Technical Report

ICTPRG537 – Implement security for applications

Carlos Camacho

10/11/2023

Table of Contents

[Task 1 – Implement basic Authentication 2](#_Toc105152928)

[Selected Application 2](#_Toc105152929)

[Viability Review 2](#_Toc105152930)

[Login Page Screenshot 2](#_Toc105152931)

[Secure Resource 2](#_Toc105152932)

[Authentication System 2](#_Toc105152933)

[Testing Authentication 2](#_Toc105152934)

[Task 2 – Plan and Research Advanced Security 3](#_Toc105152935)

[Selected Application 3](#_Toc105152936)

[Manager Email 3](#_Toc105152937)

[Target Deployment Platform 3](#_Toc105152938)

[Security Strategy Report 3](#_Toc105152939)

[Outcomes 3](#_Toc105152940)

[Authentication Strategy 3](#_Toc105152941)

[Authorised Access 3](#_Toc105152942)

[Protected Resources 4](#_Toc105152943)

[Authorisation Strategy 4](#_Toc105152944)

[Cryptographic Algorithms 4](#_Toc105152945)

[Input/Output Handling 4](#_Toc105152946)

[Security Strategy Meeting 4](#_Toc105152947)

[Task 3 – Implement Security Strategy 5](#_Toc105152948)

[Target Environment Configuration 5](#_Toc105152949)

[Task 4 – Review and Test Security 6](#_Toc105152950)

[Testing Report 6](#_Toc105152951)

[Access to protected resources when Authenticated 6](#_Toc105152952)

[Access to protected resources when not Authenticated 6](#_Toc105152953)

[Access to protected resources when Authorised 6](#_Toc105152954)

[Access to protected resources when not Authorised 6](#_Toc105152955)

[Accessing contents of encrypted data 6](#_Toc105152956)

[Accessing contents of and ensuring validity of decrypted data 6](#_Toc105152957)

[SQL Injection on at least 2 endpoints: 6](#_Toc105152958)

[Script Injection / Execution on at least one Form/Page/View 6](#_Toc105152959)

[Testing Report Summary 7](#_Toc105152960)

# Task 1 – Implement basic Authentication

## Selected Application

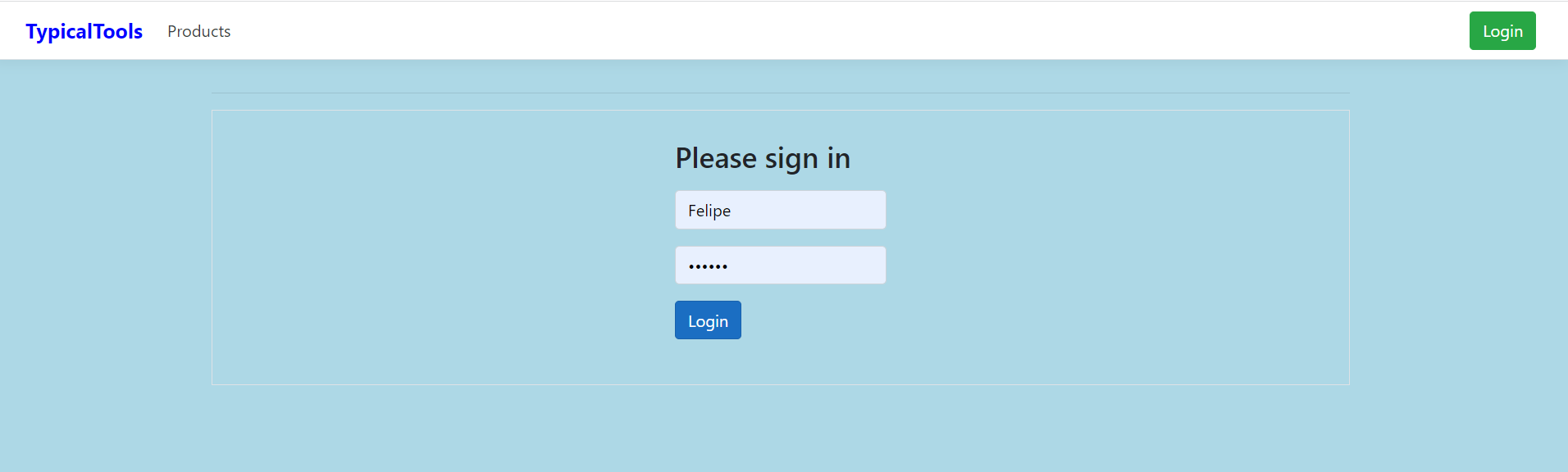
Typical Tech Tools

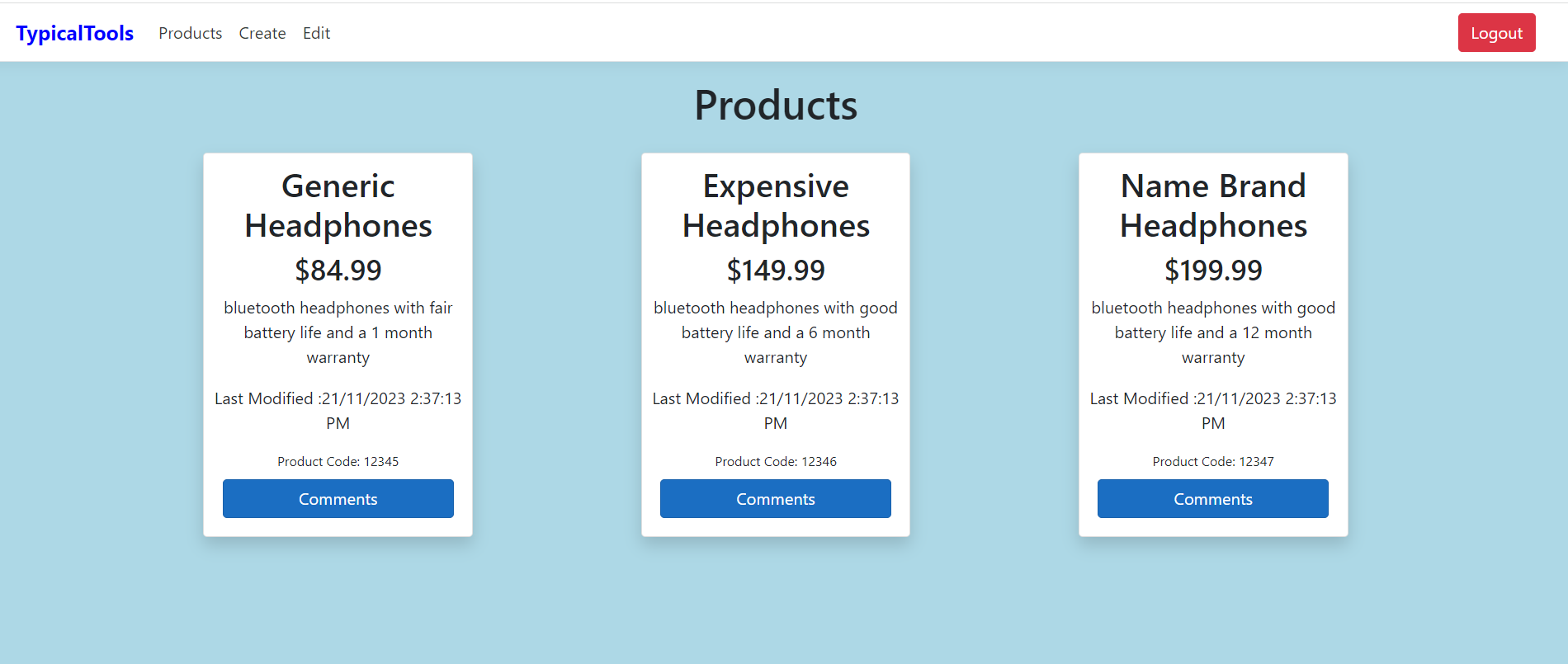
Typical Tech Tools solution is a software solution that offers basic authentication and authorisation techniques. Users must log in to access certain services, and certain operations can only be performed by authorised administrators. However, the current authentication and authorisation techniques are insecure, necessitating an update to a more secure solution.

