**TECHNICAL REPORT**

Shopping List Application

Student Name: Benjamin Moore

Date: 28/05/24

[**Part 1 Program and Design Specification 3**](#_heading=h.gjdgxs)

[Program Specifications Report 3](#_heading=h.30j0zll)

[Application Logic Requirements 3](#_heading=h.1t3h5sf)

[User Interface 3](#_heading=h.17dp8vu)

[User Interface Sketches 4](#_heading=h.lnxbz9)

[Colour Scheme 4](#_heading=h.35nkun2)

[Design Scheme 4](#_heading=h.44sinio)

[**Part 2 User Interface Design Plan 5**](#_heading=h.2jxsxqh)

[Email to the client 5](#_heading=h.z337ya)

[Feedback from the Client/Teacher 5](#_heading=h.3j2qqm3)

[User Interface Plan 5](#_heading=h.1y810tw)

[UI sketch for each planned page or form 5](#_heading=h.4i7ojhp)

[User Experience (UX) flowchart 5](#_heading=h.2xcytpi)

[Interaction Design Patterns 5](#_heading=h.1ci93xb)

[Client-Side validation techniques that will be implemented 5](#_heading=h.3whwml4)

[Plans for providing user interface customisation 5](#_heading=h.2bn6wsx)

[Graphical and Multimedia requirements and planned implementation 5](#_heading=h.qsh70q)

[**Part 3 Application Development 6**](#_heading=h.3as4poj)

[Screenshots 6](#_heading=h.1pxezwc)

[**Part 4 Testing + Validation 7**](#_heading=h.49x2ik5)

[Testing Table 7](#_heading=h.2p2csry)

[Testing Summary (including testing tools used) 7](#_heading=h.147n2zr)

[Testing Screenshots 7](#_heading=h.3o7alnk)

[Model, View + Controller connection Screenshots 7](#_heading=h.23ckvvd)

[Client Email demonstrating developed application 8](#_heading=h.ihv636)

[Feedback from the Client/Teacher 8](#_heading=h.32hioqz)

[Adjustments made to the application to suit feedback 8](#_heading=h.1hmsyys)

[**Part 5 Handoff 9**](#_heading=h.2grqrue)

[Client Email 9](#_heading=h.vx1227)

[Sign off sheet 9](#_heading=h.3fwokq0)

# Part 1 Program and Design Specification

## Program Specifications Report

### Technology

The program will be developed to the technical standards laid out in the scenario. The application will be built in C# using the ASP.NET MVC (Model, View, Controller) framework. The Database used will be a local MSSQL database. The application will have the inbuilt functionality to populate the database if not already populated, The database will be creatable through an included PowerShell script.

### Development Environment

The application will be developed in Visual Studio 2022 using the MS SQL Server Management Studio to manage SQL statements and the development of the database. The application will be developed using C# and the C# .NET tools to ensure that the application compiles and debugs correctly.

### Testing Tools

The tools used are the native tools that come with Visual Studio, these include breakpoints. watches and error outputs, in code testing, will also be in place in the form of client-side validation where entries of unsanitized information are required. The Google web debugging tools are also used to see how the code is rendered and operates on the client side.

### Application Frameworks or templates

### This application uses .Net version 8.0.4. It runs with the microsoft SQL Connection library for connection to the database. There was no template used, it was built from scratch.

### Operating System and Platform

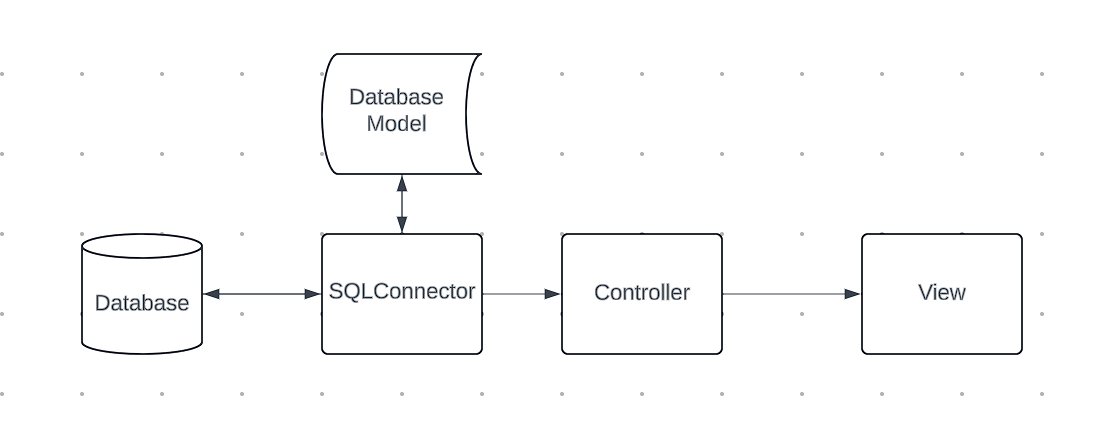
This application as it is a MVC web app will run on any device with a browser. To host the application however a windows 10+ or windows server 2022+ licenced device will be required.

## Application Logic Requirements

### Application Input + Output

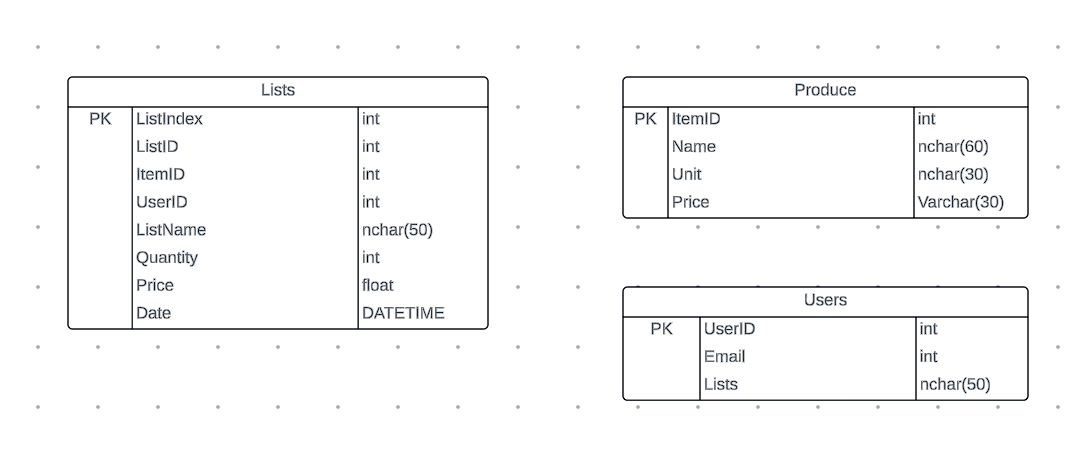
The shopping list application requires input and output from three tables, one to manage users, one for the produce in the shopfront and one for the users lists. These tables require no more than their own primary keys with auto incrementation.

The main functional inputs and outputs are as follows; Create List, Delete List, Add Item, Remove Item, Login, logout and user preferences in the form of dark mode, these wil be implemented in a logical approach shown in the following diagram.



Data Into the application will come from the MSSQL Database, it will then be processed by methods in the SQLConnector class that will create data models to return to the controller to then be processed into the view.

### Data Model for Application

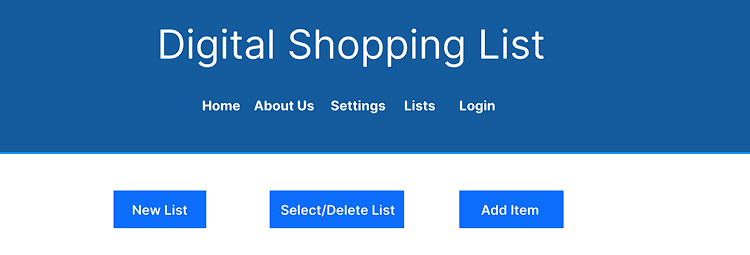


## 

## User Interface

### Navigation methods

When designing the navigation I wanted to ensure that at all times the user can access any needed links. this will be implemented through general links in the header that will be visible at all times and then on many of the pages there are buttons specific to the function. this can be seen in this sketch.



### User Interface Sketches

| **Screen 1 sketch** | **Screen 2 sketch** |
| --- | --- |
| **Screen 3 sketch** | **Screen 4 sketch** |

### Colour Scheme

| **Colour 1** | #155c9e |
| --- | --- |
| **Colour 2** | #0c4f6c |
| **Colour 3** | #b8cdf8 |

### 

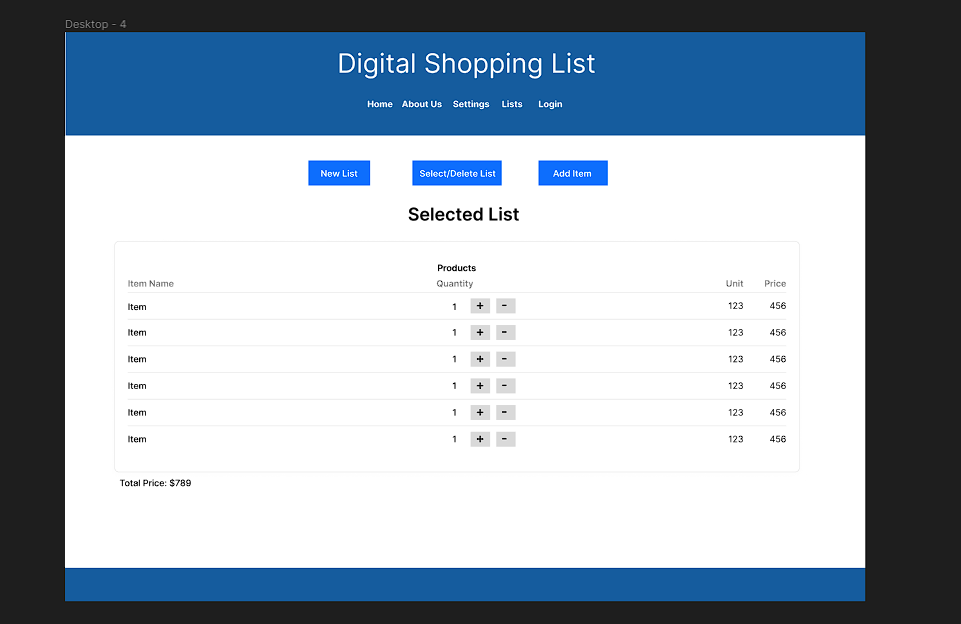
### 

### 

### 

### Design Scheme

The design follows a layout scheme with page specific content in the main content secion, this can be seen here.



All pages have the same links at the top for quick navigation and the page specific buttons are at the top and easily read to ensure maximum accessibility.

# Part 2 User Interface Design Plan

## Email to the client

From: ben@uptownit.com

Monday 08/05 12:34:56 PM

To: james.jones@shop.com

Hi James,

I have created the user interface design plan and would like to hear your thoughts on the design and if there are any improvements that can be made. We have tried to keep the design within the requirements laid out in prior communication. Please see the design plan attached.

Regards,

Benjamin Moore

Software Developer

Uptown IT

## Feedback from the Client/Teacher

From: ben@uptownit.com

Monday 08/05 12:34:56 PM

To: james.jones@shop.com

Hi Ben,

Thankyou for your design plan, overall we are happy with the design however we want to make sure that the user can see if they are logged in and what cart is selected when they are on the home page, please update the plan to include this change. We are otherwise happy with the proposed design and change being taken into development.

