

All scripts and webpages have all the following requirements to run: A successful login for the index.html page, a internet connection that does not restrict web access to the website and SQL database, and an up to date browser that can support javascript and HTML 5.

1. **index.html:** Verifies a user of the account so they will have access to the websites features. This will eventually be necessary for the _____ pages.

a) Important Code: `$('#form').submit(function(event){...}` - Submits the users information to the login.php script where that page will store the user's information so they have access to all the necessary functions.

2. **Warehouse.html** : Gets the user's personal warehouse information. This page will present a dynamic GUI that will present a model of the warehouse in a 10x10 ft squares.

a.) Important Code: `.get('getWarehouse.php', function(data){....})` - Gets the user's warehouse information including its size, its name, and ID. This information will pull it from the session page of Login.php eventually.

3. **Create.php** : Will create users for the website. This is given to Admin accounts so they can assign different roles for their warehouse.

a.) Important Code: `$query = "INSERT INTO users (username, password, addDate, changeDate) VALUES ('$username', '$password', now(), '$DOB');";` - Submits user data to the SQL database.

4. **getItems.php** : Gets all items from the item catalog (item database). This will mainly be used for customers for shopping they can then add items in a shopping cart where the items are stored in a php session.

a.) Important Code: `$query = "Select * FROM user;";` - gets all items from the database.

5. **dbcontroller.php** : creates a session for user to login. We have no created a secure login session for this script so we will keep it offline until then.

6. **Search.html** - used to create an instance of a shopping cart for the user. It does this by create a php session that stores each item that the user clicks on to order.

a.) Important Code:

```
case "remove":
if(!empty($_SESSION["cart_item"])) {
    foreach($_SESSION["cart_item"] as $k => $v) {
        if($_GET["code"] == $k)
```

```

        unset($_SESSION["cart_item"][$k]);
        if(empty($_SESSION["cart_item"]))
            unset($_SESSION["cart_item"]);
    }

    break;
case "empty":
    unset($_SESSION["cart_item"]);
    break;

```

This removes items from the shopping cart. If the shopping cart becomes empty it will end the php session.

```

if(!empty($_POST["quantity"])) {
    $productByCode = $db_handle->runQuery("SELECT * FROM Items");

```

This creates a list of all available items to be purchased.

```

if(!empty($_SESSION["cart_item"]))
{

if(in_array($productByCode[0]["itemCode"],array_keys($_SESSION["cart_item"]))) {
    foreach($_SESSION["cart_item"] as $k => $v) {
        if($productByCode[0]["itemCode"] == $k) {

if(empty($_SESSION["cart_item"][$k]["quantity"])) {

$_SESSION["cart_item"][$k]["quantity"] = 0;
        }
        $_SESSION["cart_item"][$k]["quantity"] +=
$_POST["quantity"];
    }
}
} else {
    $_SESSION["cart_item"] =
array_merge($_SESSION["cart_item"],$itemArray);
}
} else {
    $_SESSION["cart_item"] = $itemArray;
}
}
}

```

This will create the shopping cart queue and add the selected item to the queue.

7. **SplashPage.php** : This is essential the same as search.php for now.

8. Unnecessary, unfinished, or retired pages : splashpage.html, getWarehouse.html, searchItems.php, test.php.