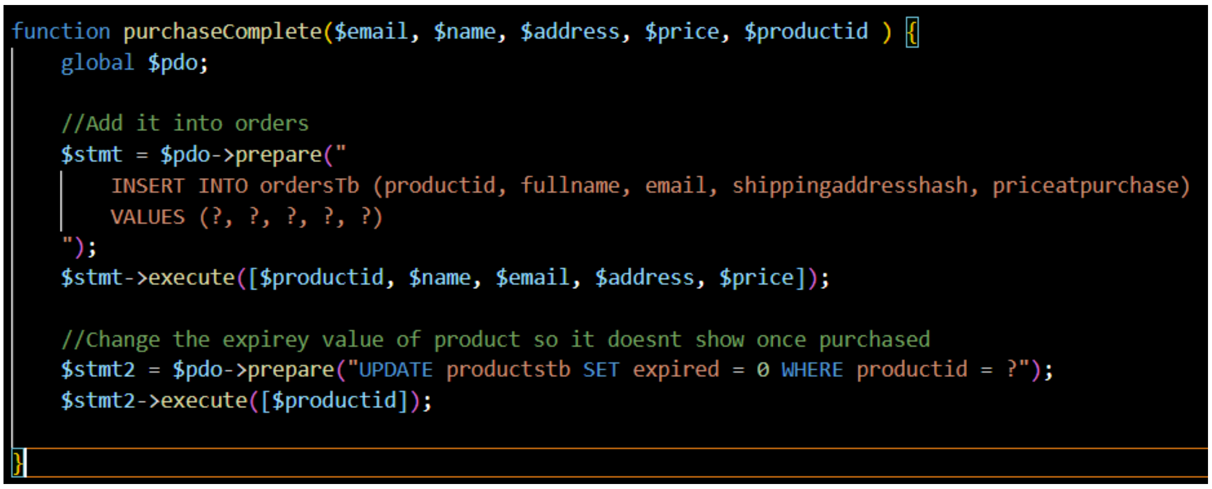
Live Stock Backend Detailed Changes:

Database & Backend Logic Updates:

1. Added new insert function in db.php

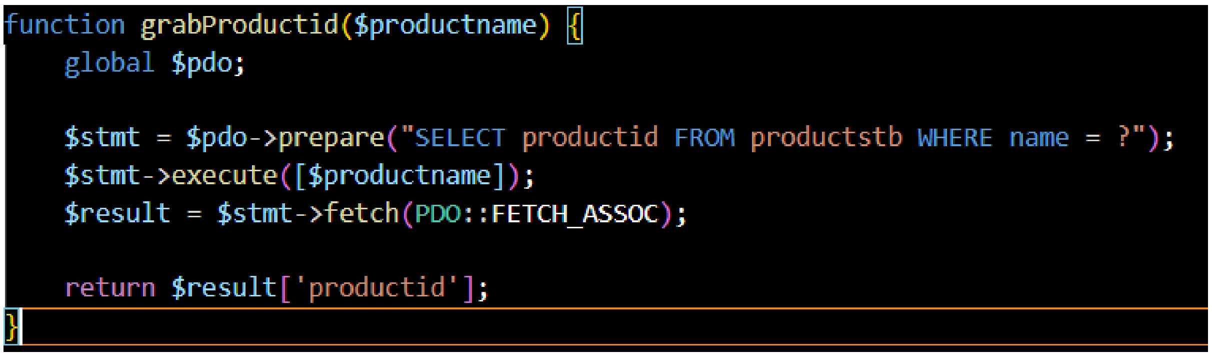
Previously the database file only had a function for retrieving data.
A new function has now been added to handle inserting records into the database properly.



```
function purchaseComplete($email, $name, $address, $price, $productid ) {
    global $pdo;

    //Add it into orders
    $stmt = $pdo->prepare("
        INSERT INTO ordersTb (productid, fullname, email, shippingaddresshash, priceatpurchase)
        VALUES (?, ?, ?, ?, ?)
    ");
    $stmt->execute([$productid, $name, $email, $address, $price]);

    //Change the expirey value of product so it doesnt show once purchased
    $stmt2 = $pdo->prepare("UPDATE productstb SET expired = 0 WHERE productid = ?");
    $stmt2->execute([$productid]);

}
```

