Individual Project

Store Front
Github Repo

CSC-17B

Date: 06/11/2023

Josue Carlito

Extra Github Link: https://github.com/JCarlito/StoreFrontWeb

Implementation:

Implementation began with structuring the sql tables before writing any code. I took this approach so I could visualize how I was going to go about storing data. Then I went on to creating the simple store front page and then went onto creating the cart. Once the store front and cart worked, I then began to create the User and Admin pages for signing in and keeping track of customers

<u>Instructions on How to Run 17B GUIStoreFrontv4</u>

- 1. Create an SQL schema named "storefront". If you can't create a schema with the name "storefront" then name it anything and then navigate to the db_connection.php file in 17B GUIStoreFrontv4 and change the \$db variable to fit the name chosen.
- 2. Next import the storefront sql file into SQL workbench and run the query inside the storefront schema to set up the database.
- 3. Next make sure in XAMPP that mySQL and Apache are running
- 4. Then open the 17B_GUIStoreFrontv4 project and run the sign_in.html file and navigate through the store freely

NOTE: If the following credentials do not work. A user account can simply be created. To create an admin account an admin key is necessary which is "1234" in case the admin credentials do not work.

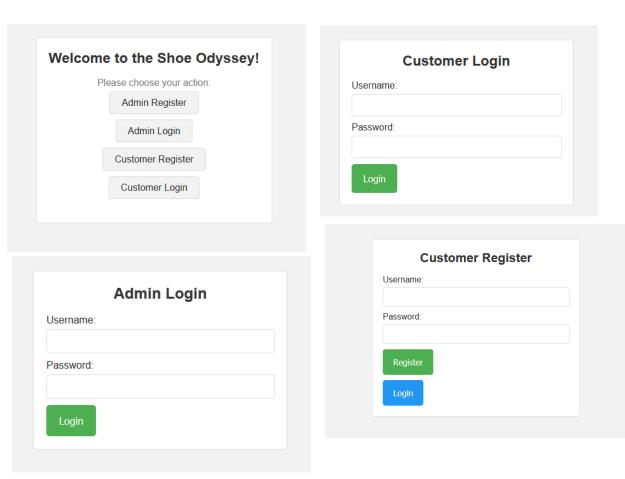
Login Credentials

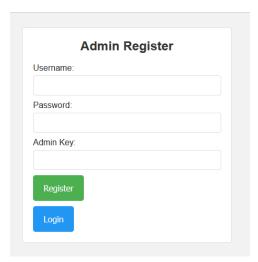
Admin username: mlehr password: mlehr

User username: test123 password: test123

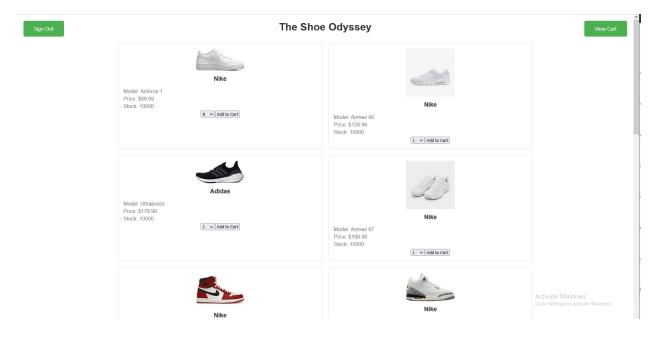
Sample Input/Output

Login/Registration

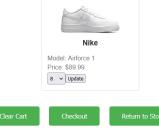




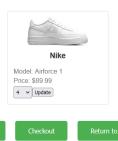
Store Front







Shopping Cart



Shopping Cart



Welcome to the Admin Page

Add Inventory

Edit Inventory

Edit Users

View Purchase History

Total Revenue

Add Inventory Item

Brand:
Model:
Price:
Stock:
Shoe Image:
Browse No file selected.
Add Item



View User Purchase History

Username	Actions
jcarlito	View Purchase History
gcarlito	View Purchase History
Icarlito	View Purchase History
test123	View Purchase History
test12345	View Purchase History



Revenue

Brand	Model	Quantity	Price	Total	Order ID	Date
Nike	Airforce 1	1	\$89.99	\$89.99	11	05/18/2023
Nike	Airforce 1	11	\$89.99	\$989.89	12	05/18/2023
Nike	Air Jordan 1	10	\$219.99	\$2199.90	13	05/19/2023
Nike	Air Jordan 3	3	\$199.99	\$1819.91	14	05/19/2023
Nike	SB Dunk	2	\$109.99	\$1819.91	14	05/19/2023
Nike	Air Jordan 4	4	\$249.99	\$1819.91	14	05/19/2023
Nike	Airforce 1	4	\$89.99	\$1709.90	15	05/19/2023
Nike	Airmax 90	5	\$129.99	\$1709.90	15	05/19/2023
Nike	Airmax Sean Wotherspoons	1	\$699.99	\$1709.90	15	05/19/2023

