

The background of the slide is a dense field of small, light purple flowers with darker purple centers. A thick white L-shaped frame is positioned around the text, starting from the top-left corner and extending towards the bottom-right corner, leaving a rectangular opening for the content.

# THE SIMPLE SHOP

Technical presentation

# Presentation Plan

- General information about the website
- Sign in and log out mechanism
- Brief information about Filter
- Shopping cart and checkout process
- Further development



# Database



# Sign in and log out mechanism

**Sign Up!**

Full Name

Email

Username

Password

Repeat password

Sign Up

**Log In!**

ewa123

.....

Log In!

**includes\dbh.inc.php** – connection to the database

**includes\functions.inc.php** – the backend file of signup and login form with requirements for filling out etc.

**signup.php** – signup site, the visible part

**login.php** - login site, the visible part

**includes\login.inc.php** - connects dbh.inc.php, functions.inc.php, and login.php, triggered by clicking “login!” button, spits out error messages in HTTP header

**includes\logout.inc.php** – logging out function, destroying the session

**includes\signup.inc.php** – the same as login.inc.php only for signup

# Examples of login/sign up constraints

## includes\functions.inc.php file

```
//requirements for filling out the signup form
function emptyInputSignup($name, $email, $username, $pwd, $pwdRepeat) {
    $result;
    if (empty($name) || empty($email) || empty($username) || empty($pwd) || empty($pwdRepeat)){
        $result = true;
    }
    else {
        $result = false;
    }
    return $result;
}

function invalidUid($username) {
    $result;
    if (!preg_match("/^[a-zA-Z0-9]*$/", $username)) {
        $result = true;
    }
    else {
        $result = false;
    }
    return $result;
}

function invalidEmail($email) {
    $result;
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)){
        $result = true;
    }
}
```



# Database for users

usersId	usersName	usersEmail	usersUid	usersPwd
1	ewa	ewa@gmail.com	ewa123	\$2y\$10\$vUYT28RrZI1kxEQqJc6pF..Km9zpFxC6HFKJa.X0/F/CxYmQHsXJe
2	ala ali	alaali@gmail.com	alaala	\$2y\$10\$O1dyFf2ZX2Ludi05idqhtuBBMAoGLfZt.uXOYRmi6uyz.CVa9BQxe
3	asdf	asdf@gmail.com	asdf	\$2y\$10\$mxr.3Jk8Yn4i8QowttfgAOI/7PCRVC5PEtneYFXmF/9/uBIPCuXU2
4	Rayan Willams	raywilliams@gmail.com	rayray67	\$2y\$10\$UtQFxfweVHpFlmcQRzjTqO3xS0hOcykqDxBMhyGI002qCylz0pzxW
5	Daniel Nellson	catcat@gmail.com	catcat66	\$2y\$10\$I0NxSLkVga1NGyKG2bAEPutIPYrwK2009BmcMpgwd4YvhQcjeNudi
6	Mary Lou	marylou@gmail.com	marylou89	\$2y\$10\$QB7nGa8G/ERXmxupeida2uR3CQ0gV7vLT9VQVle7yGzo.u6lHomlC
7	Adam Kowalski	AdamK@gmail.com	AdamK97	\$2y\$10\$ZxrDgLQcb285Lgusji7LLOQqoOrcBBKgtmyx1umhYGyaZ521JSvce