2. Added function to get product ID from product name

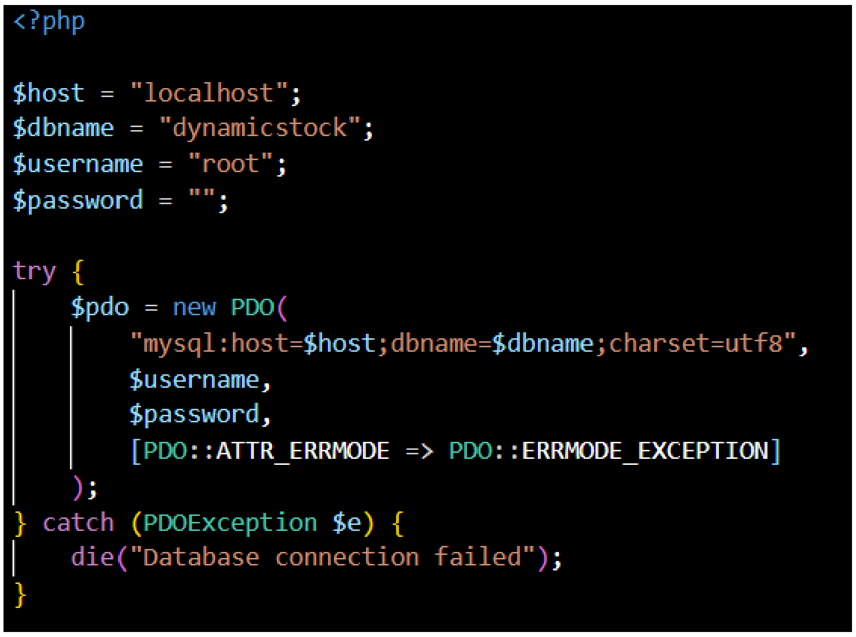
Created a separate function that returns the product ID when given the product name.
This is used to avoid repeating queries and keeps the order handling cleaner.



```
function grabProductid($productname) {
    global $pdo;

    $stmt = $pdo->prepare("SELECT productid FROM productstb WHERE name = ?");
    $stmt->execute([$productname]);
    $result = $stmt->fetch(PDO::FETCH_ASSOC);

    return $result['productid'];
}
```

3. Basic database connection established

The backend now establishes a simple database connection and successfully passes data through it.
Nothing is hashed at this stage, and the leftover hashing fields in the products table still need to be removed.

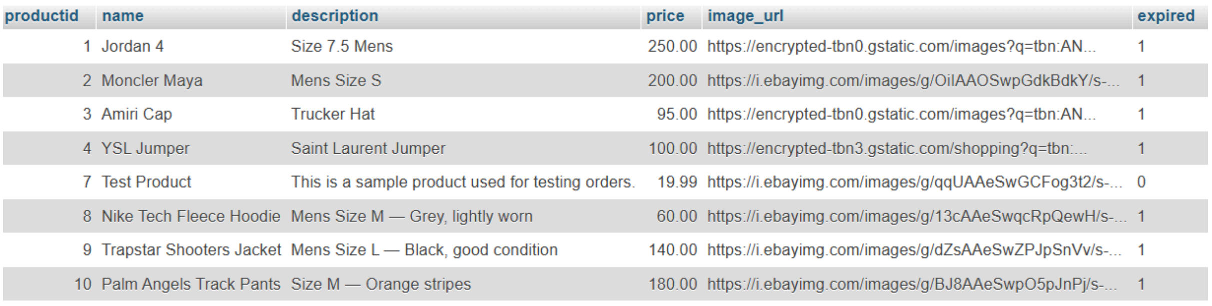


```php
<?php

$host = "localhost";
$dbname = "dynamicstock";
$username = "root";
$password = "";

try {
    $pdo = new PDO(
        "mysql:host=$host;dbname=$dbname;charset=utf8",
        $username,
        $password,
        [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]
    );
} catch (PDOException $e) {
    die("Database connection failed");
}
```

4 . Added expiry boolean to products

A new expiry field was added to the products table.
 This allows expired or inactive products to remain in the database for records but not appear on the website as it has been purchased.