Regards,

James Jones

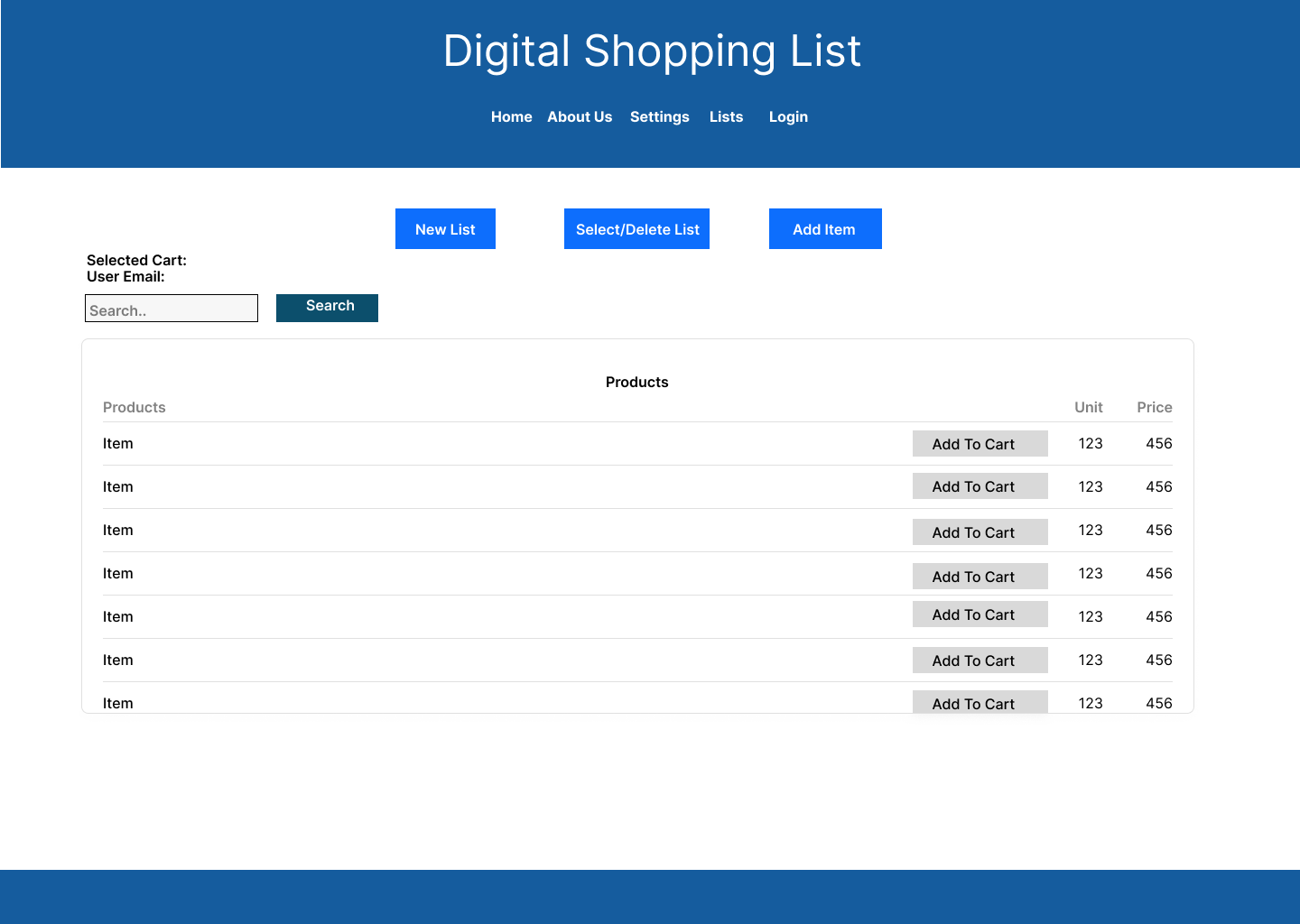
General Manager

Miscellaneous Shop INC

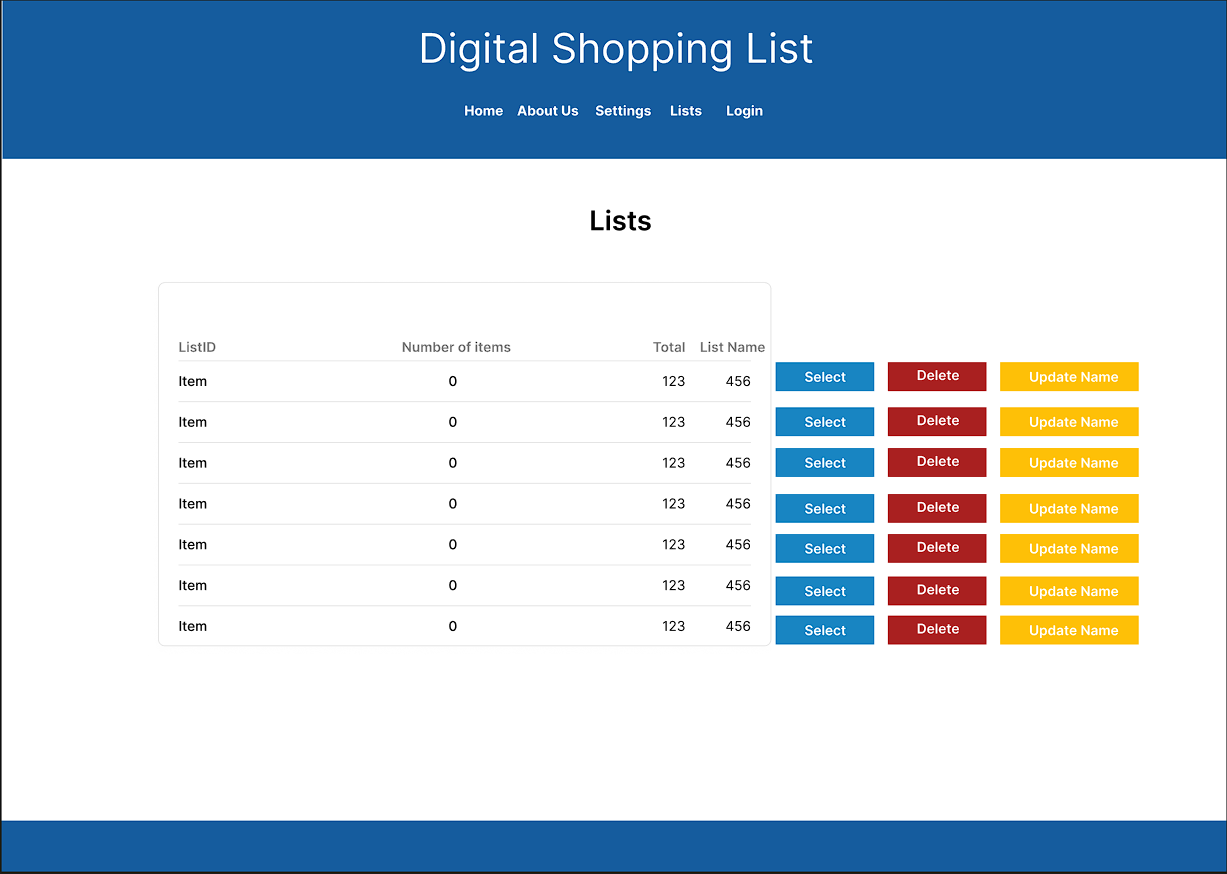
## User Interface Plan

### UI sketch for each planned page or form

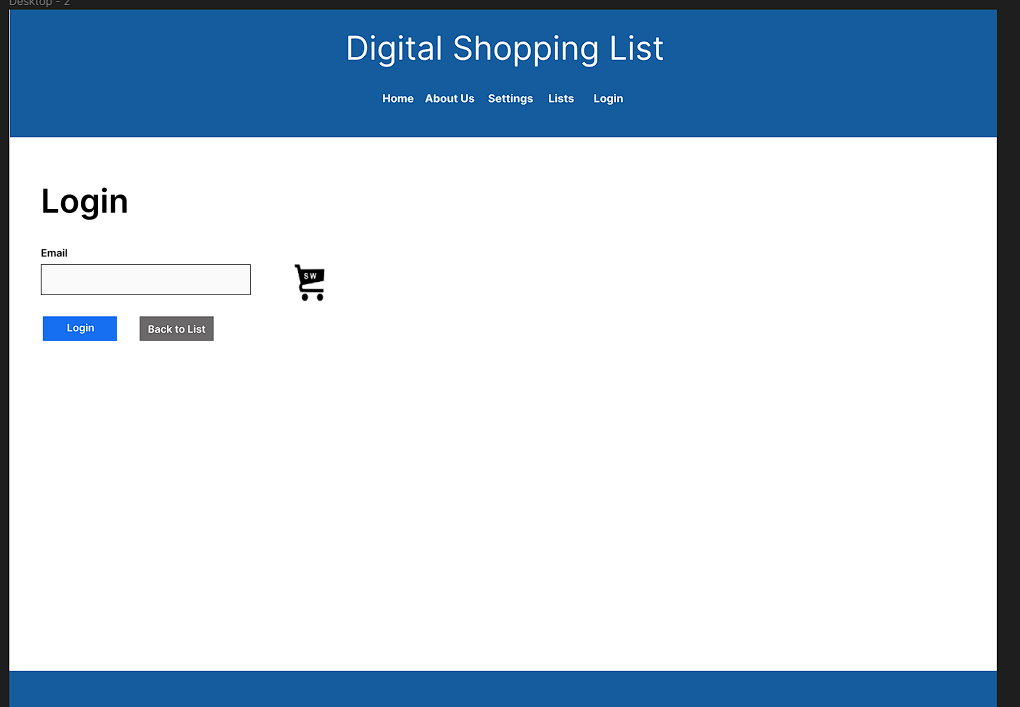
Home



Lists



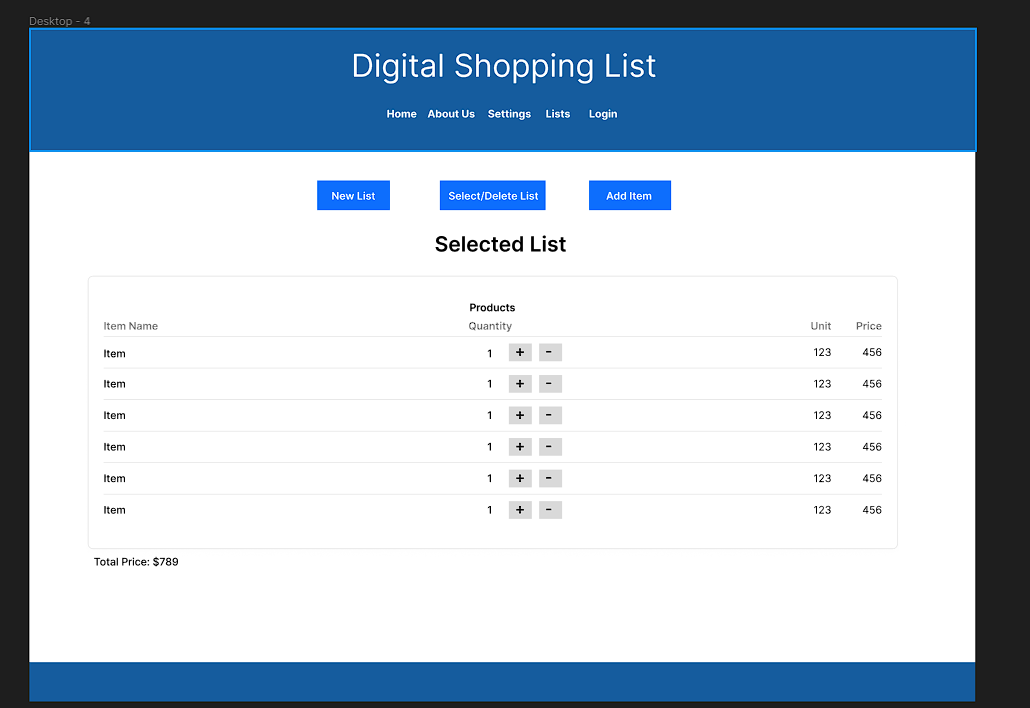
Login



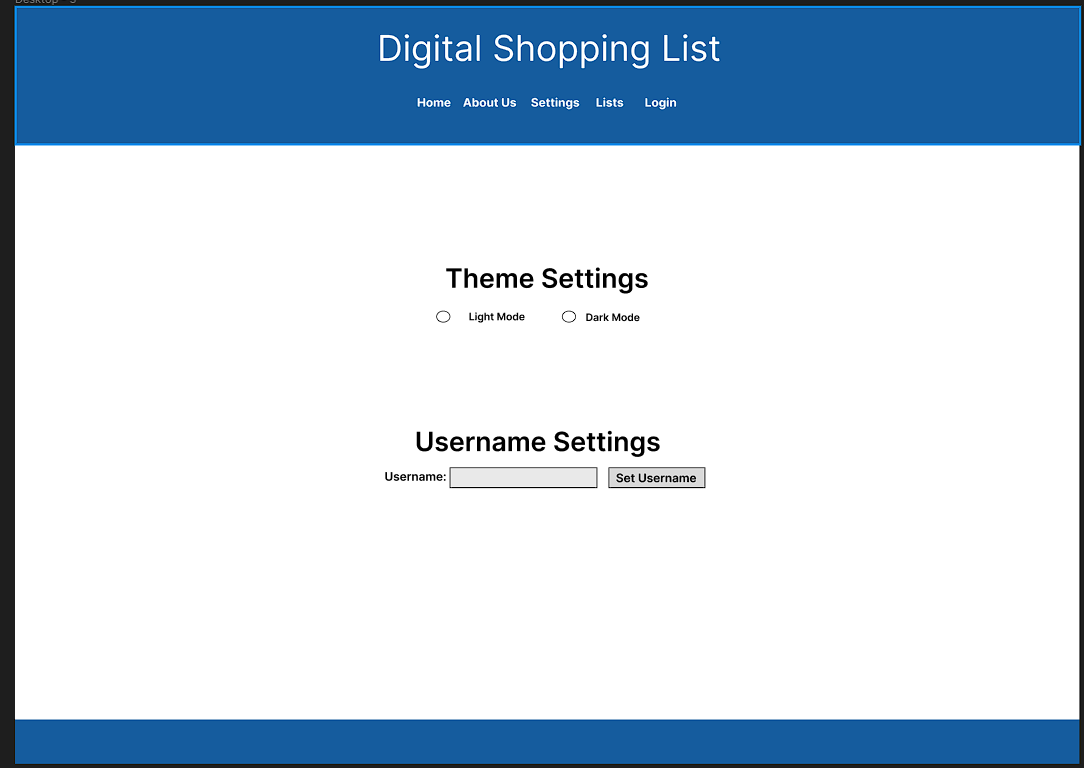
Logout



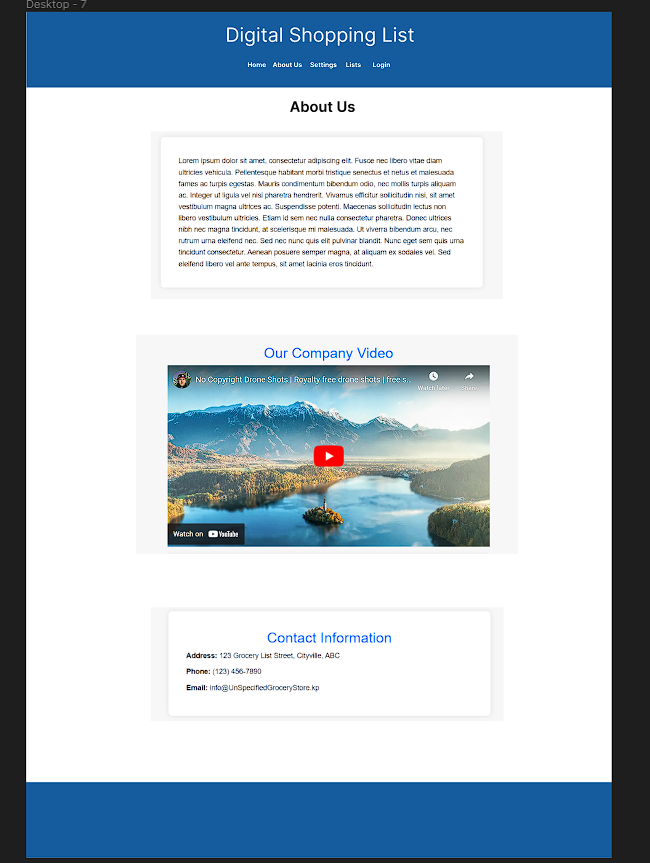
List View



Settings

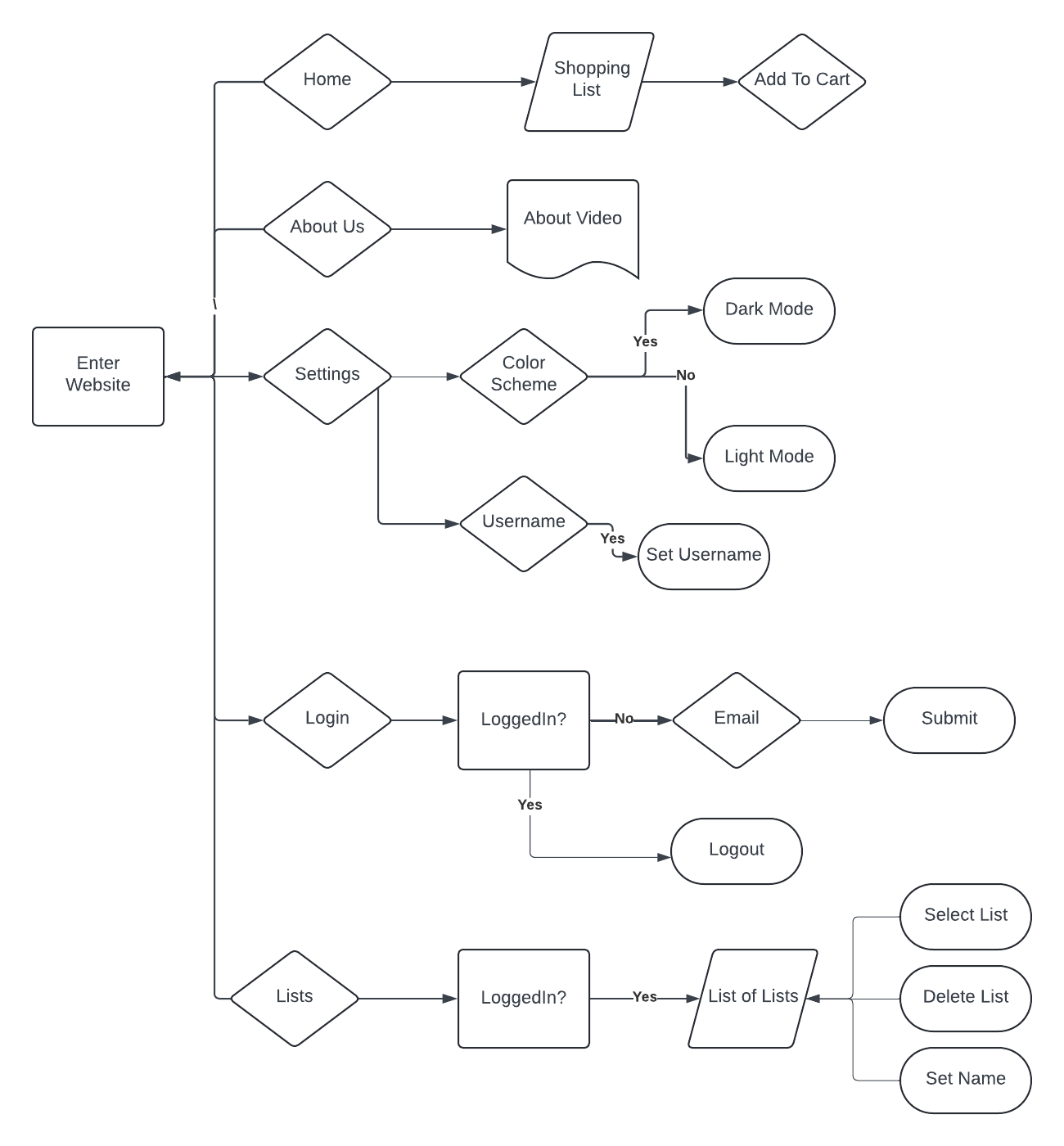


### About Us



### 

### User Experience (UX) flowchart



### 

### 

### 

### 

### 

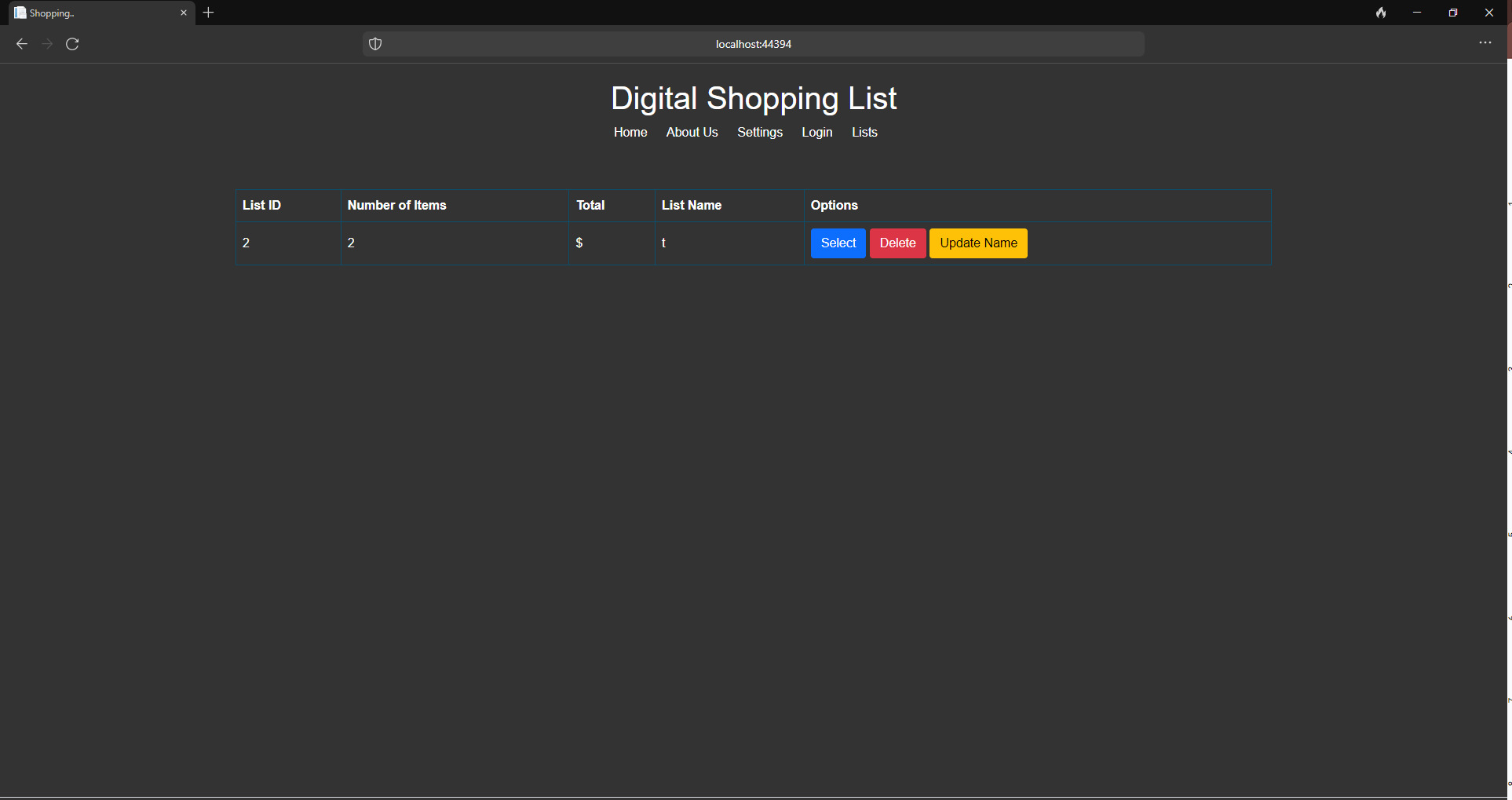
### 

### Interaction Design Patterns

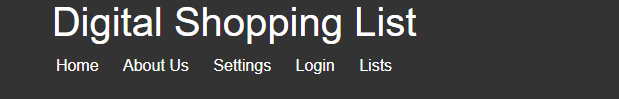
Throughout the development of the user interface, two major interaction design patterns were followed, this was to ensure the usability of the application to the target audience which is broad in age range and computer literacy, The design patterns used are as follows.

***Progressive Disclosure***

Throughout the pages of the application only the necessary information is included, the website was designed in this modular fashion to ensure that when visitors are using the site they can quickly and without confusion navigate to the required page. This can be seen in the following screenshots,



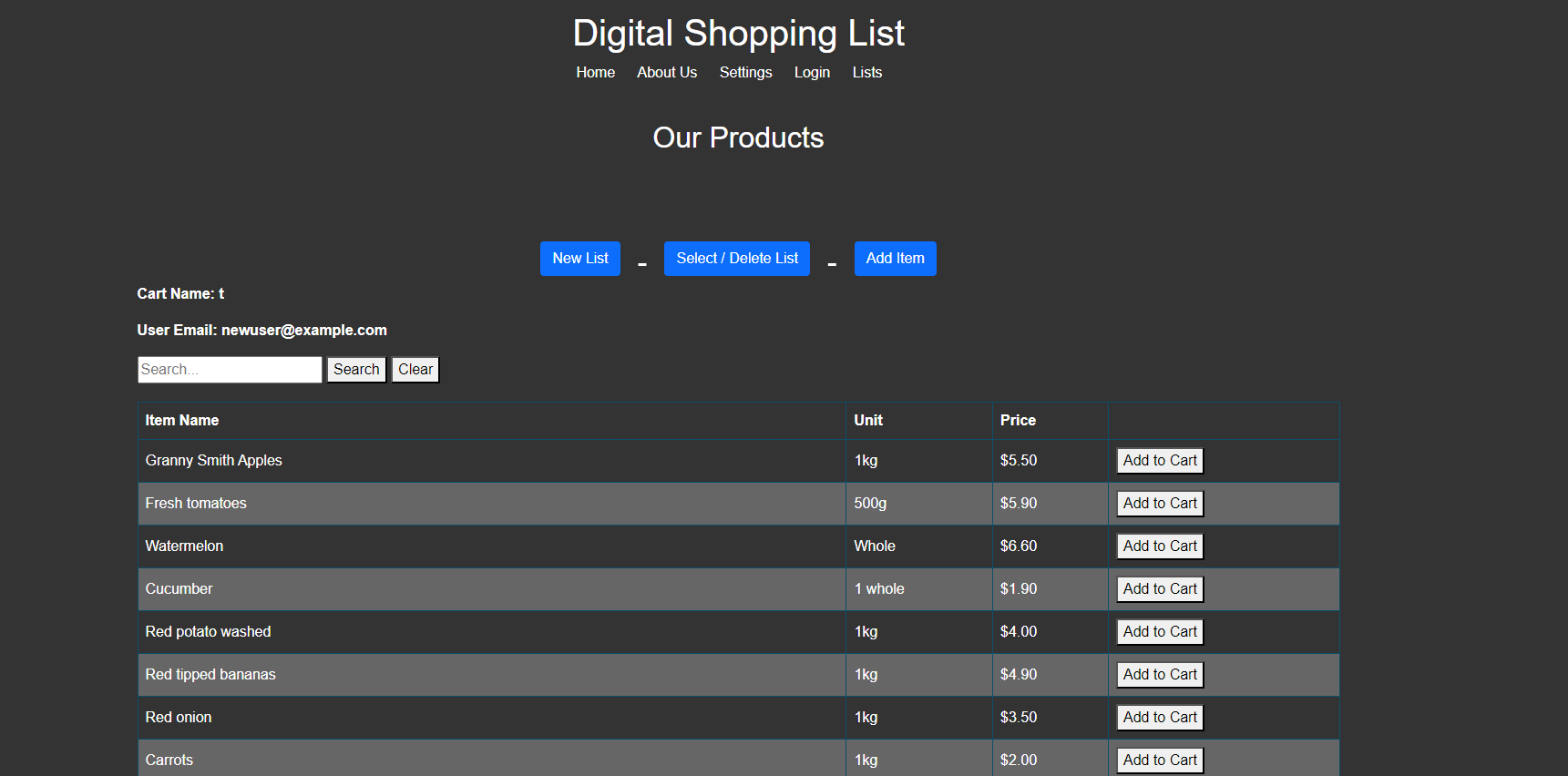
Nothing other than the required functions on the lists page.



Layout menu on each page in the same location ensuring consistency.

***Simplistic Layout***

Utilising the Progressive disclosure the website was designed to ensure that there were no more interface components than absolutely necessary to perform the functions of the site in order not to confuse or overwhelm more vulnerable consumers. This can be seen in the following screenshots,



### Client-Side validation techniques that will be implemented

Client side validation is implemented any time a raw user input is required. the client side validation will ensure its not null or an parameter that would be incorrect in the context, i.e data type.