## Viability Review

The benefits of integrating authentication and authorisation systems in this application during this discussion. The meeting concluded that the chosen application is an appropriate alternative for increasing security.

## Login Page Screenshot



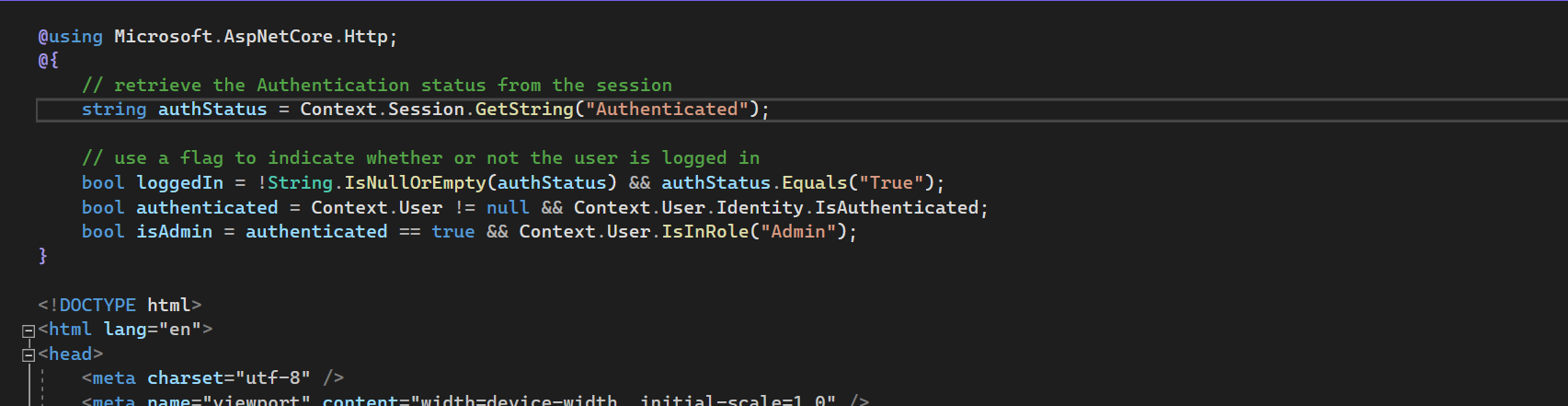


## Secure Resource

The ability to create new items and modify existing ones was the first resource protected in this programme. Only users with the admin role will have access to and control over these resources. This is not available to ordinary users.

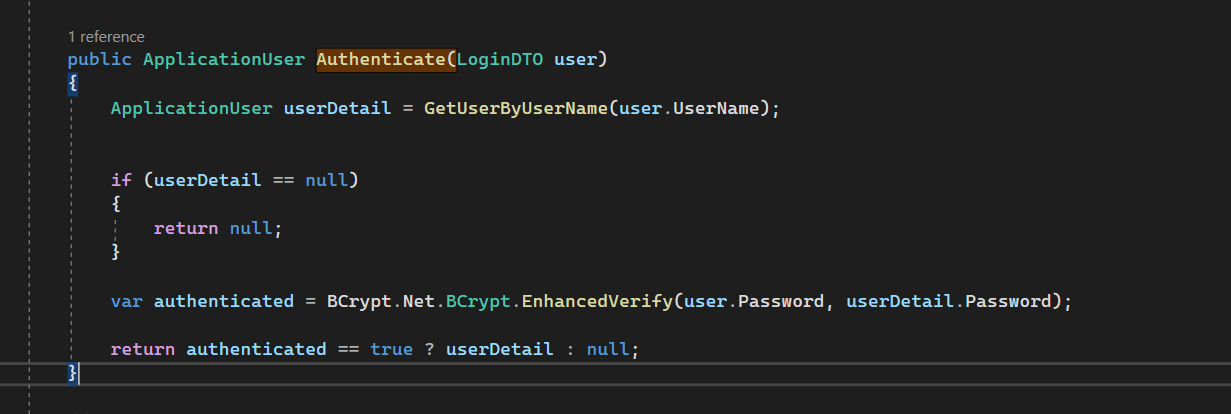
The application's second protected resource is the ability to make, alter, and delete comments. All logged-in users will be able to add a new remark, change an existing comment they created in the previous session, and remove a comment they created in the previous session for a product.