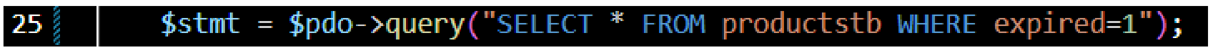
- 1 = visible
- 0 = hidden
- Stored using a BIT data type



productid	name	description	price	image_url	expired
1	Jordan 4	Size 7.5 Mens	250.00	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	1
2	Moncler Maya	Mens Size S	200.00	https://i.ebayimg.com/images/g/OilAAOSwpGdkBdkY/s-...	1
3	Amiri Cap	Trucker Hat	95.00	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	1
4	YSL Jumper	Saint Laurent Jumper	100.00	https://encrypted-tbn3.gstatic.com/shopping?q=tbn:...	1
7	Test Product	This is a sample product used for testing orders.	19.99	https://i.ebayimg.com/images/g/qqUAAeSwGCFog3t2/s-...	0
8	Nike Tech Fleece Hoodie	Mens Size M — Grey, lightly worn	60.00	https://i.ebayimg.com/images/g/13cAAeSwqcRpQewH/s-...	1
9	Trapstar Shooters Jacket	Mens Size L — Black, good condition	140.00	https://i.ebayimg.com/images/g/dZsAAeSwZPJpSnVv/s-...	1
10	Palm Angels Track Pants	Size M — Orange stripes	180.00	https://i.ebayimg.com/images/g/BJ8AAeSwpO5pJnPj/s-...	1

5. Updated grabproducts function

The product-fetching function now checks the visibility flag and only returns products where visibility = 1.

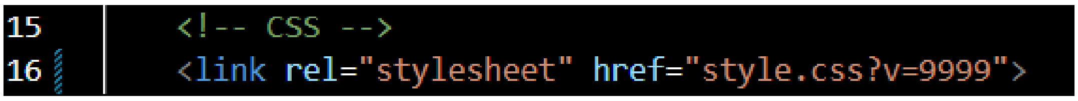


```php
    $stmt = $pdo->query("SELECT * FROM productstb WHERE expired=1");
```

Frontend & UI Fixes

6. Forced CSS refresh

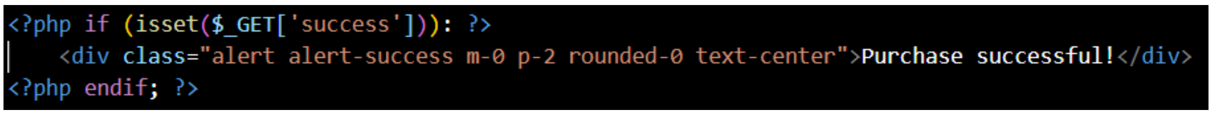
A browser caching issue was fixed by appending ?v=9999 to the stylesheet link.
This forces the browser to load the updated CSS instead of using the old cached version.



```
    <!-- CSS -->
    <link rel="stylesheet" href="style.css?v=9999">
```

7. Added thank-you notification bar after purchase

A Bootstrap notification bar was added to confirm the order was successful and improve the user experience instead of seeing a blank PHP screen.



```
<?php if (isset($_GET['success'])): ?>
    <div class="alert alert-success m-0 p-2 rounded-0 text-center">Purchase successful!</div>
<?php endif; ?>
```

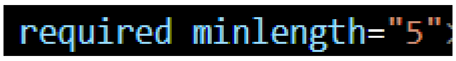
Security & Validation Improvements

8. Hashing deletion

I made the executive decision that hashing will not work as there is no need for excessive user data to be stored which will just increase security risks and DPA/GDPR legislation risks.

9. Input validation

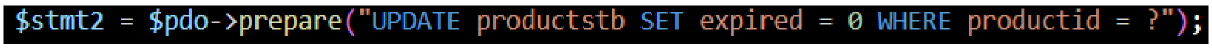
Started writing validation manually in PHP, but switched to Bootstrap's built-in validation system since it handles most cases automatically with no bugs.



```
required minlength="5"
```

10. SQL injection protection

All database queries now use prepared statements with ? placeholders. This prevents SQL injection and makes the backend more secure.



```
$stmt2 = $pdo->prepare("UPDATE productstb SET expired = 0 WHERE productid = ?");
```