### Plans for providing user interface customisation

The main point of user interface customisation is the dark mode feature, as will be described below. A dark mode will be implemented using a global override of CSS classes that are activated by a JS button. this will allow, through a radio button users to change their color scheme. Also as it uses cookies, as long as you don't delete session cookies in your browser the theme change will be saved.

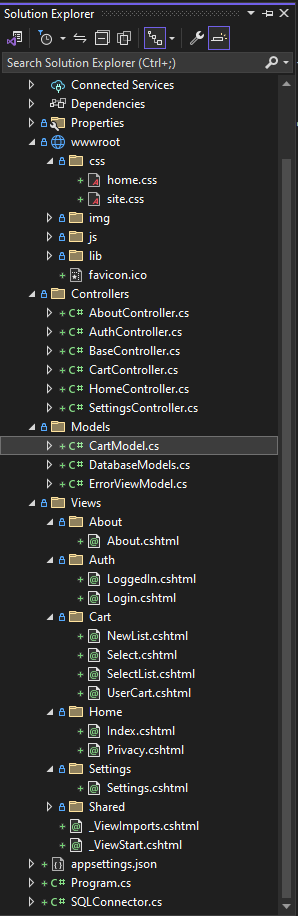
### Graphical and Multimedia requirements and planned implementation

The application specifications require both a video and an image to be added to the site, the image on the login page and the video on the about page. these will both be implemented by putting the assets into the www root folder from there on their respective pages they will be referenced in the form of an <img> (Image) and <iframe> (Video) respectively. A dark mode will also be implemented using a global override of CSS classes that are activated by a JS button.

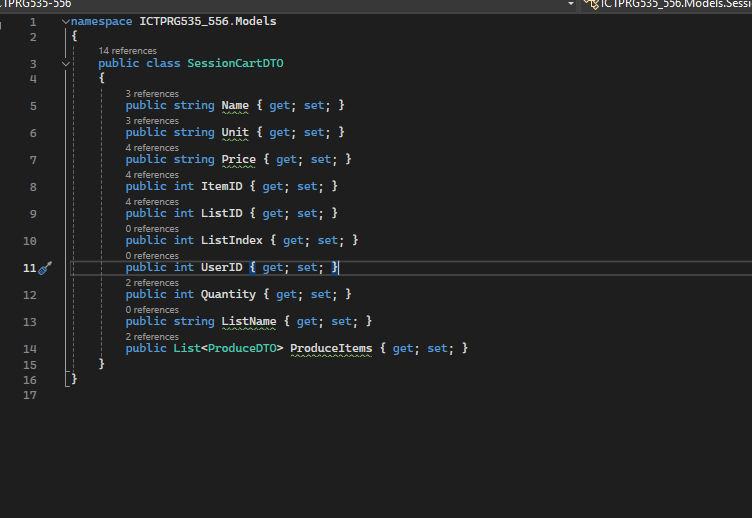
# Part 3 Application Development

## Screenshots

Screenshot of the newly created project in the Chosen Development Environment, Language and Framework

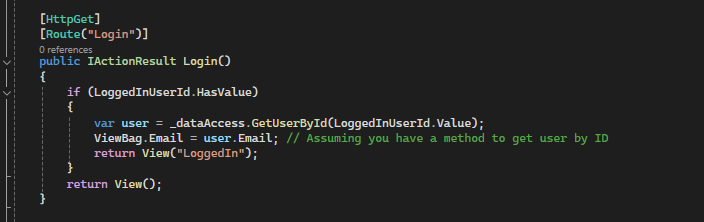


Screenshot of at least one completed data model within the Integrated Development Environment

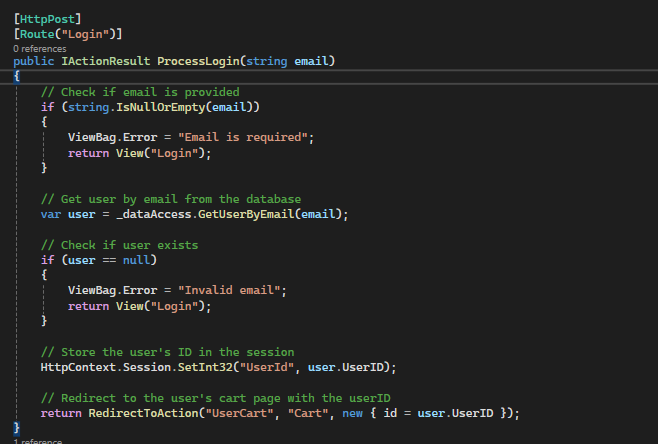


### Screenshots of the code created in at least one controller to handle the following HTTP Methods:

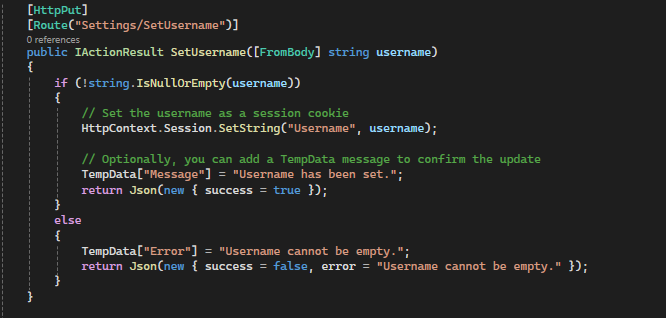
* HTTP GET



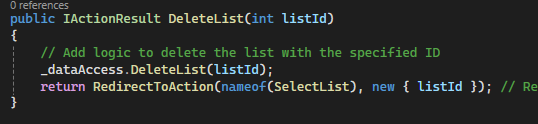
* HTTP POST



* HTTP PUT

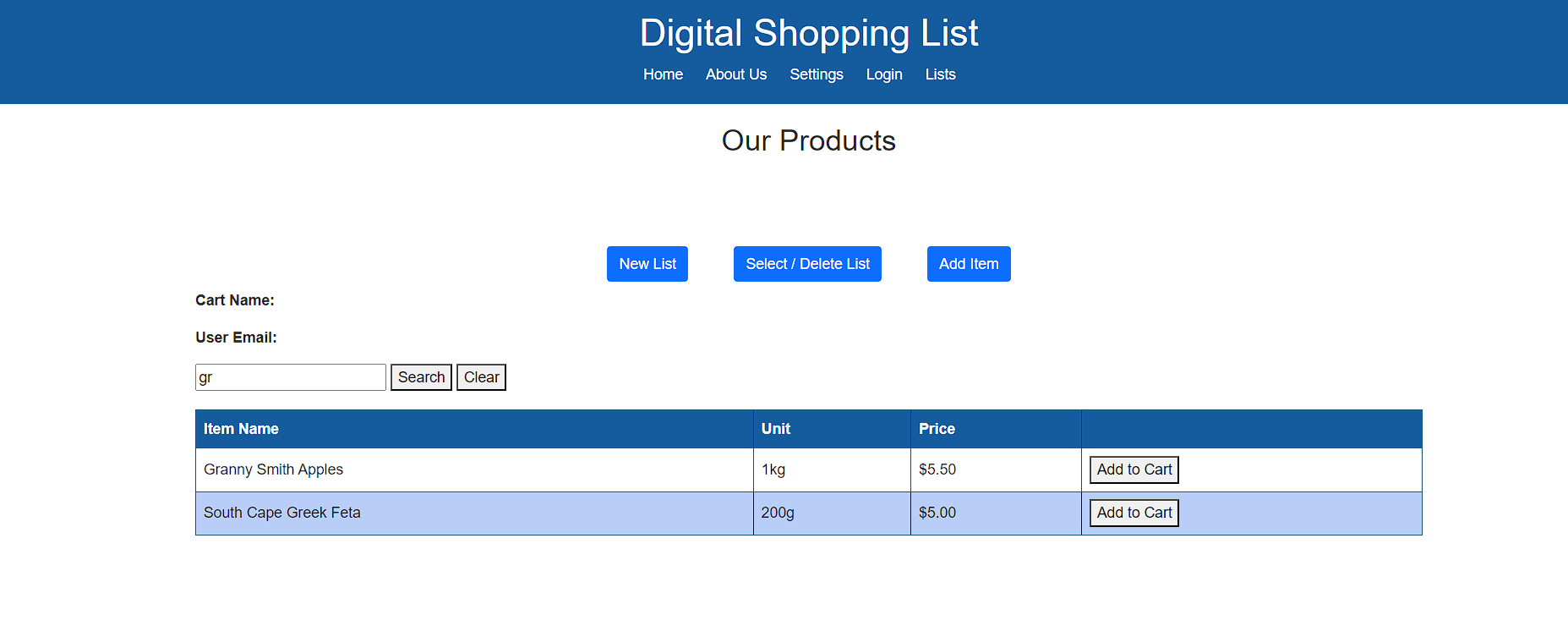


* HTTP DELETE

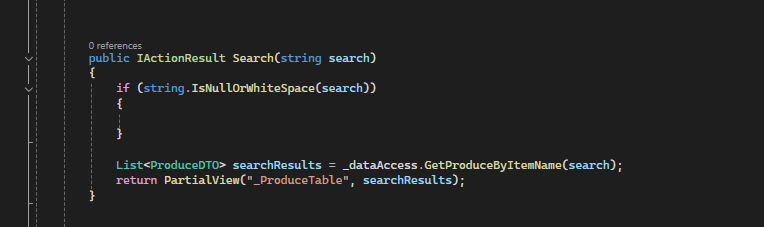


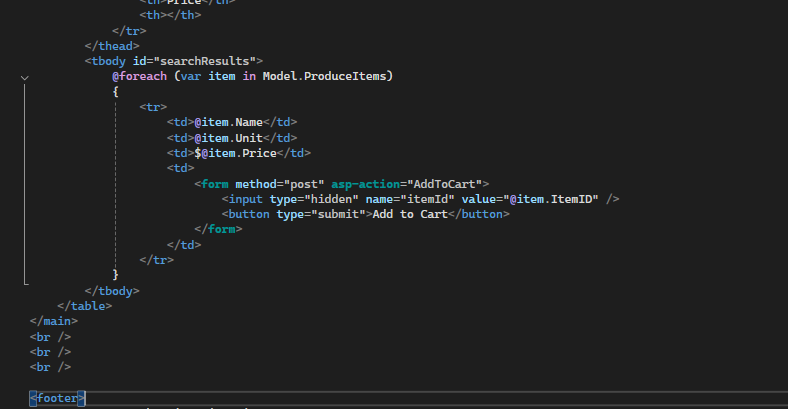
### Screenshots of the created views and source code demonstrating the following points:

* Data being rendered dynamically

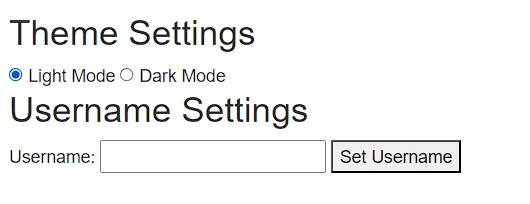


* Code used to render data dynamically

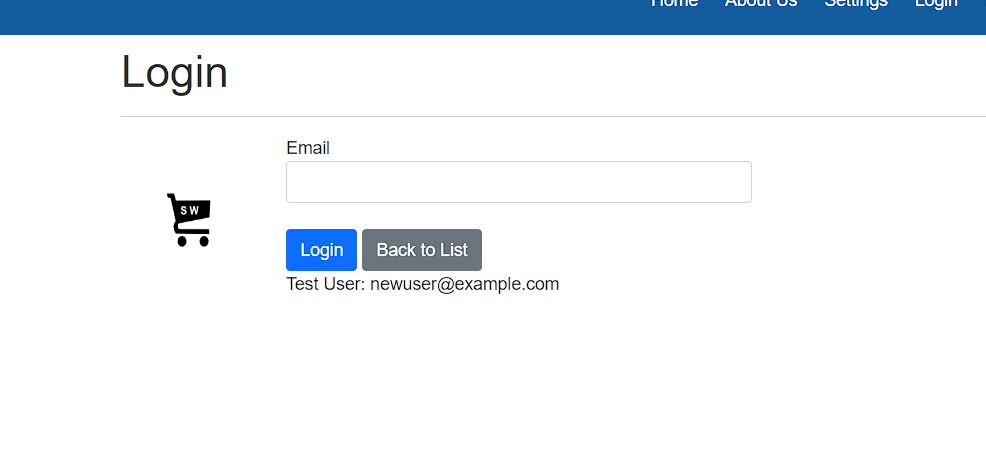




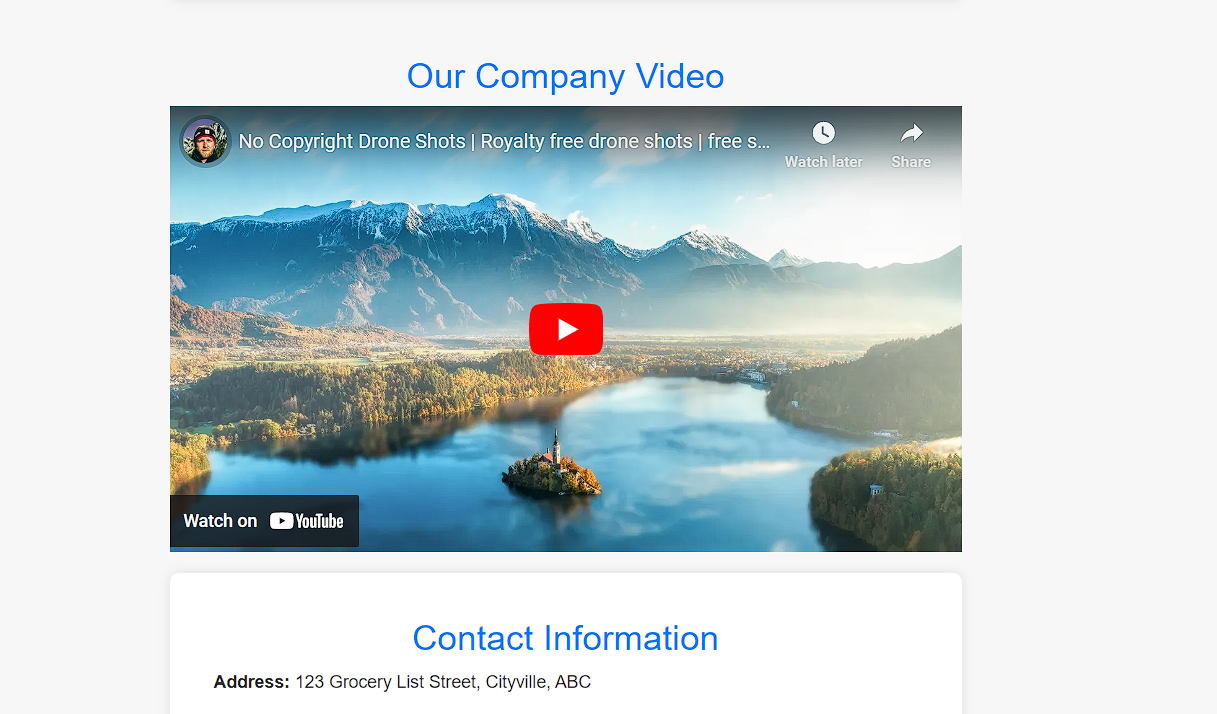
* User Customisation view or component



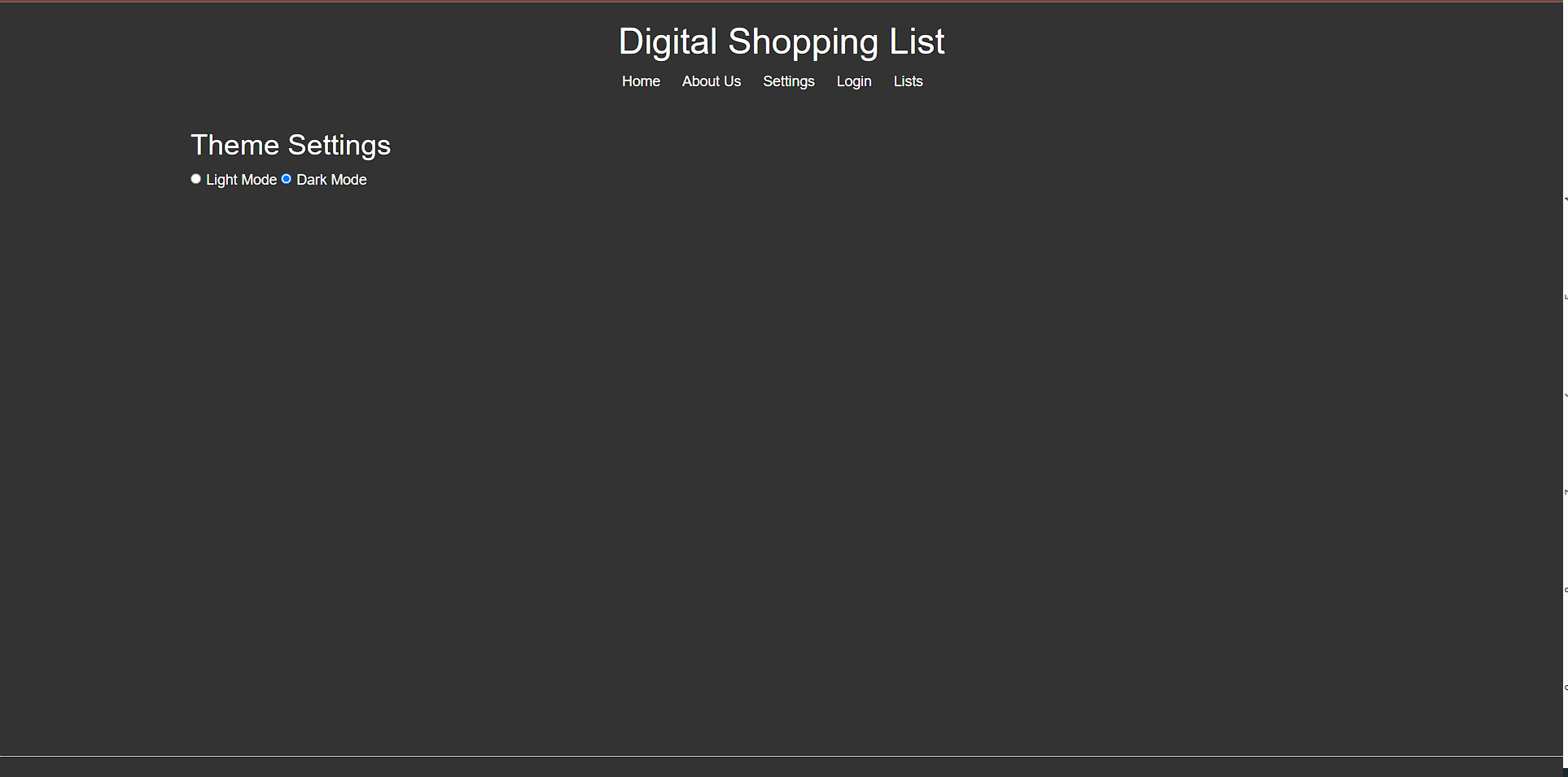
* Rendered Graphics



* Included Multimedia content

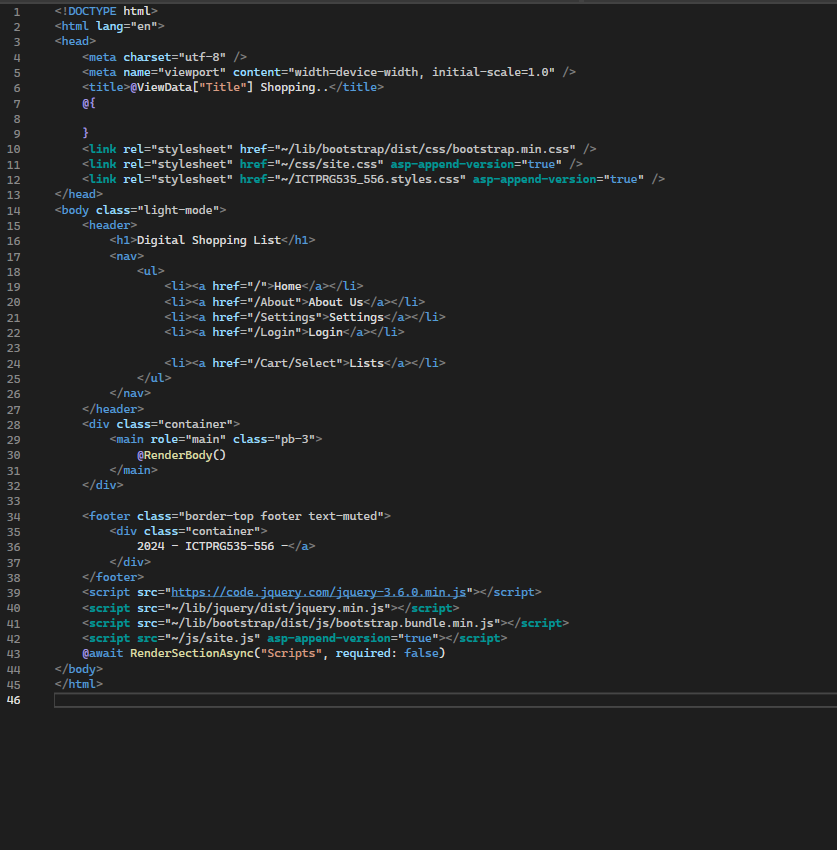


* User interface personalisation requirements as defined in the user interface design plan.



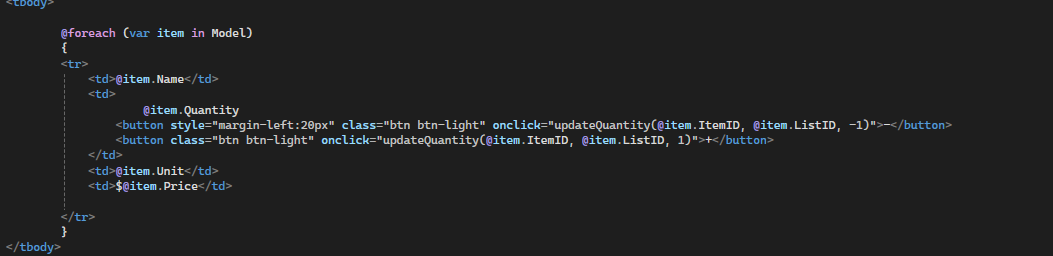
* Code demonstrating interaction design patterns on at least 2 occasions

This is the layout file that shows the same navigation menu on all pages

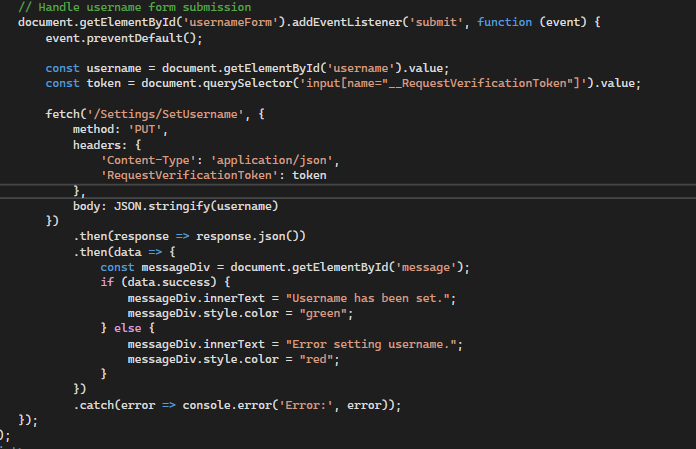


The list select and list view both utilise the same design and placement of options buttons allowing for smooth usage





* Code demonstrating the Client Side validation implemented in at least 1 view



# 

# Part 4 Testing + Validation

## Testing Table

| **Functionality** | **Test** | **Result** |
| --- | --- | --- |
| New List | Select the New List button on the home page once logged into the application and then navigate to the Select List page through the header links to check if the newly created list displays | Once created, the list showed in the select list page without items or a name as designed |
| Delete List | Select the Delete button in the select page once logged in and with a list present. page should refresh without the list in either the view or the database | Once the list delete button is pressed the list select page refreshes without the deleted list and it is deleted from the database |
| Delete Item in the list | Select a list with items in it, then press - on the quantity. this should remove the item from the list, changing its quantity from one to zero | Once the item quantity is changed to zero the item is removed from the list. |
| Select List | Select a list using the select button on the select list page. once selected it should display the list in its contents, an option to delete an item and the total | Once a list is selected all the contents of the list and the options are displayed correctly |
| Login | Login using the login screen and the test email, this should redirect to the select list page | Login works, saves the cookie and redirects |
| Add to cart | On the home page, once a list is selected, as seen by the selected list label each item in the products list should be addable to that list, Add an item and checklist for the result | Once a list is selected and an item is added it is displayed in the list and the total is updated |
| Quantity change | Select a list with items in it, then press + on the quantity. this should add once to the quantity and update the total. | Once a list is selected and the quantity is changed it is reflected in the quantity column and the total |

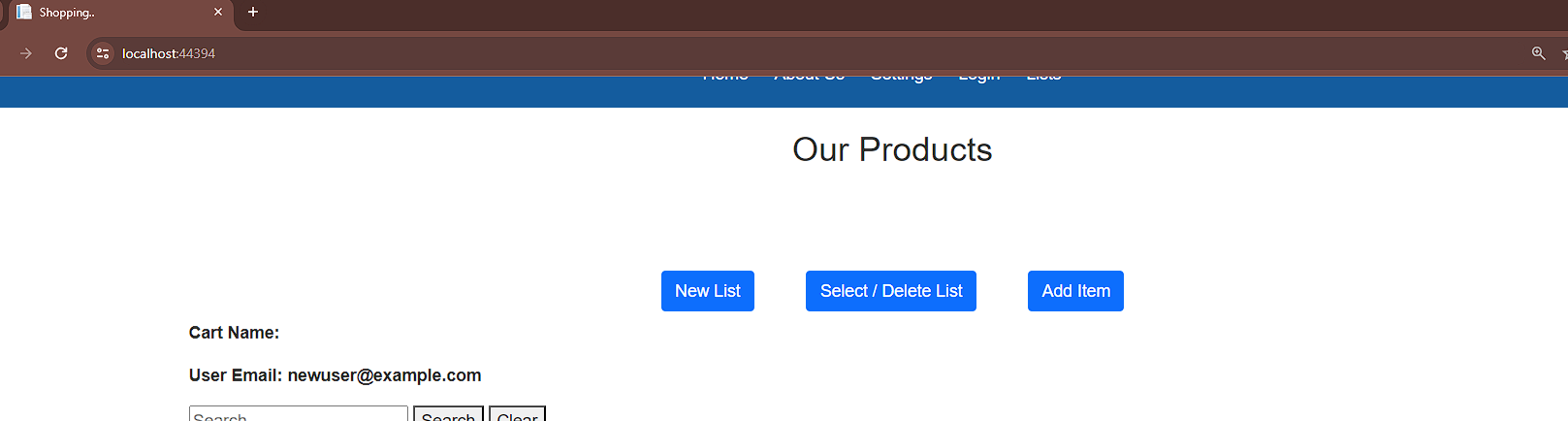
## Testing Summary (including testing tools used)

