }
    public function users(){
        $users = User::orderBy('id', 'desc')->paginate(9);
        return view('admin.user.index')->with('users', $users);
    }
    public function userList()
    {
        return new UserCollection( User::all() );    }
}
```

UserController.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\User;
class UserController extends Controller
{
    public function index()
    {
        //
    }

    public function create()
    {
        //
    }
    /**
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response */
    public function store(Request $request)
    {
        //
    }
    /**
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        $user = User::find($id);
        return view('admin.user.view')->with('user', $user);
    }
    /**
     * Форма для редагування.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        //
    }
    /**
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request)
    {
        $user = User::find($request->id);
```

```

        $user->name = $request->name;
        $user->is_admin = $request->is_admin;
        $user->save();
        return back()->with('success', 'You have successfully update
profile. ');
    }
    /**
     * Видалення ресурсу зі сховища.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        //
    }
}

```

Showblade.php

```

@extends('layouts.master')

@section('title', $productSelected->name)

@section('content')
<!-- Portfolio Item Row -->
<div class="row mt-4">
    <div class="col-lg-6 col-md-12" data-aos="fade-right">
        <div id="carouselProductIndicators" class="carousel slide" data-
ride="carousel">
            <ol class="carousel-indicators mb-4">
                <li data-target="#carouselProductIndicators" data-slide-
to="{{ $i = 0 }}" class="active">
                    
                </li>
                @if (count($productSelected->image) > 1)
                @foreach (array_slice($productSelected->image, 1) as
$image)
                    <li data-target="#carouselProductIndicators" data-slide-
to="{{ $i = $i + 1 }}">
                        
                    </li>
                @endforeach
                @endif
            </ol>
            <div class="carousel-inner">
                <div class="carousel-item active">
                    
                </div>

```

```

        @if (count($productSelected->image) > 1)
        @foreach (array_slice($productSelected->image, 1) as
$image)
        <div class="carousel-item">
            
        </div>
        @endforeach
    @endif
</div>
<a class="carousel-control-prev"
href="#carouselProductIndicators" role="button" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-
hidden="true"></span>
    <span class="sr-only">Previous</span>
</a>
<a class="carousel-control-next"
href="#carouselProductIndicators" role="button" data-slide="next">
    <span class="carousel-control-next-icon" aria-
hidden="true"></span>
    <span class="sr-only">Next</span>
</a>
</div>
</div>
<div class="col-lg-6 col-md-12 text-center" data-aos="fade-left">
    <p class="mt-2 mb-4">
        {{ trans('product.category_' . $productSelected->gender,
['category' => $categorySelected->name]) }}
    </p>
    <h3 class="my-2 text-uppercase text-primary">{{ $productSelected-
>name }}</h3>
    <h3 class="my-4">$<b>{{ $productSelected->price }}</b></h3>
    {{ Form::open(['class' => 'form-horizontal']) }}
    <div class="form-group row">
        <label class="col-md-6 col-form-label text-uppercase">
            <b>{{ trans('product.color') }}</b>
        </label>
        <div class="col-md-6">
            <select class="form-control" name="color" id="color">
                @foreach ($productSelected->colors()->pluck('name')-
>sort() as $color)
                    <option value="{{ $color }}">
                        {{ $color }}
                    </option>
                @endforeach
            </select>
        </div>
    </div>
    <div class="form-group row">
        <label class="col-md-6 col-form-label text-uppercase">
            <b>{{ trans('product.size') }}</b>

```

```

        </label>
        <div class="col-md-6">
            <select class="form-control" name="size" id="size">
                @foreach ($productSelected->sizes()->pluck('name')-
>sort() as $size)
                    <option value="{{ $size }}">
                        {{ $size }}
                    </option>
                @endforeach
            </select>
        </div>
    </div>
    <div class="form-group row">
        <label class="col-md-6 col-form-label text-uppercase">
            <b>{{ trans('product.quantity') }}</b>
        </label>
        <div class="col-md-6">
            <select class="form-control" name="qty" id="qty">
                @for ($i = 1; $i<=10; $i++) <option value="{{ $i }}">
                    {{ $i }}
                </option>
                @endfor
            </select>
        </div>
    </div>
    <div class="form-group row mb-0">
        <div class="col-md-12">
            {{ Form::hidden('productId', $productSelected->id, ['id'
=> 'productId']) }}
            <button type="button" class="btn btn-primary btn-block
text-uppercase" id="addToCart">
                {{ trans('product.add') }}
            </button>
        </div>
    </div>
    {{ Form::close() }}
    <div class="my-4">
        <p>{{ $productSelected->description }}</p>
        <a class="text-uppercase" href="#review">
            {{ trans('product.read_review', ['reviews' =>
count($reviews)]) }}
        </a>
    </div>
</div>
</div>
</div>
<html>
<head>
<link rel = "stylesheet" href =
"https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