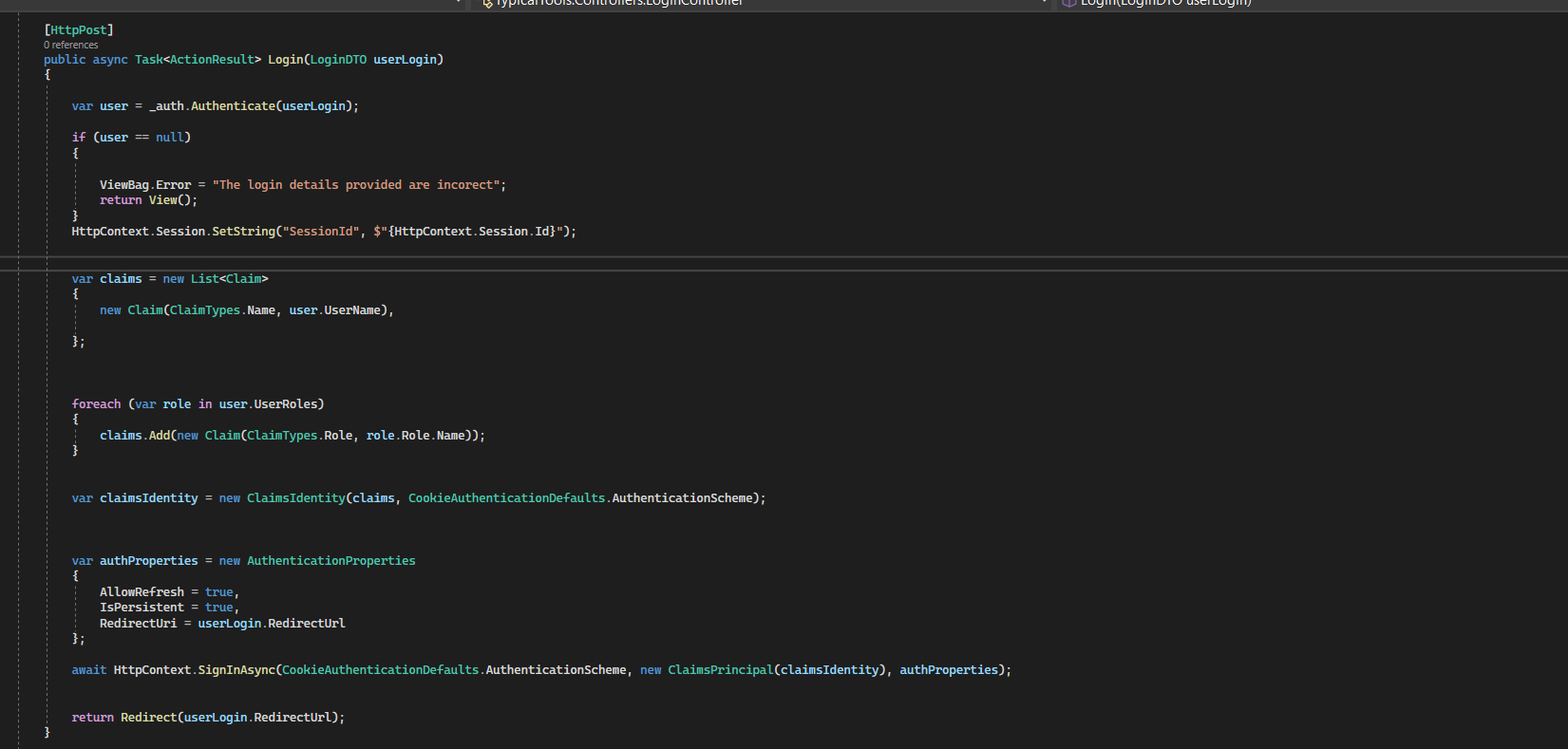
## Authentication System



A screenshot of a computer program

Description automatically generated





## Testing Authentication

A screenshot of a computer

Description automatically generated

A screenshot of a login form

Description automatically generated

Task 2 – Plan and Research Advanced Security

## Selected Application

Sam's Warehouse has been chosen as the application for implementing sophisticated security. This programme would benefit immensely from better security, particularly if it allowed users to upload encrypted receipt images and restricted access to specific resources.

## Manager Email

To: <projectmanager@UptownIT.com.au>

From: <carloscamacho@UptownIT.com.au>

CC:

BCC:

Date: 15/11/2023

Subject: Application of choice

-------------------------------------------------------------

Hello, Project Manager.

Sam's Warehouse was chosen as the application to add sophisticated security.

This application satisfies the following requirements for advanced security to be used:

1. Authorisation and authentication will be beneficial. This would help to limit user permissions and access to private resources.

2. Encryption and decryption of sensitive data. This will help to protect the information if an unauthorised user gains access.

3. Secure user input and output management. Sanitisation and validation will clean the input of harmful information and evaluate user input to ensure only properly structured data is entered.

I am also asking all organisational paperwork pertaining to the application's security.

Regards,

Carlos Camacho

## Target Deployment Platform

Because the web application was designed to work with web browsers, the target system for deployment is a PC running Windows 10 or 11 with a web browser installed. The application is incompatible with Internet Explorer; instead, use Firefox, Chrome, or Edge. To improve the application's security, each user will be assigned a role, with each role granting the user access to particular resources.

We will limit access to the data to particular users. Some users, such as admin, will have complete permissions, while others will only have read and write privileges, which are configured in the database. Admin users will have higher powers, allowing them to create, delete, and amend user data, as well as create new goods. Normal users will not have access to these resources; however, they will be able to construct a shopping list, change which products appear in the list, and delete things from the list.

To do this, security configurations for application users will be created within the database management service to limit user capabilities. The roles will be created within the database, with read and write capabilities restricted. We will be able to test the users configured in the database using IIS (internet information services).

## Security Strategy Report

### Outcomes

The results of adding Authentication and Authorisation to the application.

The result of adding further security is that undesirable data cannot be added to the database, and the data that is currently stored may only be altered or deleted by admin users.

In addition to the ability to create new goods, admin users will have access to enhanced features that allow them to create, delete, and modify user data. Normal users will be able to make shopping lists, edit which commodities appear in the lists, and remove items from lists, but they will not be able to access these resources.

A non-logged in user will be able to see the application's about us and login pages.

### Authentication Strategy

The authentication method used is a cookie-based authentication. When a user enters their email address and password and successfully signs into the application, they are given an authenticated cookie. This cookie will expire after 300 seconds on a sliding scale. With each request, this cookie will be exchanged between the user and the server.

### Authorised Access

How Authorisation will be used to protect resources from access.

Roles and claims-based authorisation will be used to prevent specified resources from being accessed. Only admin users will be allowed to create new goods, create new users, edit current users, and delete users with this authorisation. A regular user will be unable to access those resources. A regular user can create a new shopping list, add things to their own shopping lists, and remove products from their own shopping lists.