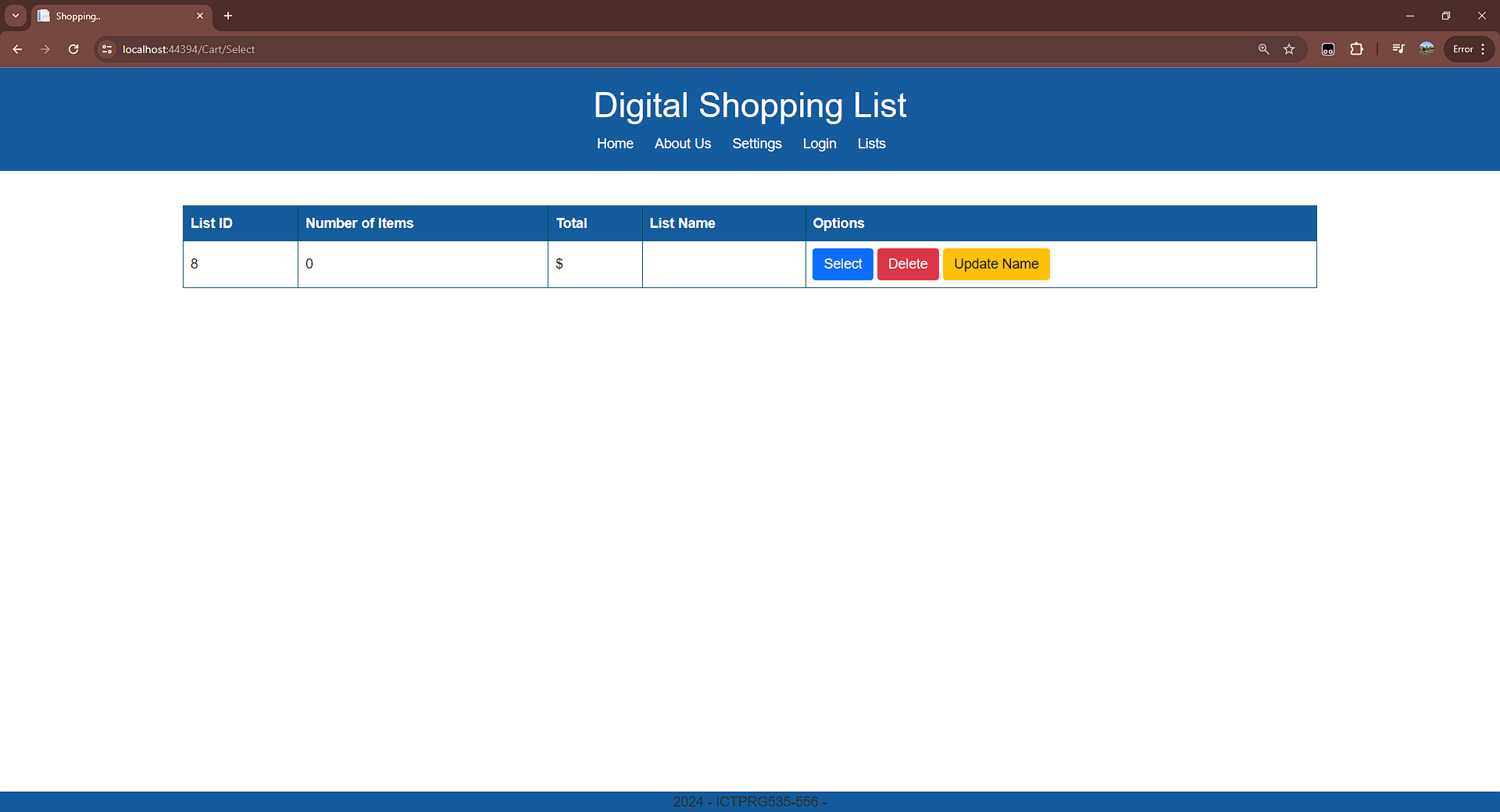
The tests were completed through unit testing of each major feature of the application, each function had a breakpoint on its object/s and stepped through to check each variable for any discrepancies in the logic or data flow of the application. Throughout this process, the user interface is checked through the browser to ensure that the user interface interacts as required, the tools used were the Visual Studio debug kit that is included with Visual Studio and Google Chrome with the inspect element tool to view the front-facing code in runtime.