Total Revenue: \$6809.59

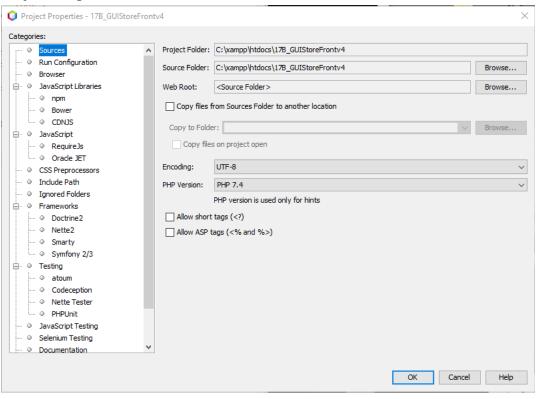
Summary

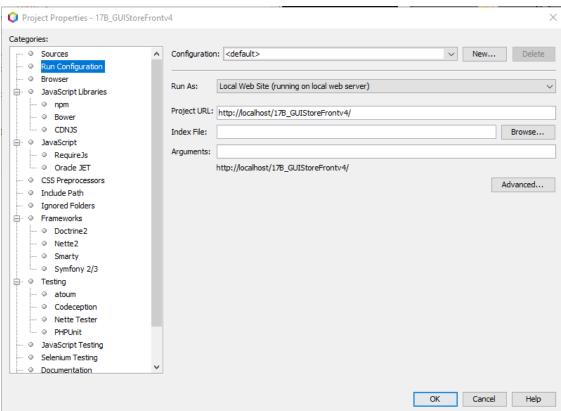
Folder/Files Layout 17B GUIStoreFrontv4

Source Files

- Images
 - Holds image paths that can be used for shoes when adding inventory
- Php scripts
 - Add_item.php
 - Add_to_cart.php
 - Admin_login_process.php
 - Admin register process.php
 - Checkout cart.php
 - Clear cart.php
 - Customer login process.php
 - Customer register process.php
 - Edit item.php
 - Edit user.php
 - Sign out.php
 - Update cart.php
 - View user history.php
- Add inventory.php
- Admin.php
- Admin login.html
- Admin register.html
- Checkout_success.php
- Customer login.php
- Customer_register.php
- Db connection.php
- Edit_inventory.php
- Edit users.php
- Shopping cart.php
- Sign in.html
- Store front.php
- Total revenue.php
- view purchase history.php

Project Properties





- 1. Objects Javascript/PHP Serialization
 - a. \$_SESSION in PHP is an object
- 2. SQL Store Front Database

Users SQL Tables

Admins:

	admin_id	username	password	
D	5	mlehr	\$2y\$10\$FYBB	
♦	<autofield></autofield>			

Cart:

	cart_id	user_id	product_id	quantity		
>>	<autofield< th=""><th></th><th></th><th></th></autofield<>					

Inventory:

	product_id	brand	model	price	stock	status	image
D	1	Nike	Airforce 1	89.99	10000	~	/images/
	2	Nike	Airmax 90	129.99	10000	~	/images/
	3	Adidas	Ultraboost	179.99	10000	~	/images/
	4	Nike	Airmax 97	169.99	10000	~	/images/
	5	Nike	Air Jordan	219.99	10000	~	/images/
	6	Nike	Air Jordan	199.99	10000	~	/images/
	7	Nike	Air Jordan	249.99	10000	~	/images/
	8	Nike	SB Dunk	109.99	10000	~	/images/
	9	Vans	Old Skool	49.99	10000	~	/images/
	10	Puma	RS-X	119.99	10000	~	/images/
	13	Nike	Air Monar	69.99	10000	~	/images/
	14	Nike	Airmax 95	139.99	10000	~	/images/
	18	Nike	Airmax Se	699.99	10000	~	/images/
❖	<autofield></autofield>						

Order_details:

	order_detail_id	order_id	product_id	quantity	price
>	9	11	1	1	89.99
	10	12	1	11	89.99
	11	13	5	10	219.99
	12	14	6	3	199.99
	13	14	8	2	109.99
	14	14	7	4	249.99
	15	15	1	4	89.99
	16	15	2	5	129.99
	17	15	18	1	699.99
♦	<autofield></autofield>				

Orders:

	order_id	user_id	total
D	11	6	89.99
	12	6	989.89
	13	7	2199.90
	14	8	1819.91
	15	8	1709.90
♦	<autofield></autofield>		

Users:

ł		user_id	username	password	status
ł	D	6	jcarlito	\$2y\$10\$g8Uy	~
ł		7	gcarlito	\$2y\$10\$wSe1	~
I		8	Icarlito	\$2y\$10\$PUCV	~
1		9	test123	\$2y\$10\$Kmal	~
I		10	test12345	\$2y\$10\$dWw	~
	٥	<autofield< th=""><th></th><th></th><th></th></autofield<>			
ľ					

- 3. Form Validation Regular Expressions
 - a. Registration uses form validation and hashing for user information and verification

4. User-Admin-Login

- a. Admin interface implemented. Can edit users' status, username, password. Add inventory and edit inventory as well. Also view user purchase history and total revenue
- 5. Cookies Sessions Securing Pages
 - a. Login and logout functions use PHP sessions.
 - b. User can not directly access the store or admin page without login in. If they try to access a page they will be redirected.