### Protected Resources

* All delete, put, and post endpoints for goods and users are protected with the authorise tag, which only permits admin access. All logged in users can use the get endpoint for products, as well as the post and delete endpoints for shopping lists. This is accomplished by employing the approved tag.

• Products (for all users).

• Make a new product (admin only)

• Shopping list (for all users)

• Users (administrators only)

• Add a new user (admin only)

• Edit a user (admins only)

• Remove a user (admin only)

### Authorisation Strategy

Roles are utilised for authorisation: "Admin" and "User"

To protect some resources from unauthorised access, roles and claims-based permission will be used. Only admin users will be allowed to add new users with the role of "User" automatically, update current users, add new goods, and delete users with this authorisation. Such resources will be inaccessible to the average user. A ordinary user can create new shopping lists, add things to existing shopping lists, and delete items from existing shopping lists.

### Cryptographic Algorithms

To process images with 128-bit blocks, the Advanced Encryption Standard (AES) approach employs three different cypher key sizes with lengths of 128, 192, or 256 bits. Depending on the key size length used, the method goes through 10, 12, or 14 execution cycles. The suggested system has a block size of 128 bits and a key size of 256 bits. The method is used to both decode and encrypt images. Because the key size is 256 bits, it will take 14 cycles.

The encryption technique utilised involves the use of an encryption key, and in order to decode a file, the same key that was used to encrypt it must be used. If the encrypted files are read by unauthorised individuals, they will be protected against dangerous exploitation.

Encrypting sensitive photos is an effective means of safeguarding them from attackers. The client will be uploading receipts, which may contain personal information. We are compelled to employ an encryption algorithm to protect the data because the photographs contain personal information.

### Input/Output Handling

The application will use measures like parameterised query to promote input sanitization to prevent SQL injection.

Both sanitization and model binding are used in this application. HTML sanitization creates a new HTML document by combing through an HTML page and only maintaining the "safe" and desired tags. By cleaning up any HTML code provided by a user, HTML sanitization can be used to prevent XSS attacks. Model binding, as opposed to HTTP requests, allows controller activities to interact directly with model types (provided as method parameters).

While performing client-side validation, I use the model's data annotations to specify the required minimum length. I've established a minimum and maximum length for user input in the views to ensure that it isn't too short or too long for the application to handle.

## Security Strategy Meeting

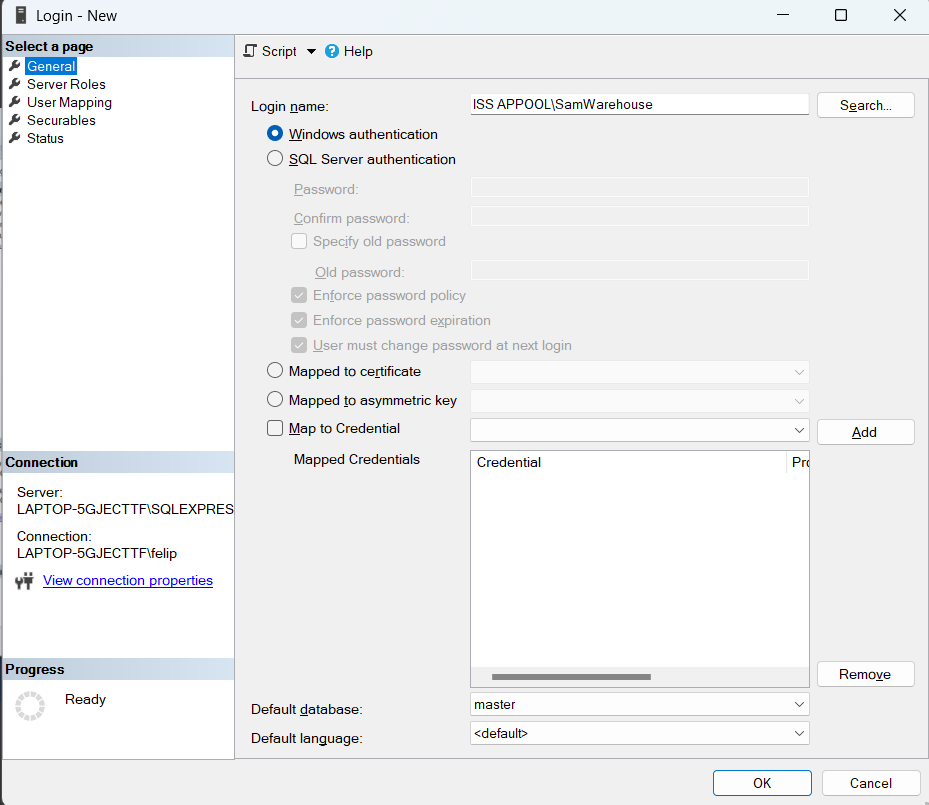
Organise a meeting with your assessor to present a summary of the proposed security strategy, record any feedback or changes required to gain approval.

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting Purpose: | Security Strategy Review | | |
| **Manager Name:** | Manager | | |
| **Developer Name:** | Carlos Camacho | | |
| **Date:** | 17/11/2023 | | |
| **Feedback:** **Manager provided feedback on the viability of the proposed enhancements.**  **Instead of showing on screen shops, make a description about how it is going to work.** | | | |
|  | | | |
| Approval Received? | Y / N | |  |
| Student | | Student | |
| Developer | | Signature | |
| Assessor | | Assessor | |
| Manager | | Signature | |

# Task 3 – Implement Security Strategy

## Target Environment Configuration

To do this, security configurations for application users will be created within the database management service to limit user capabilities. The roles will be created within the database, with read and write capabilities restricted. We will be able to test the users configured in the database using IIS (internet information services). To restrict a user's privileges and permissions, utilise the Windows file explorer.