## Testing Screenshots

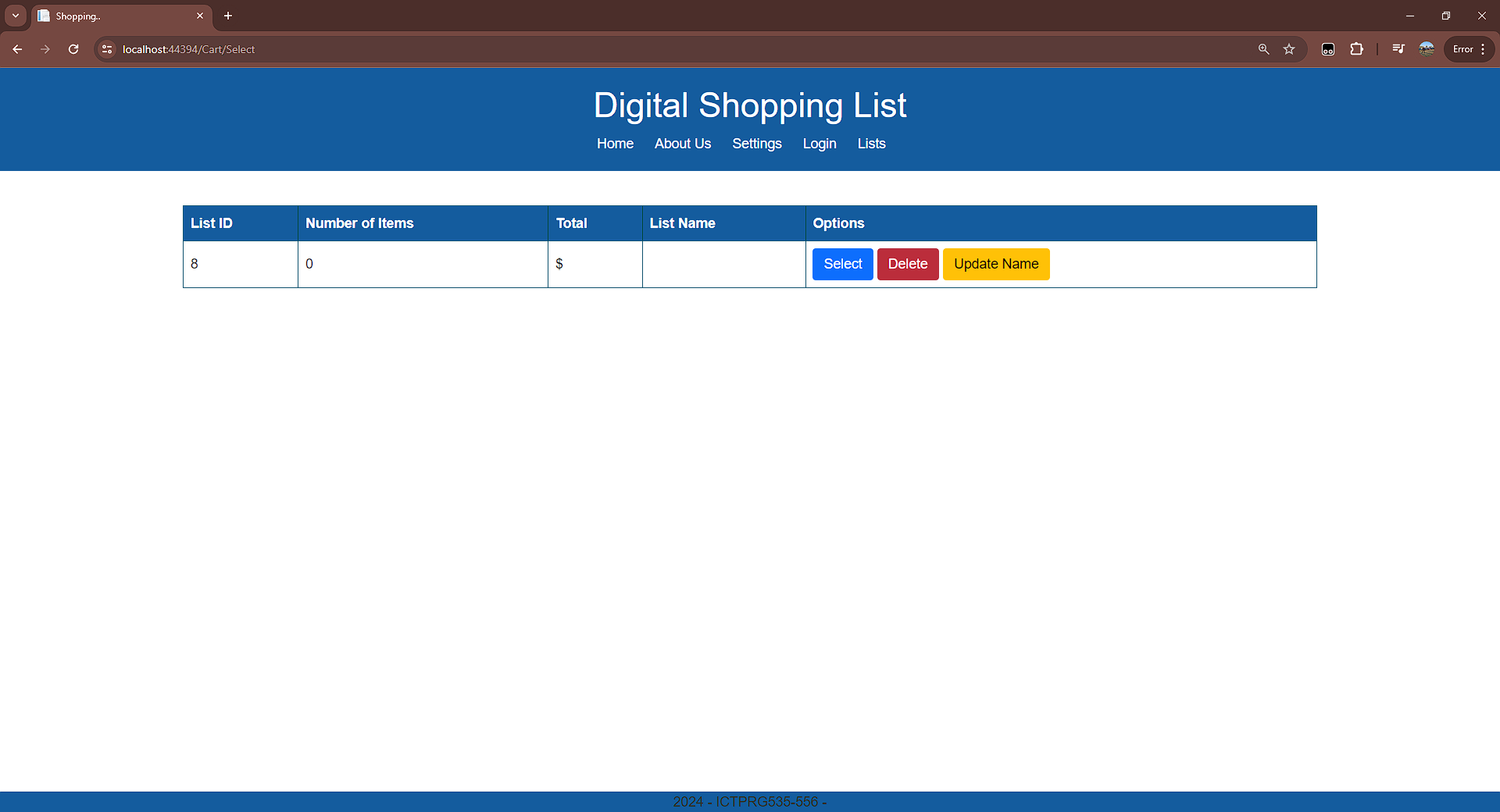
* Browser testing

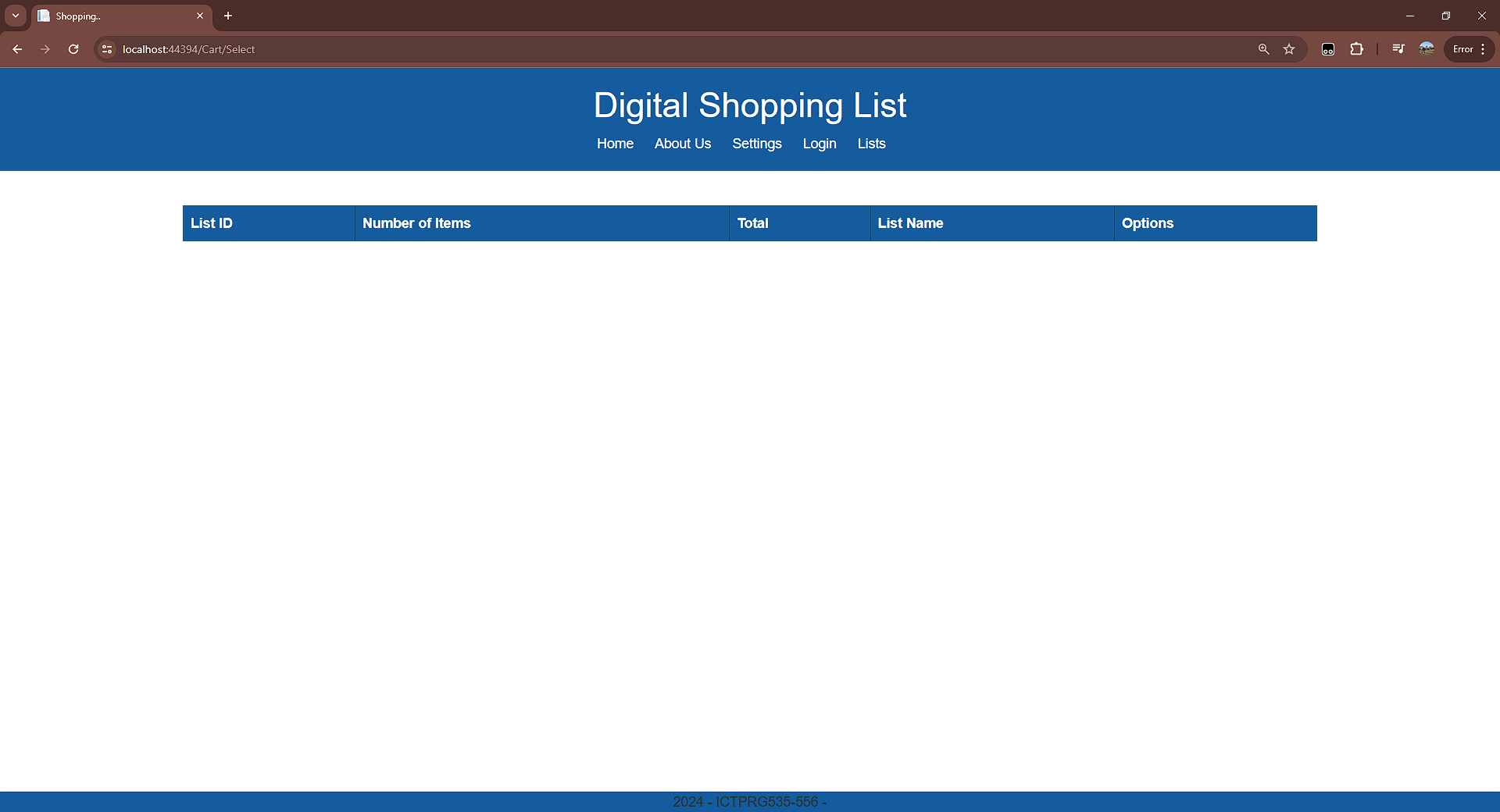
**New List**



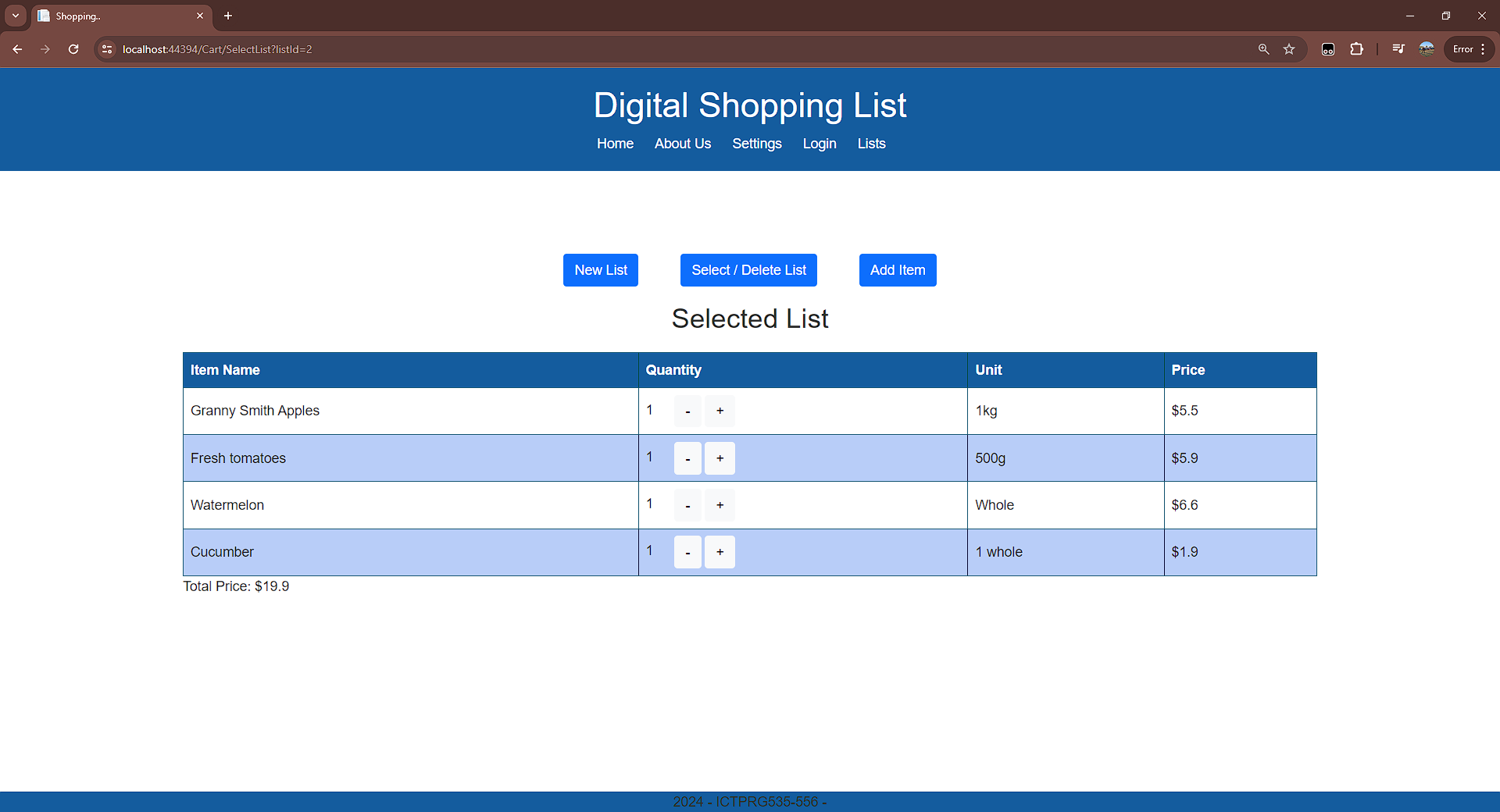


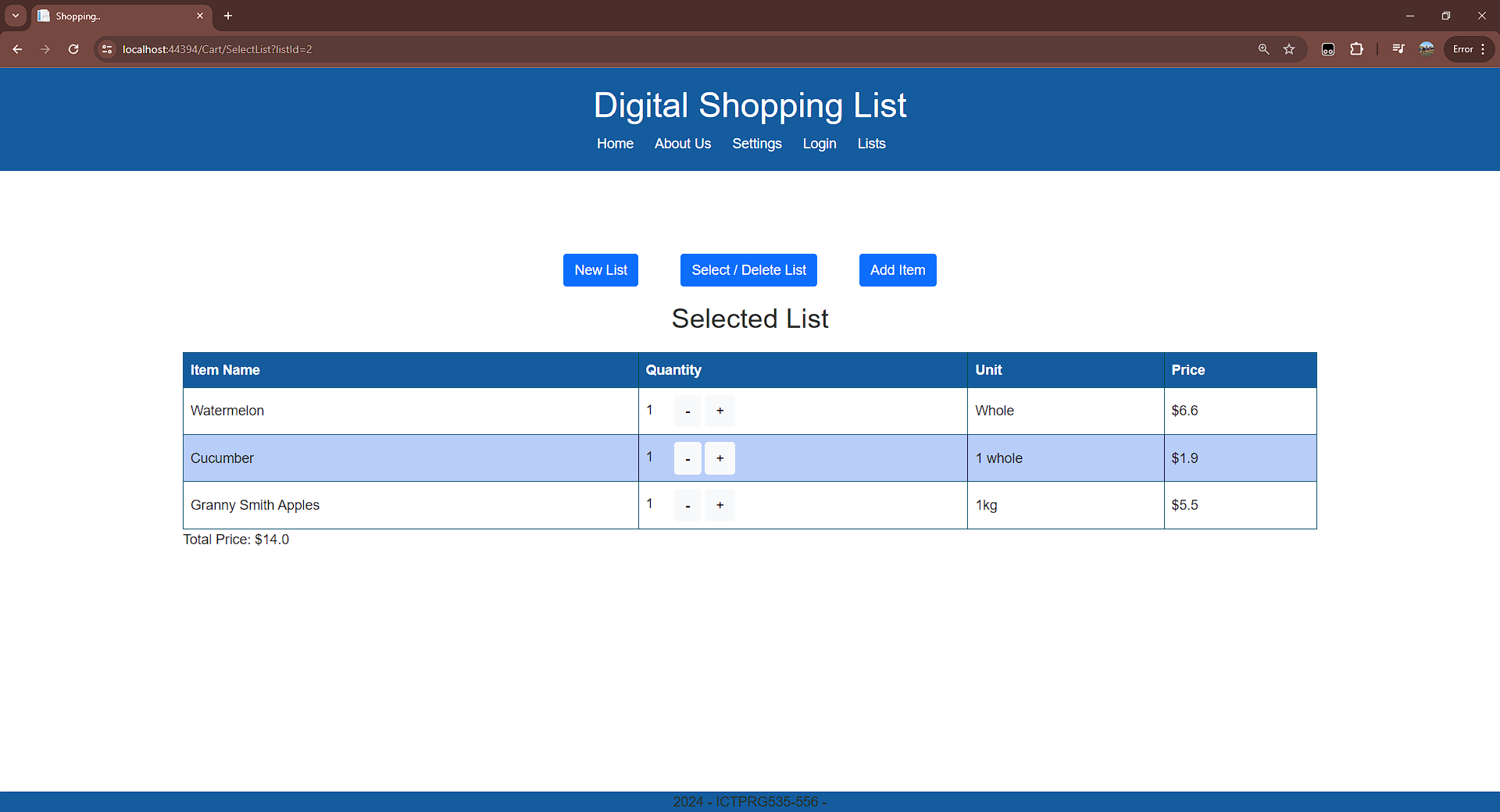
**Delete List**



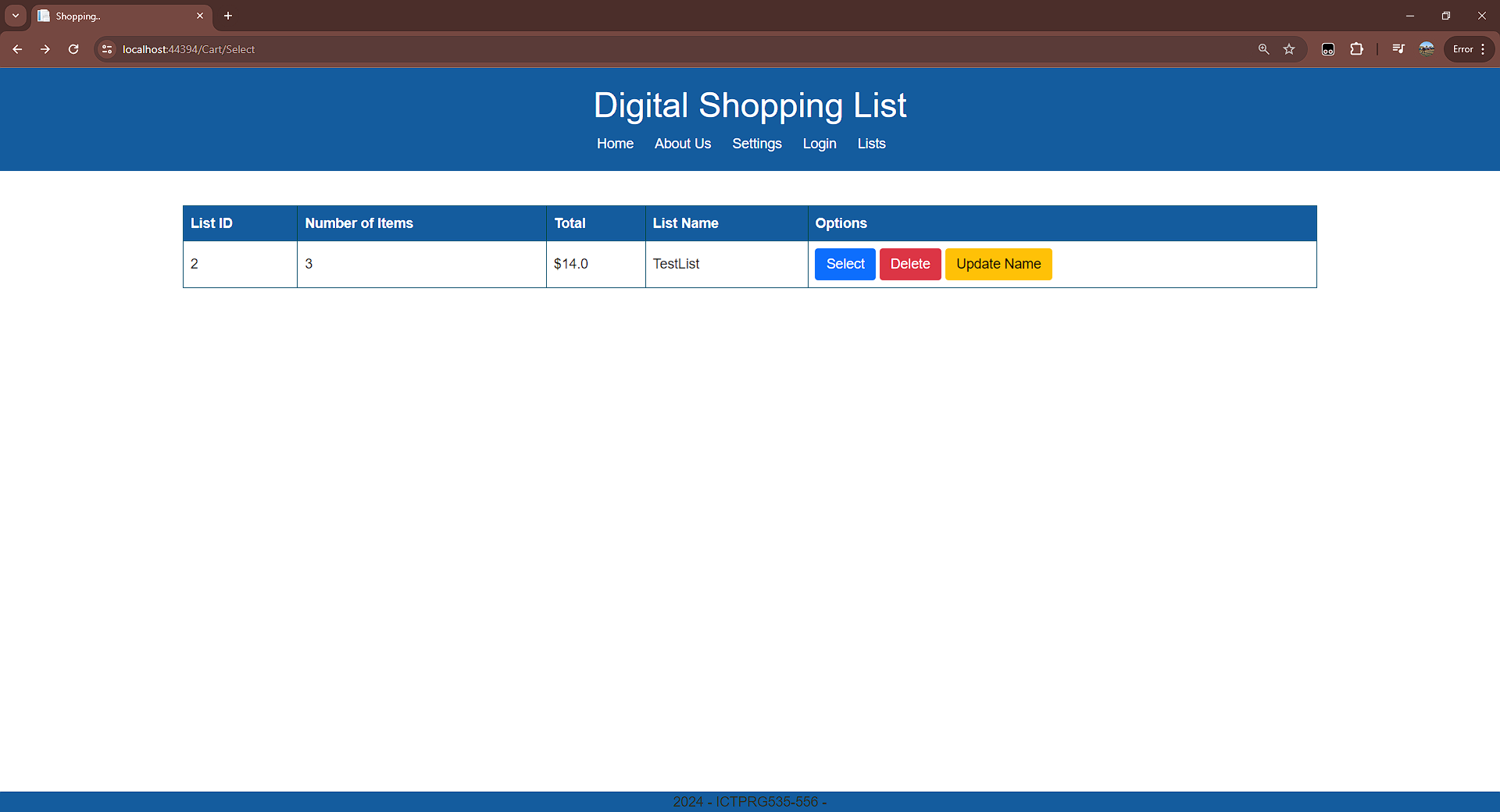


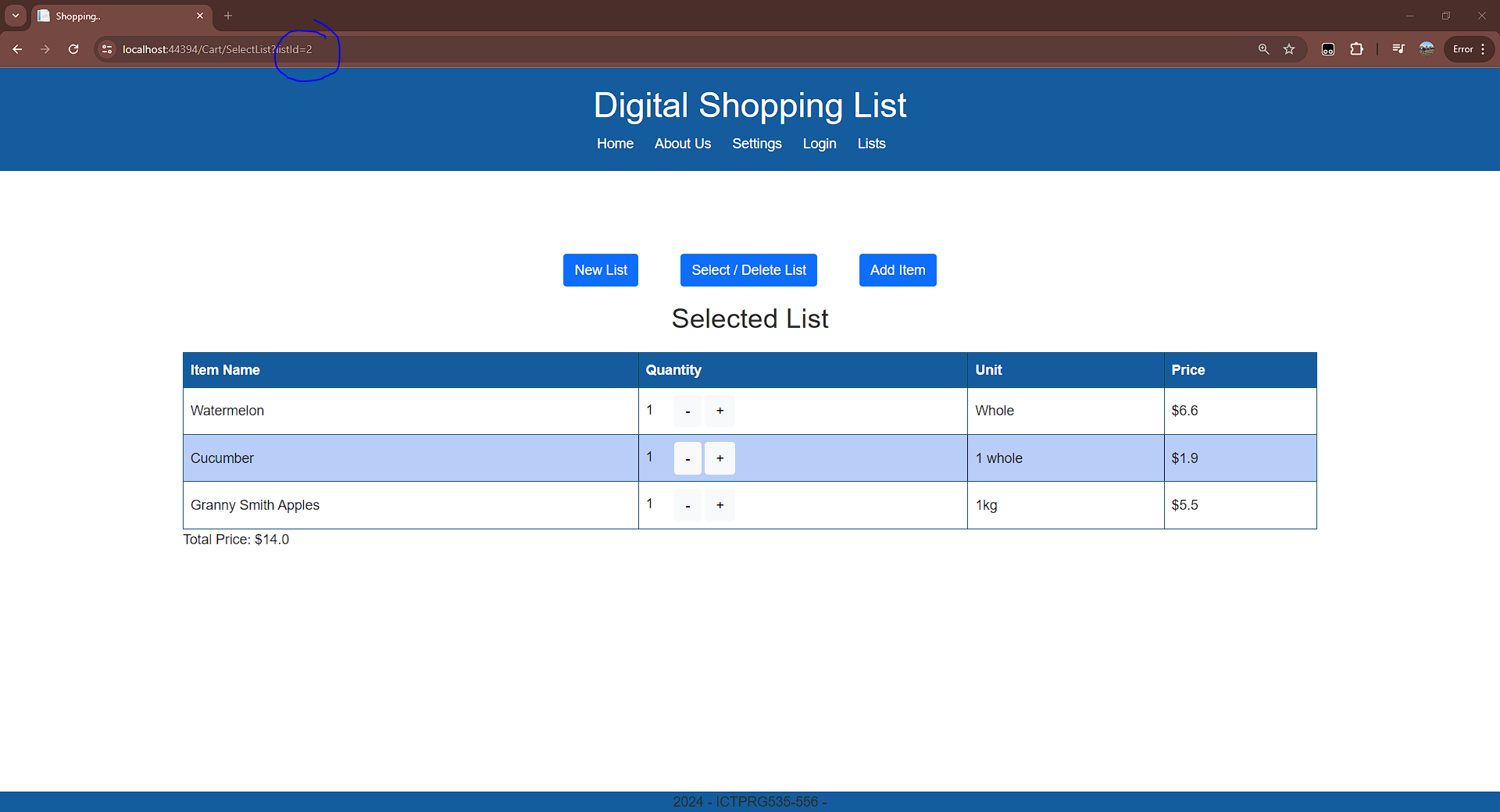
