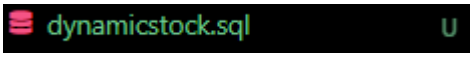
Live Backend Stock System Update Summary

SQL Implementation

I wrote the SQL based directly on the ERD Summary diagram found in the diagrams folder. This allows the database structure to match the planned relationships accurately and ensures that the tables work together in a reliable and scalable way. This can all be seen in dynamicstock.sql as I did an export dump on XAMPP.



File manager

Hashing Function for User Data

A hashing function was required to securely store user passwords. Initially, I used SHA256 however, after research I changed to using the password_hash in PHP for just the password as this method is more secure. For less sensitive data I will still use SHA256. This now makes the solution GDPR compliant. Also for maintainability I created hash.php which connects using the name and POST at the moment it does not send the hash anywhere just displays it as a test which will be done in the next version

```
13  function hashedItem($hash) {
14      return hash("sha256", $hash);
15  }
16
17  function hashPassword($password) {
18      return password_hash($password, PASSWORD_DEFAULT);
19  }
```

Hash.php

```
<form action="hash.php" method="POST">
```

index.php

CSS File Management

The CSS was previously in the main page, but to make the project more maintainable and professional, I moved the CSS into its own separate stylesheet file. In the next version I will create proper file management but for now this will do.

```
<!-- CSS File -->
<link rel="stylesheet" href="style.css">
```

index.php

File manager

Database Connection

A database connection was successfully established using PHP. This will allow the application to interact with user data, product information, and orders. The connection is tested and sends a message in the JS console with a success message

```php
<?php

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

try {
    $pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    echo "<script>console.log('Database connection: SUCCESS');</script>";

} catch (PDOException $e) {
    $errorMessage = addslashes($e->getMessage());
    echo "<script>console.error('Database connection FAILED: $errorMessage');</script>";
}

?>
```

db.php

```
<!-- Database Connection -->
<?php include "db.php"; ?>
```

index.php