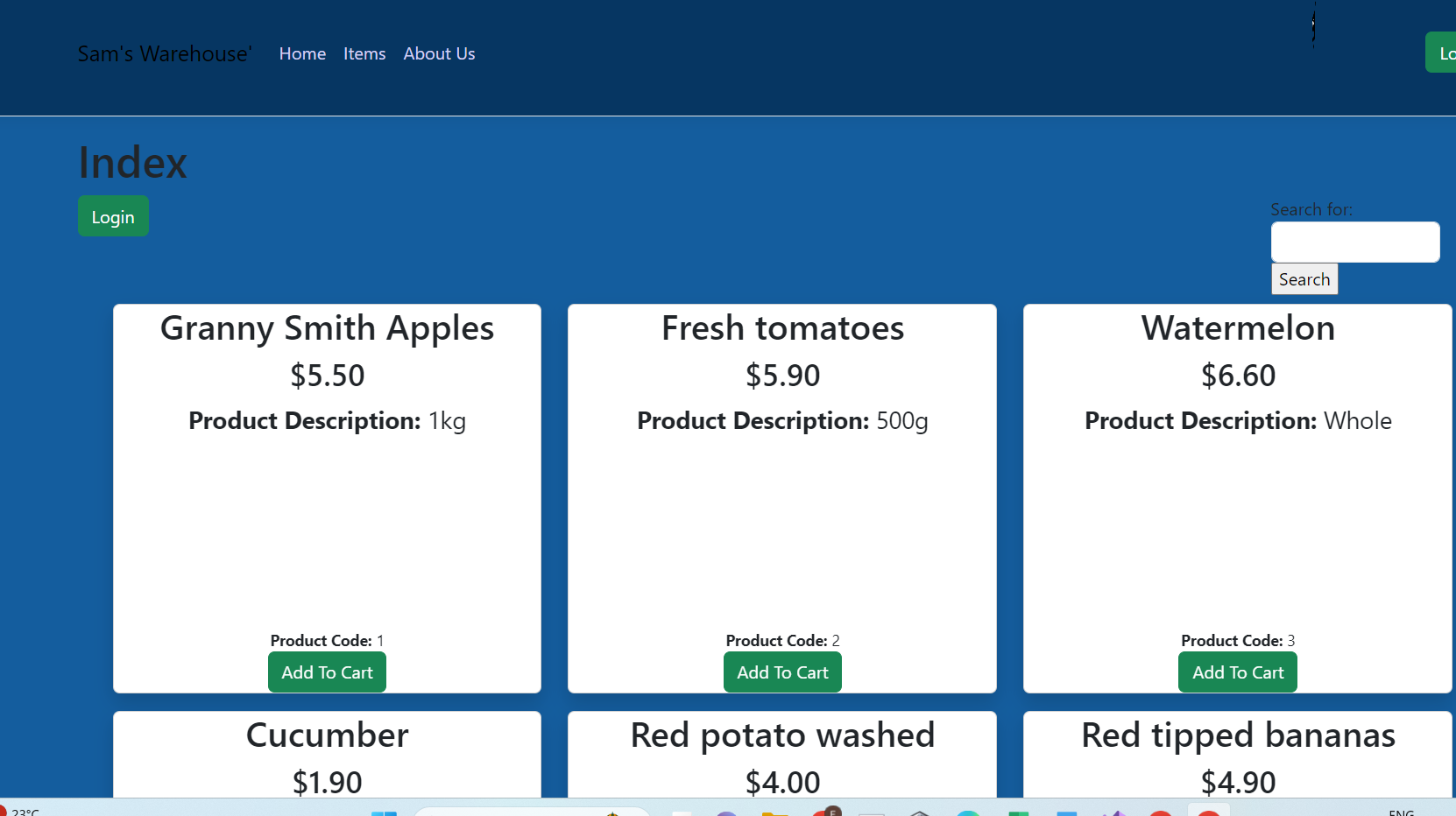


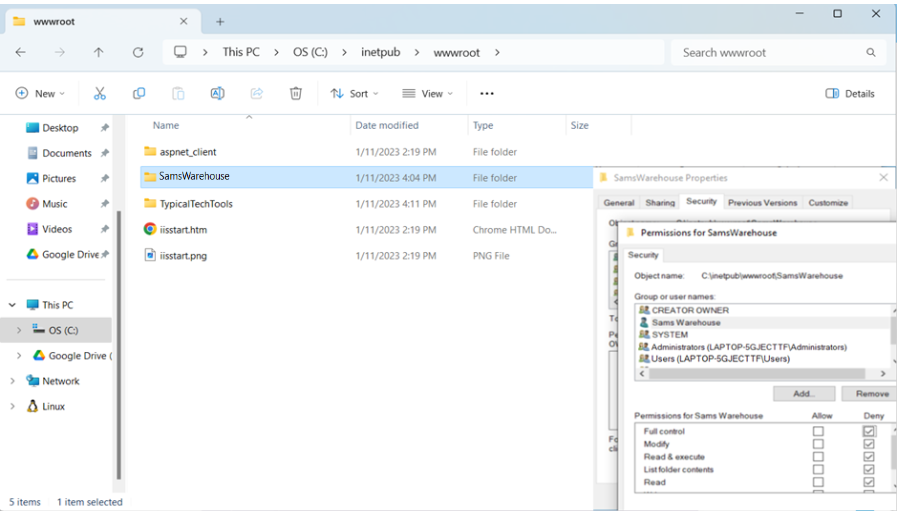
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated





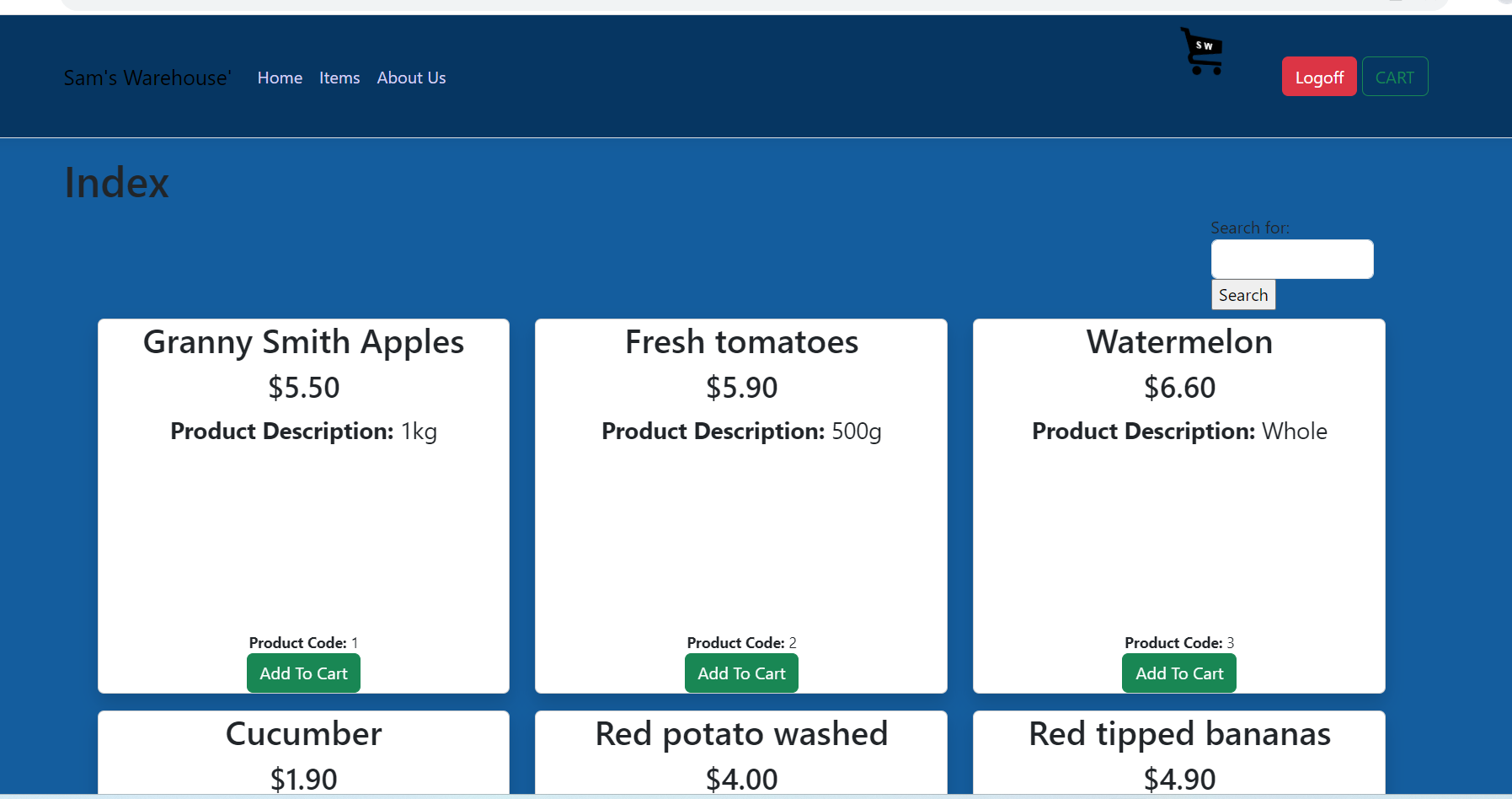
****

# Task 4 – Review and Test Security

## Testing Report

### Access to protected resources when Authenticated

The user is able to access the protected resource when authenticated



### Access to protected resources when not Authenticated

If a user is not authenticated and attempts to access protected resources, they are sent to the login page.

A blue rectangular object with white text

Description automatically generated

### Access to protected resources when Authorised

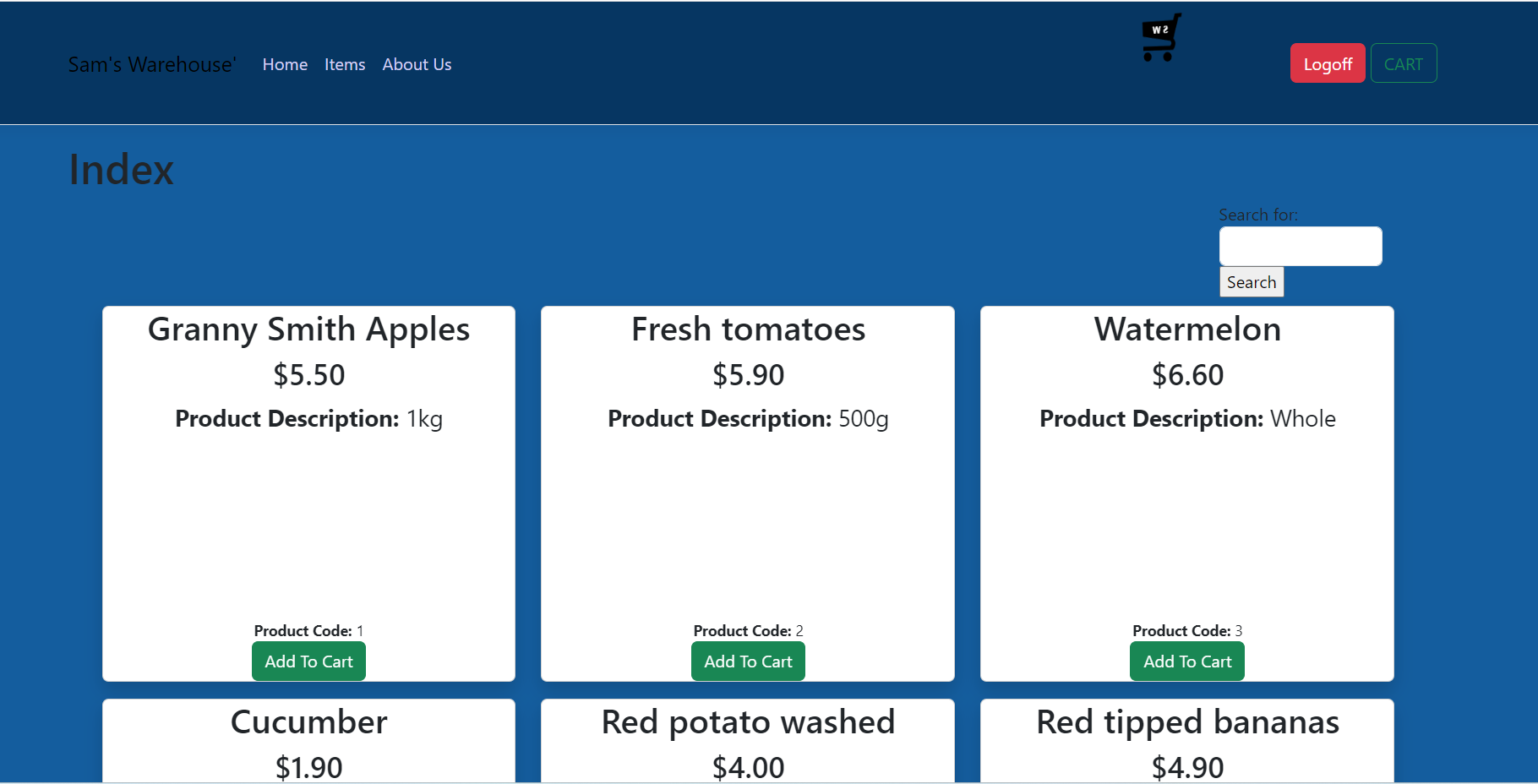
The authorised user has access to the users page.