**Delete item in list**

****

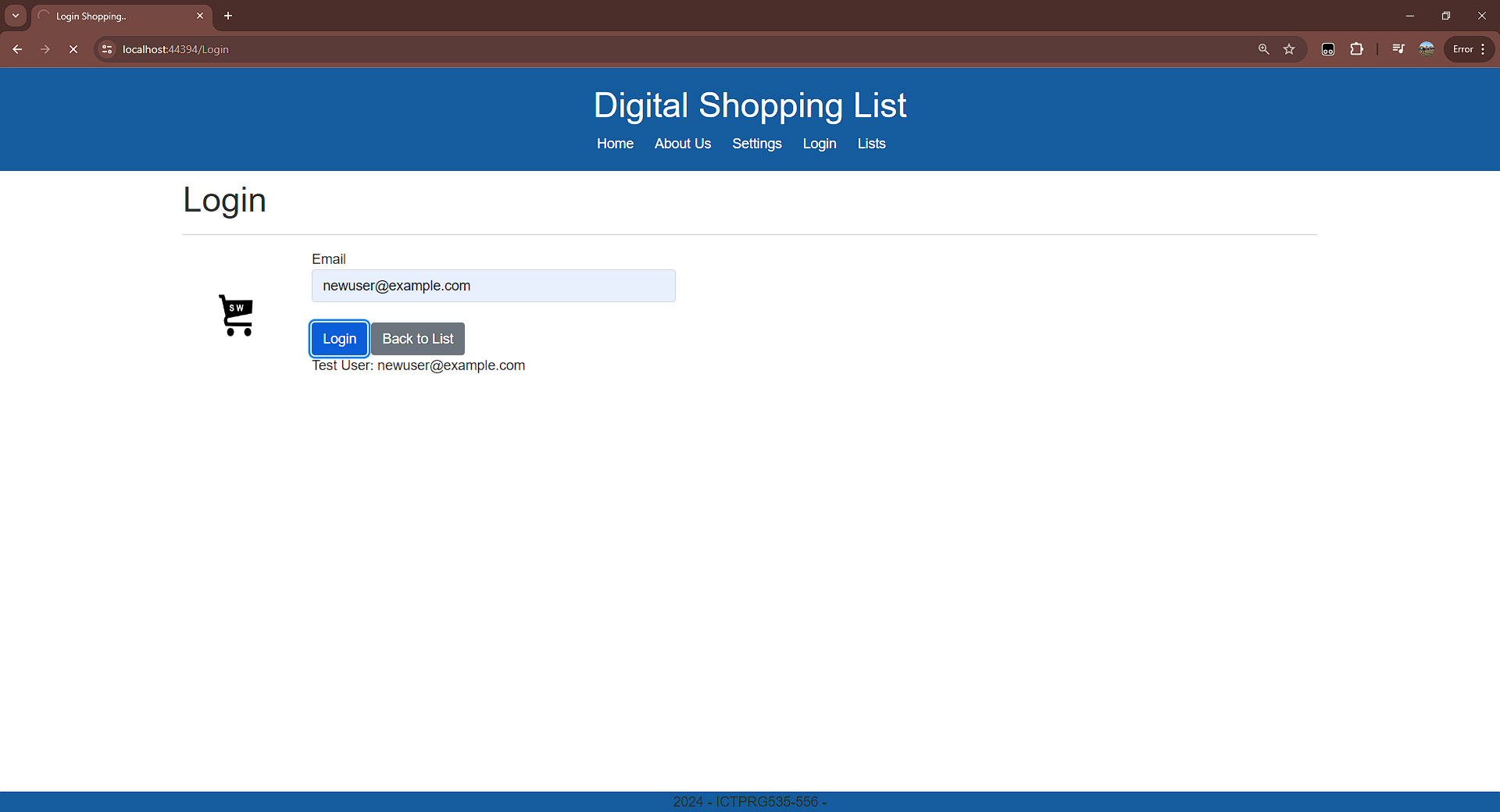


**Select List**

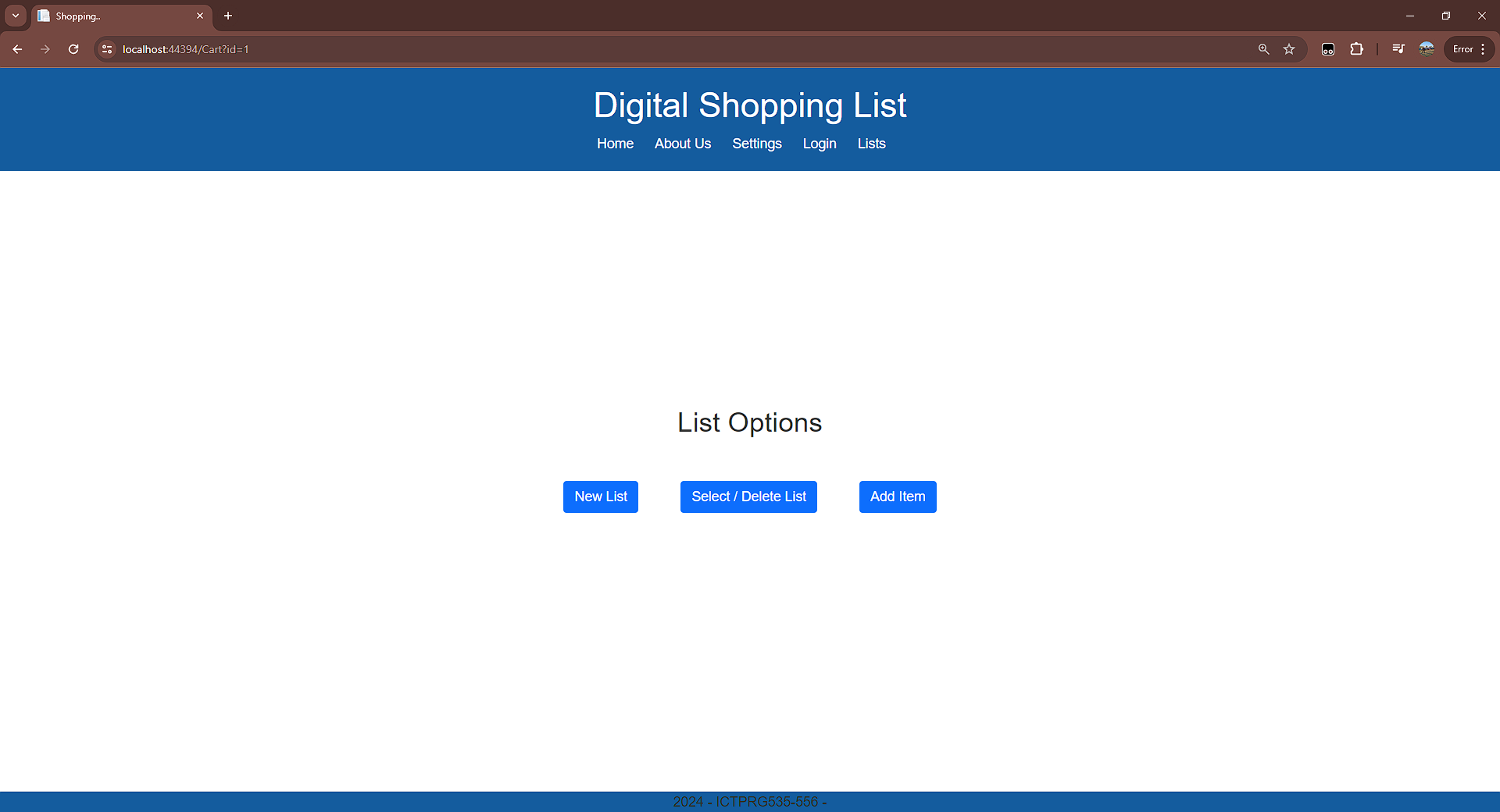
****

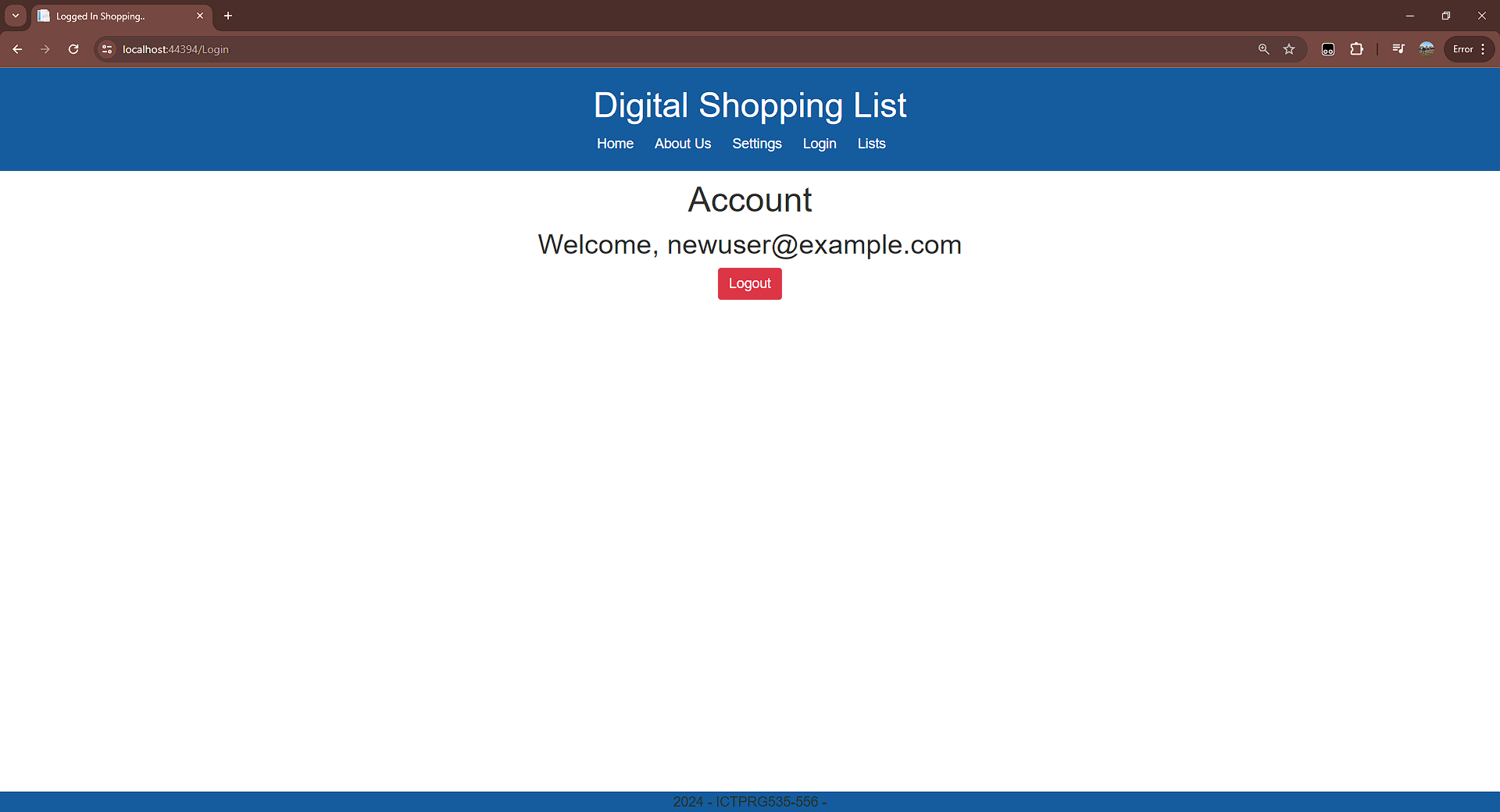


**Login**

****

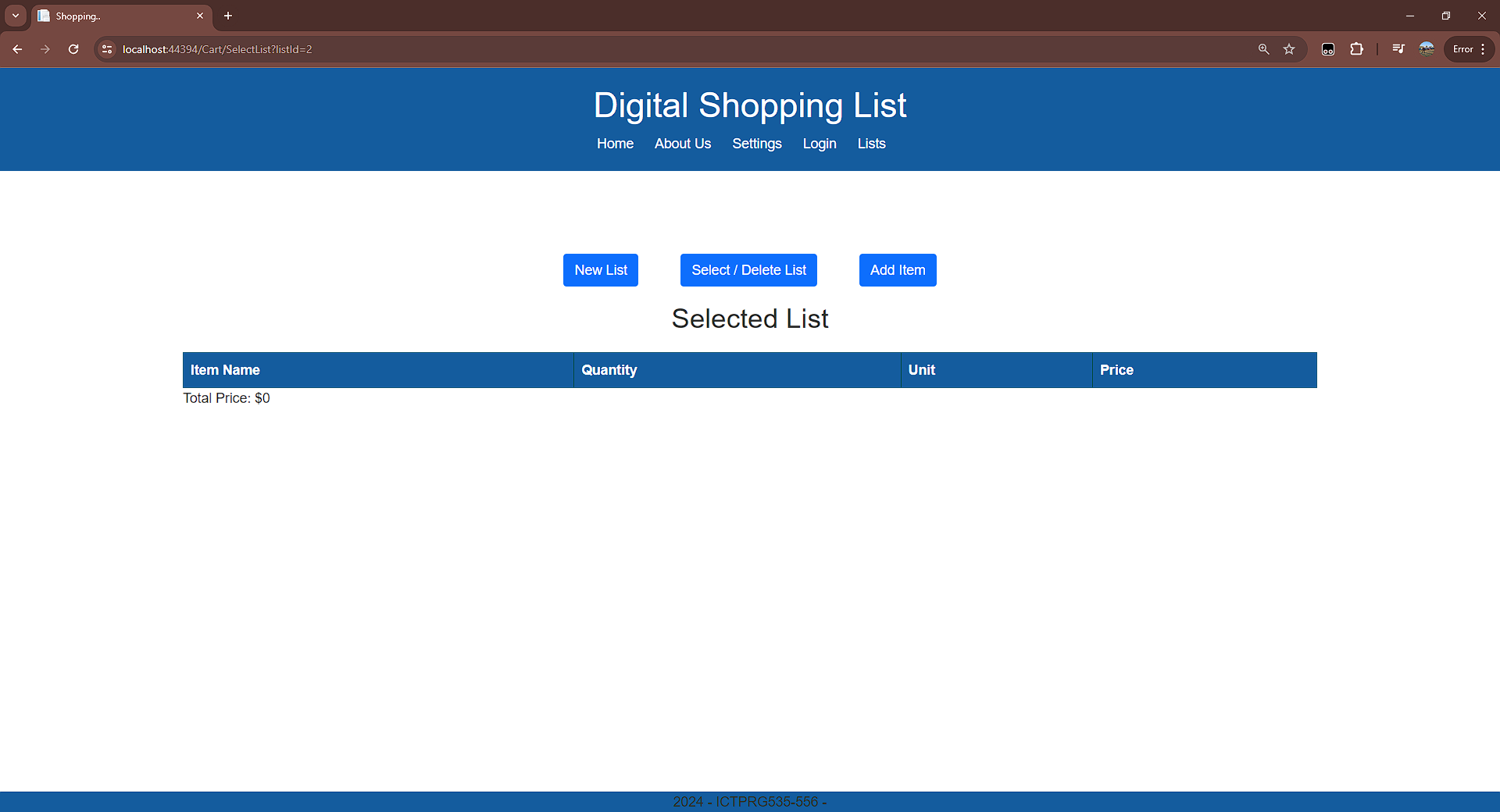
*Redirect from login*

****

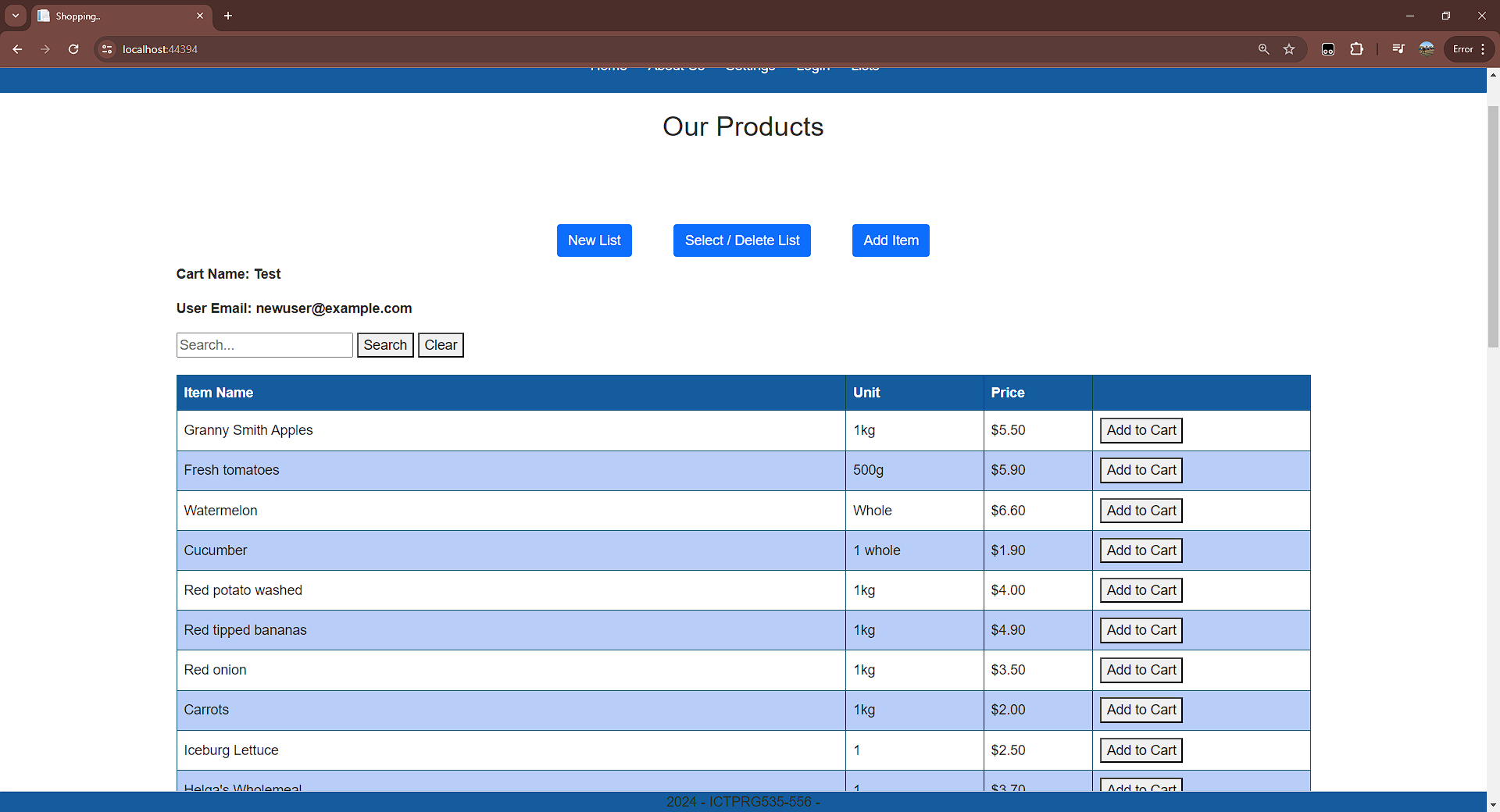


**Add to cart**

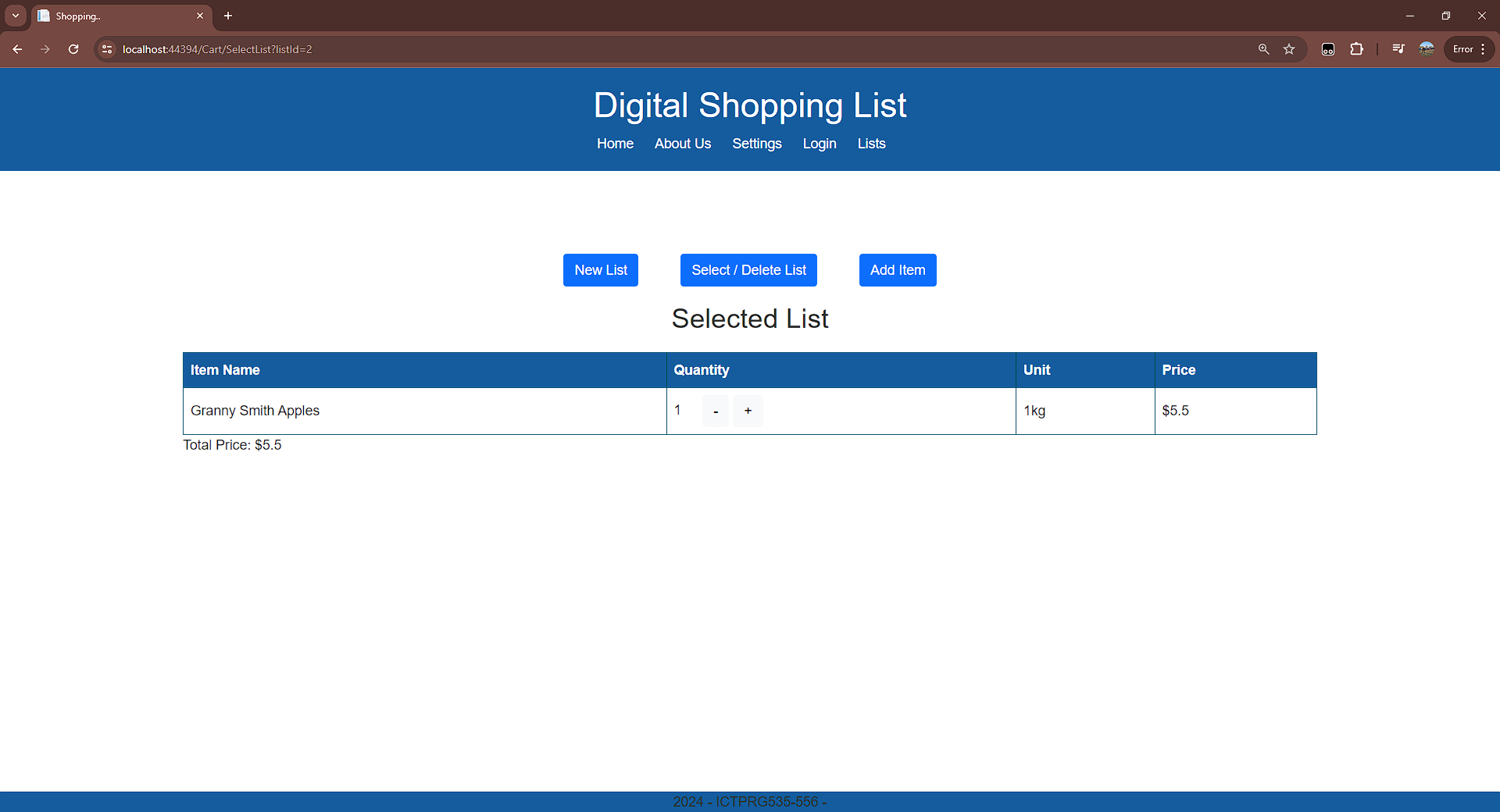
*List before addition*