# Filtering of products

## Price

1 - 100



## Category

☒ woman

☐ man

☐ girl

☐ boy

## Type of garment

☐ top

☐ pants

☐ jeans

☐ blouse

## Color

☐ white

☐ violet

☐ orange

☐ khanki

☐ grey

☐ green

☐ cream

☐ brown

☐ blue

☐ black



# Database for products

item_id	item_gender	item_type	item_name	item_color	item_price	item_actual_price	item_image	item_image2	item_image3	item_image4	item_quantity	item_status
1	woman	top	Cream jersey body	cream	15.99	NULL	./images/1/4.jpg	./images/2/4.jpg	./images/3/4.jpg	./images/4/4.jpg	100	1
3	woman	jeans	Blue boyfriend jeans	blue	49.99	NULL	./images/1/8.jpg	./images/2/8.jpg	./images/3/8.jpg	./images/4/8.jpg	100	1
5	man	blouse	Orange hoodie	orange	29.99	NULL	./images/1/3.jpg	./images/2/3.jpg	./images/3/3.jpg	./images/4/3.jpg	100	1
6	man	blouse	Brown turtleneck	brown	59.99	NULL	./images/1/7.jpg	./images/2/7.jpg	./images/3/7.jpg	./images/4/7.jpg	100	1
7	boy	blouse	Blue shirt	blue	19.99	NULL	./images/1/1.jpg	./images/2/1.jpg	./images/3/1.jpg	./images/4/1.jpg	100	1
8	girl	top	Violet unicorn T-shirt	violet	9.99	NULL	./images/1/2.jpg	./images/2/2.jpg	./images/3/2.jpg	./images/4/2.jpg	100	1
9	girl	jeans	Grey jeans	grey	35.99	NULL	./images/1/6.jpg	./images/2/6.jpg	./images/3/6.jpg	./images/4/6.jpg	100	1
10	boy	pants	Khaki pants with neon green stripe	khaki	14.99	50.59	./images/1/5.jpg	./images/2/5.jpg	./images/3/5.jpg	./images/4/5.jpg	10	1

# Example of fetching checkbox values

```
$query = "SELECT DISTINCT(item_gender) FROM product WHERE item_status = '1' ORDER BY item_gender DESC";

$statement = $connect->prepare($query);
$statement->execute();
$result = $statement->fetchAll();
foreach($result as $row)
{
    ?>
    <div class="list-group-item checkbox">
        <label><input type="checkbox" class="common_selector gender" value="<?php echo $row['item_gender']; ?>" > <?php echo $row['item_gende
    </div>
    <?php
    }
    ?>
    </div>
</div>
<script>
    $(function(){
        $('<div> .gender').each(function(e){
            if($(this).val() == 'girl'){
                $(this).attr("checked", "checked");
            }
        });
    });
</script>
```

# Application of AJAX

```
<script>
$(document).ready(function(){
    filter_data();
    function filter_data()
    {
        $('#filter_data').html('<div id="loading" style="" ></div>');
        var action = 'fetch_data';
        var minimum_price = $('#hidden_minimum_price').val();
        var maximum_price = $('#hidden_maximum_price').val();
        var gender = get_filter('gender');
        var type = get_filter('type');
        var color = get_filter('color');

        $.ajax({
            url: "fetch_data.php",
            method: "POST",
            data: {action: action, minimum_price: minimum_price, maximum_price: maximum_price, gender: gender,
            success: function(data){
                $('#filter_data').html(data);
            }
        });
    }
}
```

# Examples of filtering constraints

```
if(isset($_POST["action"]))
{
    $query = "
        SELECT * FROM product WHERE item_status = '1' ";
    if(isset($_POST["minimum_price"], $_POST["maximum_price"]) && !empty($_POST["minimum_price"]) && !empty($_POST["maximum_price"]))
    {
        $query .= "
            AND item_price BETWEEN '".$_POST["minimum_price"]." AND '".$_POST["maximum_price"]."
        ";
    }
    if(isset($_POST["gender"]))
    {
        $gender_filter = implode("'", $_POST["gender"]);
        $query .= "
            AND item_gender IN('".$gender_filter."')
        ";
    }
}
```



# JavaScript code for image carousel



```
/*img carousel*/  
const productImages = document.querySelectorAll(".product-images img");  
const productImageSlide=document.querySelector(".image-slider");  
let activeImageSlide = 0;  
  
productImages.forEach((item, i) => {  
  item.addEventListener('click', () => {  
    productImages[activeImageSlide].classList.remove('active');  
    item.classList.add('active');  
    productImageSlide.style.backgroundImage=`url('${item.src}')`;  
    activeImageSlide=i;  
  })  
})
```