A blue and white rectangle with a white stripe

Description automatically generated

### Access to protected resources when not Authorised

When an unauthorised user attempts to access a protected resource, they are routed to the previous page.

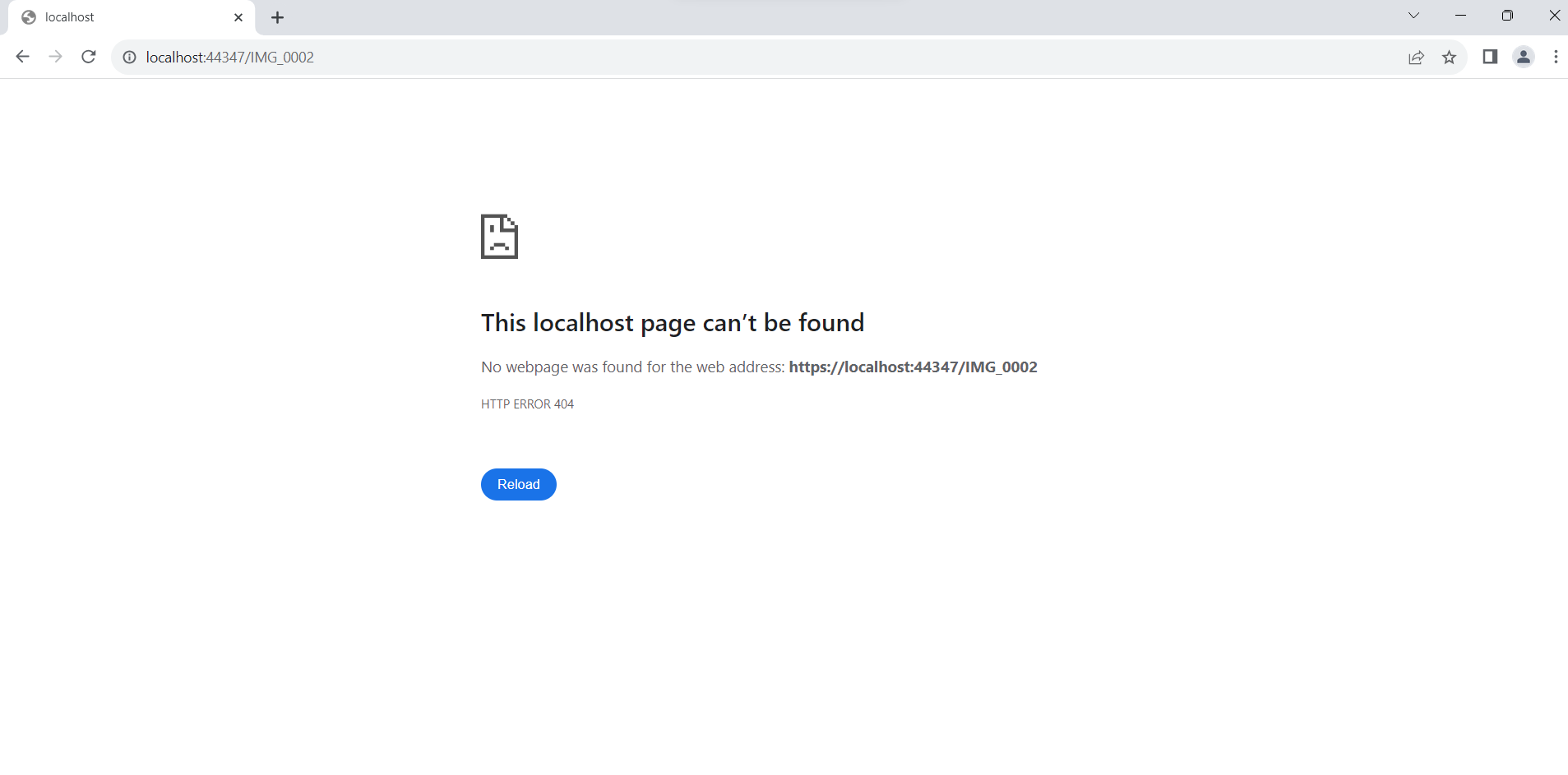


### Accessing contents of encrypted data

* Outcome

When attempting to examine the encrypted data, it was not possible.