```

```
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></scri
pt>
<script scr -
"https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></s
cript>
<div class = "container">
<h2><div class = "well text-container" style="background-
color: CadetBlue; border: none; font-size: 25px; font-weight: bold; text-
align: center">Recommendation For You:</div></h2>
</div>
</head>
</html>
<?php

include("db.php");
include("recommend.php");

$product=mysqli_query($con, "select * from user_products");

$matrix=array();

while($p=mysqli_fetch_array($product))
{
    $users=mysqli_query($con, "select username from users where
    id=$p[user_id]");
    $username=mysqli_fetch_array($users);

    $matrix[$username['username']][$p['product_name']]=$p['product_rating'];
}

$users=mysqli_query($con, "select username from users where
id=$_GET[id]");
$username=mysqli_fetch_array($users);
?>
<div class="panel panel-default">

<div class = "panel-heading">
<h2>
<a class = "btn btn-success" href = "adduser.php">Add Users</a>
<a class = "btn btn-info pull-right" href = "index.php">Back</a>
</h2>
</div>

<div class="panel-body">
<table class="table table-stripped">
<th> Product Name</th>
<th> Product Rating</th>

<?php
```

```
$recommendation=array();
$recommendation=getRecommendation($matrix,$username['username']);

foreach($recommendation as $product=>$rating)
{
?>
<tr>
<td><?php echo $product; ?> </td>
<td><?php echo $rating; ?> </td>
</tr>
<?php } ?>
</table>
</div>
</div>

# custom function to create pearson correlation method from scratch

def person_corelation(person1,person2):
    both_rated = {}
    for item in dataset[person1]:
        if item in dataset[person2]:
            both_rated[item] = 1

    number_of_ratings = len(both_rated)
    if number_of_ratings == 0:
        return 0

    person1_preferences_sum = sum([dataset[person1][item] for item in
both_rated])
    person2_preferences_sum = sum([dataset[person2][item] for item in
both_rated])

    # Sum up the squares of preferences of each user
    person1_square_preferences_sum = sum([pow(dataset[person1][item], 2)
for item in both_rated])
    person2_square_preferences_sum = sum([pow(dataset[person2][item], 2)
for item in both_rated])

    # Sum up the product value of both preferences for each item
    product_sum_of_both_users = sum([dataset[person1][item] *
dataset[person2][item] for item in both_rated])

    # Calculate the pearson score
    numerator_value = product_sum_of_both_users - (
    person1_preferences_sum * person2_preferences_sum /
number_of_ratings)
    denominator_value = sqrt((person1_square_preferences_sum -
pow(person1_preferences_sum, 2) / number_of_ratings) * (
    person2_square_preferences_sum - pow(person2_preferences_sum, 2) /
number_of_ratings))
    if denominator_value == 0:
```

```
        return 0
    else:
        r = numerator_value / denominator_value
        return r

# custom function to check most similar users

def most_similar_users(target_person,no_of_users):

    # Used list comprehension for finding pearson similarity between
    users
    scores =
    [(person_corelation(target_person,other_person),other_person) for
    other_person in dataset if other_person !=target_person]

    #sort the scores in descending order
    scores.sort(reverse=True)

    #return the scores between the target person & other persons
    return scores[0:no_of_users]

# function check by input one person name & returns the similarity score
most_similar_users('Mohan',6)

#custom function to filter the seen items and unseen items of the target
user

def target_items_to_users(target_person):
    target_person_items_lst = []
    unique_list =unique_items()
    for items in dataset[target_person]:
        target_person_items_lst.append(items)

    s=set(unique_list)
    recommended_items =list(s.difference(target_person_items_lst))
    a = len(recommended_items)
    if a == 0:
        return 0
    return recommended_items,target_person_items_lst

def recommendation_phase(person):
    # Gets recommendations for a person by using a weighted average of
    every other user's rankings
    totals = {} #empty dictionary
    simSums = {} # empty dictionary
    for other in dataset:
        # don't compare me to myself
        if other == person:
            continue
        sim = person_corelation(person, other)
```

```
# ignore scores of zero or lower
if sim <= 0:
    continue
for item in dataset[other]:
    # only score items i haven't seen yet
    if item not in dataset[person]:
        # Similarity * score
        totals.setdefault(item, 0)
        totals[item] += dataset[other][item] * sim
        # sum of similarities
        simSums.setdefault(item, 0)
        simSums[item] += sim
        # Create the normalized list

rankings = [(total / simSums[item], item) for item, total in
totals.items()]
rankings.sort(reverse=True)
# returns the recommended items
recommendations_list = [(recommend_item,score) for score,
recommend_item in rankings]
return recommendations_list
print("Enter the target person")
tp = input().title()
if tp in dataset.keys():
    a=recommendation_phase(tp)
    if a != -1:
        print("Recommendation Using User based Collaborative Filtering:
")
        for items weights in a:
            print(items,'---->',weights)
else:
    print("Person not found in the dataset..please try again")
```

Додаток В
Відомість дипломної роботи