# Shopping cart and checkout process

## Shopping Cart

Image	Product	Price	Quantity	Total	
	Grey jeans	\$35.99	<input type="text" value="1"/>	\$35.99	<button>Delete</button>
	Blue boyfriend jeans	\$49.99	<input type="text" value="1"/>	\$49.99	<button>Delete</button>
Cart Total			\$85.98		

[Continue Shopping](#)

[Checkout](#)

```

// Insert items into the cart and save it to the session
public function insert($item = array()){
    if(!is_array($item) OR count($item) === 0){
        return FALSE;
    }else{
        if(!isset($item['id'], $item['name'], $item['price'], $item['qty'])){
            return FALSE;
        }else{
            // prep the quantity
            $item['qty'] = (float) $item['qty'];
            if($item['qty'] == 0){
                return FALSE;
            }
            // prep the price
            $item['price'] = (float) $item['price'];
            // create a unique identifier for the item being inserted into the cart
            $rowid = md5($item['id']);
            // get quantity if it's already there and add it on
            $old_qty = isset($this->cart_contents[$rowid]['qty']) ? (int) $this->cart_contents[$rowid]['qty'] : 0;
            // re-create the entry with unique identifier and updated quantity
            $item['rowid'] = $rowid;
            $item['qty'] += $old_qty;
            $this->cart_contents[$rowid] = $item;

            // save Cart Item
            if($this->save_cart()){
                return isset($rowid) ? $rowid : TRUE;
            }else{
                return FALSE;
            }
        }
    }
}

```



# Checkout

## Contact Details

First Name

Justyna

Last Name

Neblik

Email

JustynaNeblik@gmail.com

Phone

787878565

Address

Gliwce, Kujawska, 2

Place Order

## Your Cart

3

Grey jeans	\$71.98
\$35.99(2)	
Blue boyfriend jeans	\$49.99
\$49.99(1)	
Total	<b>\$121.97</b>

Add Items

# Database for customers

id	first_name	last_name	email	phone	address	created	modified	status
								1=Active   0=Inactive
16	Donna	Waterford	Donna456@gmail.com	785757575	jasmin 5, Houston	2022-01-05 23:57:19	2022-01-05 23:57:19	1
17	Andrew	Hopkins	world55@gmail.com	456789456789	greenville 7, Glasgow	2022-01-06 20:22:44	2022-01-06 20:22:44	1
18	Kelly	Rabbit	blackjack678@gmail.com	756454545	blue ally 7, Berlin	2022-01-06 20:25:46	2022-01-06 20:25:46	1
19	Daniell	Huston	Danielred@gmail.com	789123123	Huston, Rose ally 5	2022-01-08 23:19:50	2022-01-08 23:19:50	1

# Database for orders

id	customer_id	grand_total	created	status
16	16	49.99	2022-01-05 23:57:19	Pending
17	17	135.98	2022-01-06 20:22:44	Pending
18	18	49.99	2022-01-06 20:25:46	Pending
19	19	149.98	2022-01-08 23:19:50	Pending

# Database for order details

id	order_id	item_id	quantity
20	16	3	1
21	17	3	1
22	17	16	1
23	17	1	1
24	18	3	1
25	19	7	1
26	19	6	1
27	19	16	1

# ORDER STATUS

Your order has been placed successfully.

## Order Info

**Reference ID:** #21

**Total:** \$121.97

**Placed On:** 2022-01-14 22:39:32

**Buyer Name:** Justyna Neblik


**Email:** JustynaNeblik@gmail.com

**Phone:** 787878565

Product	Price	QTY	Sub Total
Grey jeans	\$35.99	2	\$71.98
Blue boyfriend jeans	\$49.99	1	\$49.99

# Further development





# Thank you for your attention!

**Authors :**  
**Justyna Neblik**  
**Wojciech Obrzut**