* Screenshot

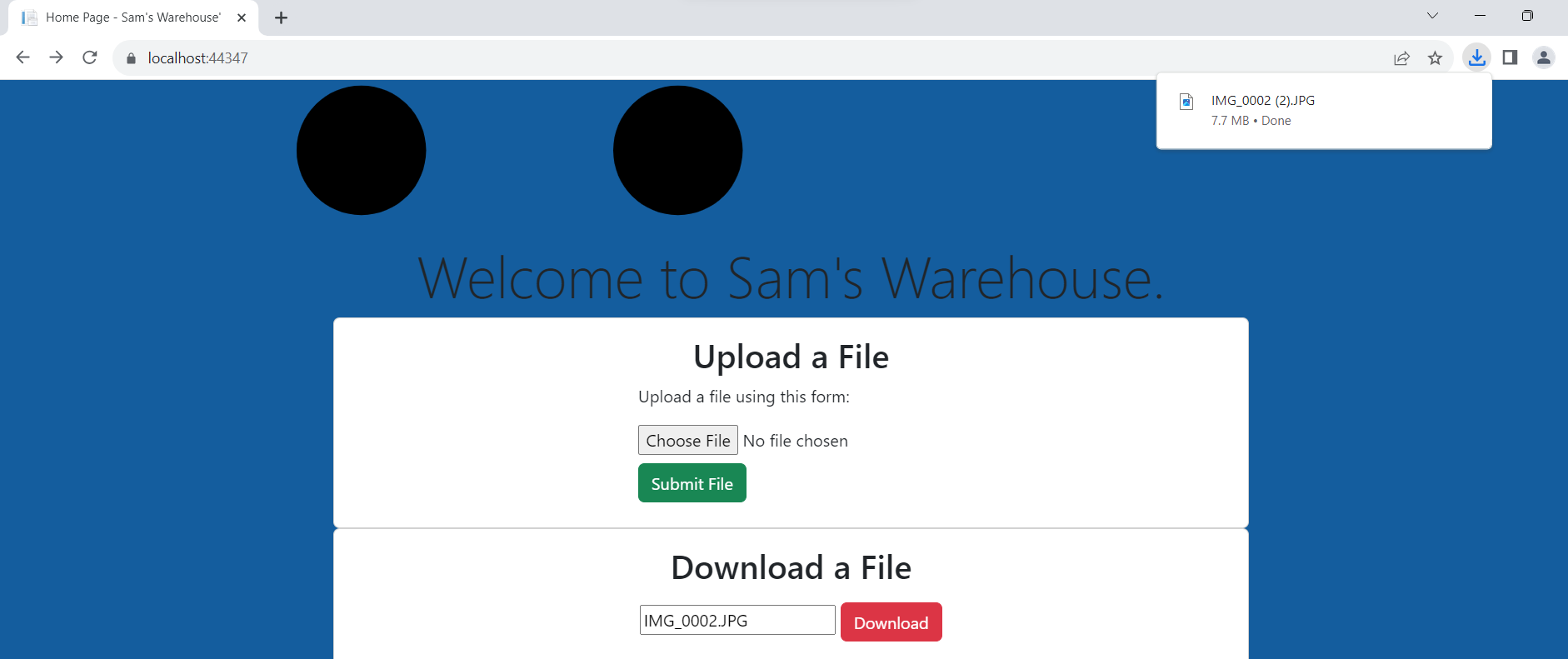


### Accessing contents of and ensuring validity of decrypted data

* Outcome

The decrypted data is able to be viewed.

* Screenshot

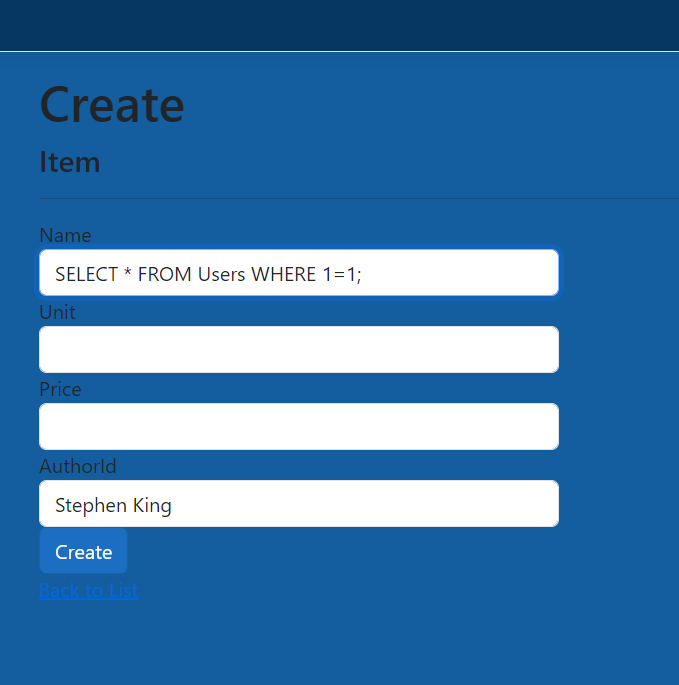


### SQL Injection on at least 2 endpoints:

* Outcome

Instead of a command, the SQL injection will be interpreted as a string. When the create button was pressed, the result was saved as a string.

* Screenshot



A screenshot of a computer

Description automatically generated

A blue screen with white text

Description automatically generated

A screenshot of a login screen

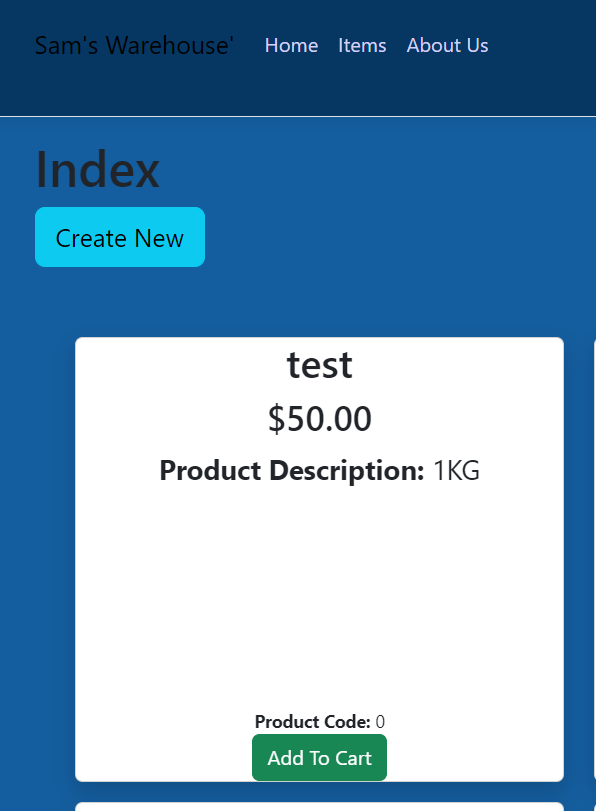
Description automatically generated

### Script Injection / Execution on at least one Form/Page/View

Outcome

The script injection will be removed from the input field and never utilised again.





**Search outcome:**

## Testing Report Summary

Provide a summary of the outcomes of implementing and testing the security strategy, including any issues identified in the process.

Browser developer tools will be used as testing tools. Firefox is the most popular browser, and it comes with a collection of web development tools called Firefox Developer Tools. They can inspect, modify, and troubleshoot HTML, CSS, and JavaScript code.

To assess an application's security, a free, open-source penetration testing tool called Zed Attack Proxy (ZAP) is utilised. ZAP functions primarily as a "man-in-the-middle proxy," as the term is known. It stands between the tester's browser and the web application in order to intercept and analyse communications sent between the tester's browser and the web application, modify the contents as needed, and then send those packets on to their intended recipient.

Entity Framework, in general, uses parametrized LINQ-to-SQL queries and is immune to traditional SQL Injection attacks.

HtmlSanitizer, a.NET package, is used to eliminate HTML constructions from documents and fragments that could be utilised in XSS attacks. It can be highly modified, for example, by specifying which objects should be removed. Because HtmlSanitizer is based on a powerful HTML parser, it may also protect you from "tag poisoning," a deliberate or unintentional process in which incorrect HTML in one fragment spoils the entire page and creates layout or style issues.

To ensure the application's security, authorisation and authentication were used to restrict access to the application's protected resources. To protect the sensitive information contained in the users' receipts, encryption and decryption have been used. By using parametrized LINQ-to-SQL queries, the entity framework safeguarded the application from SQL injections. Sanitisation was utilised to sanitise all script user inputs to prevent script injections. Html sanitiser is the sanitisation package.