****

*Produce table with the selected cart*

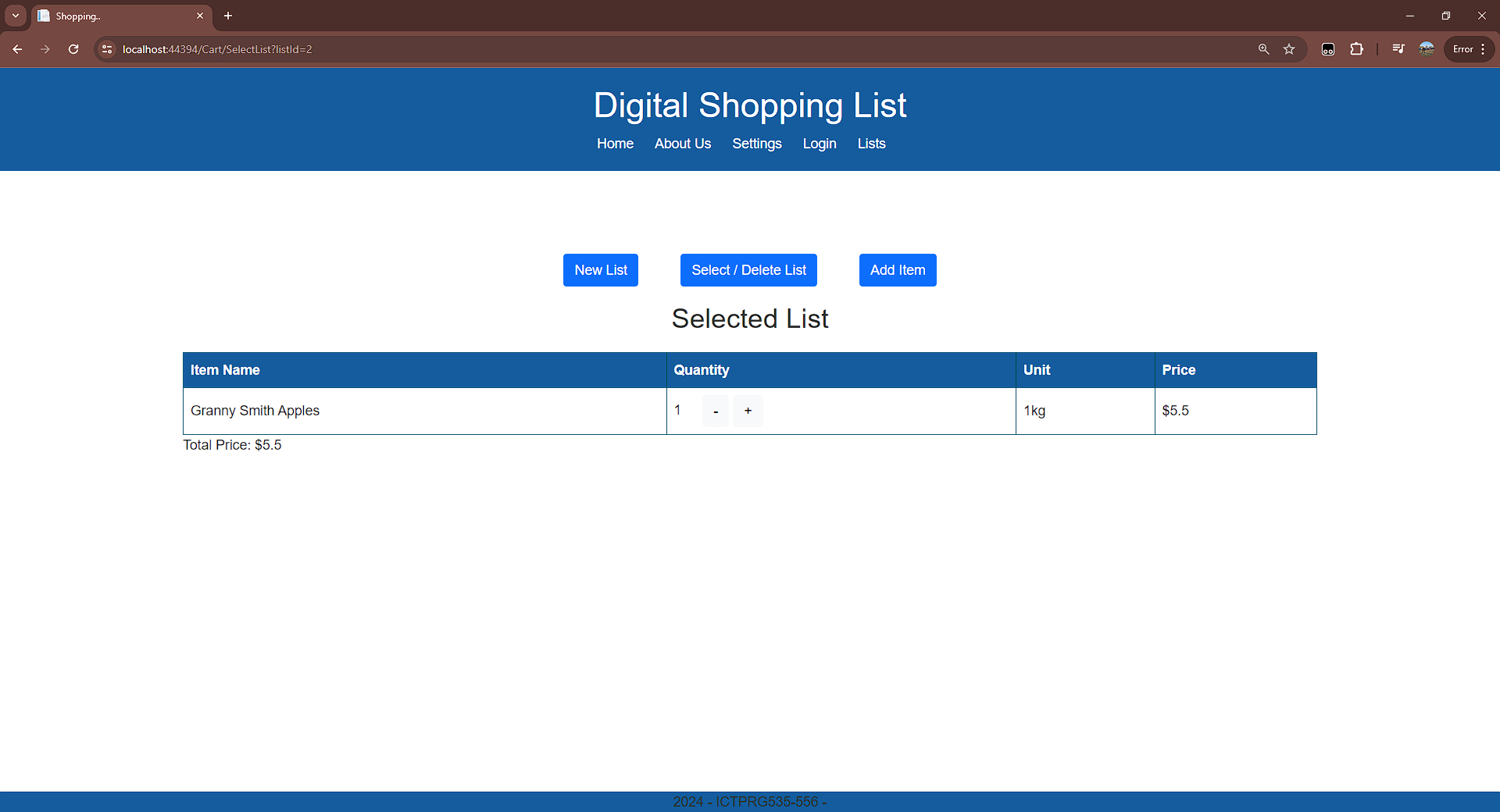


*List after addition*

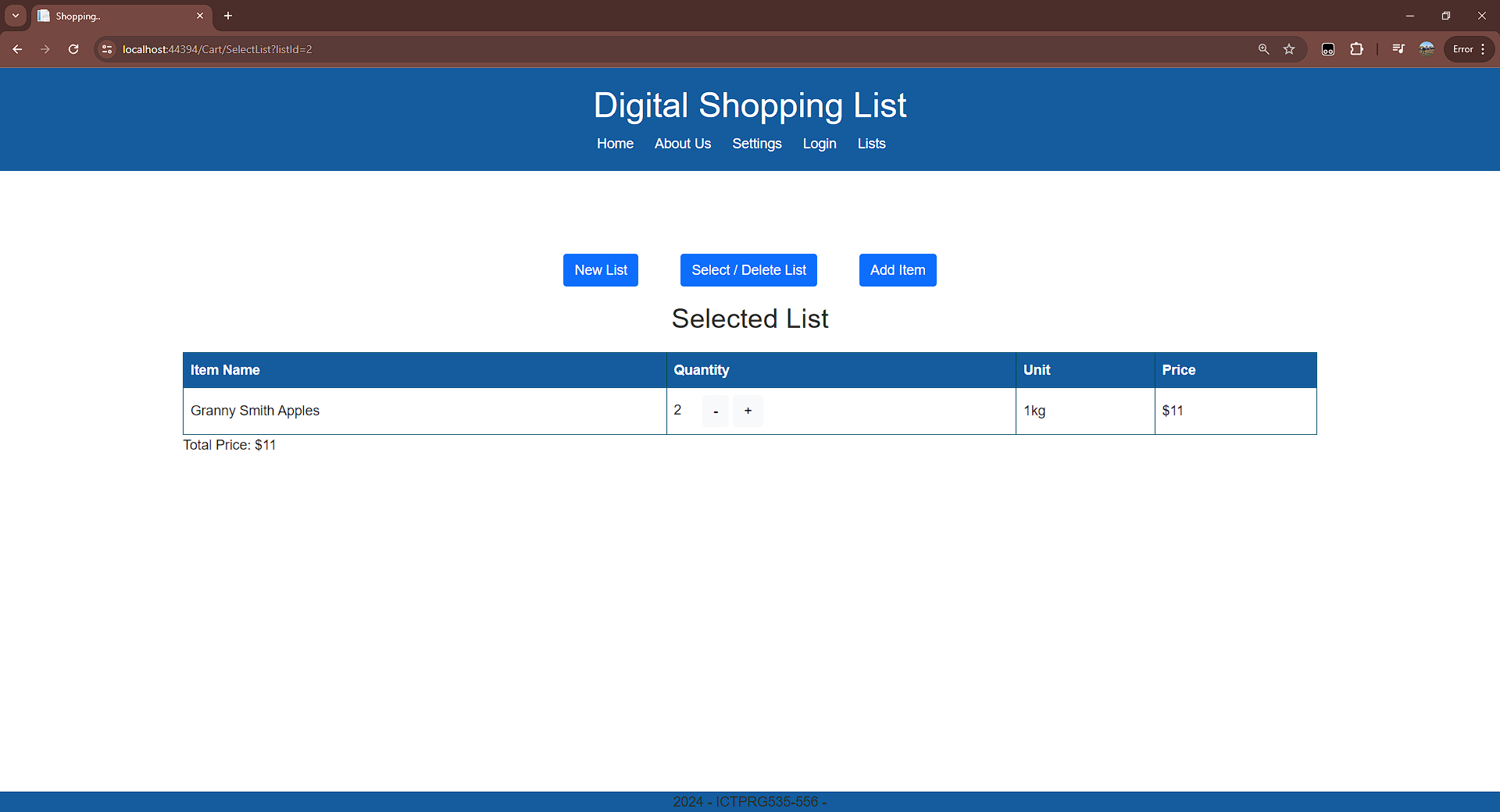


**Quantity Change**

*List before quantity change*

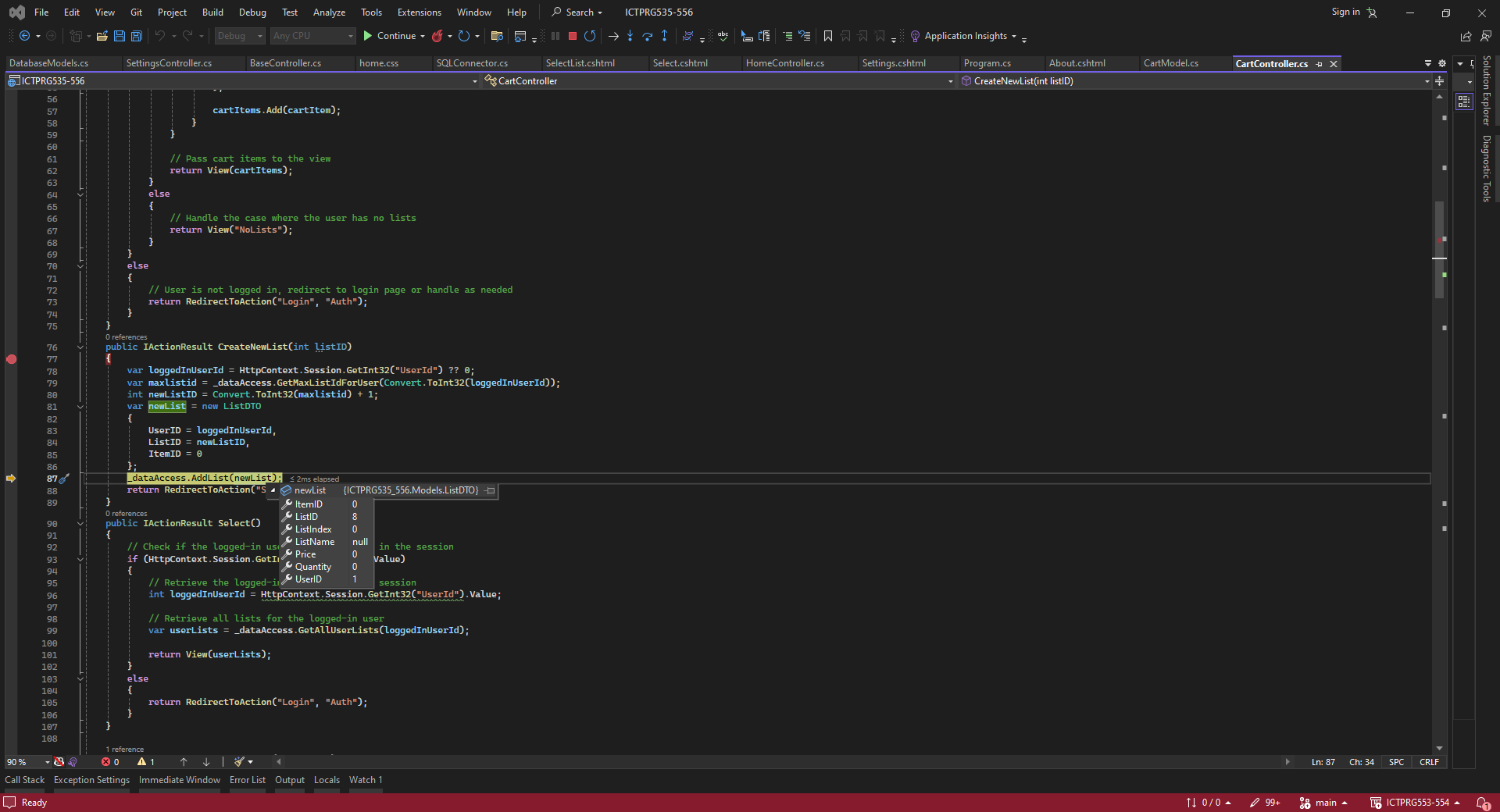


*List after quantity change with the updated total*

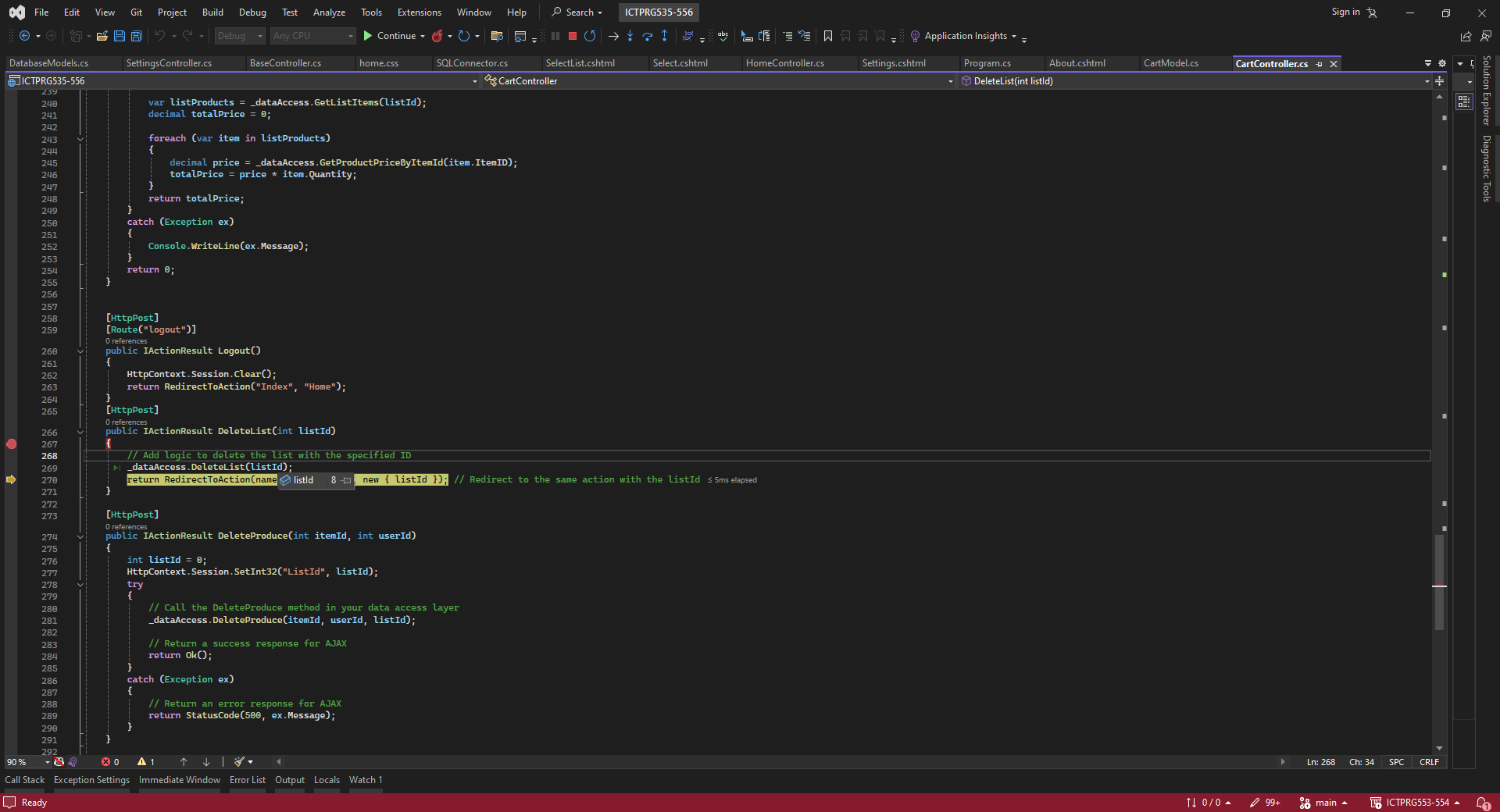
**

* IDE testing

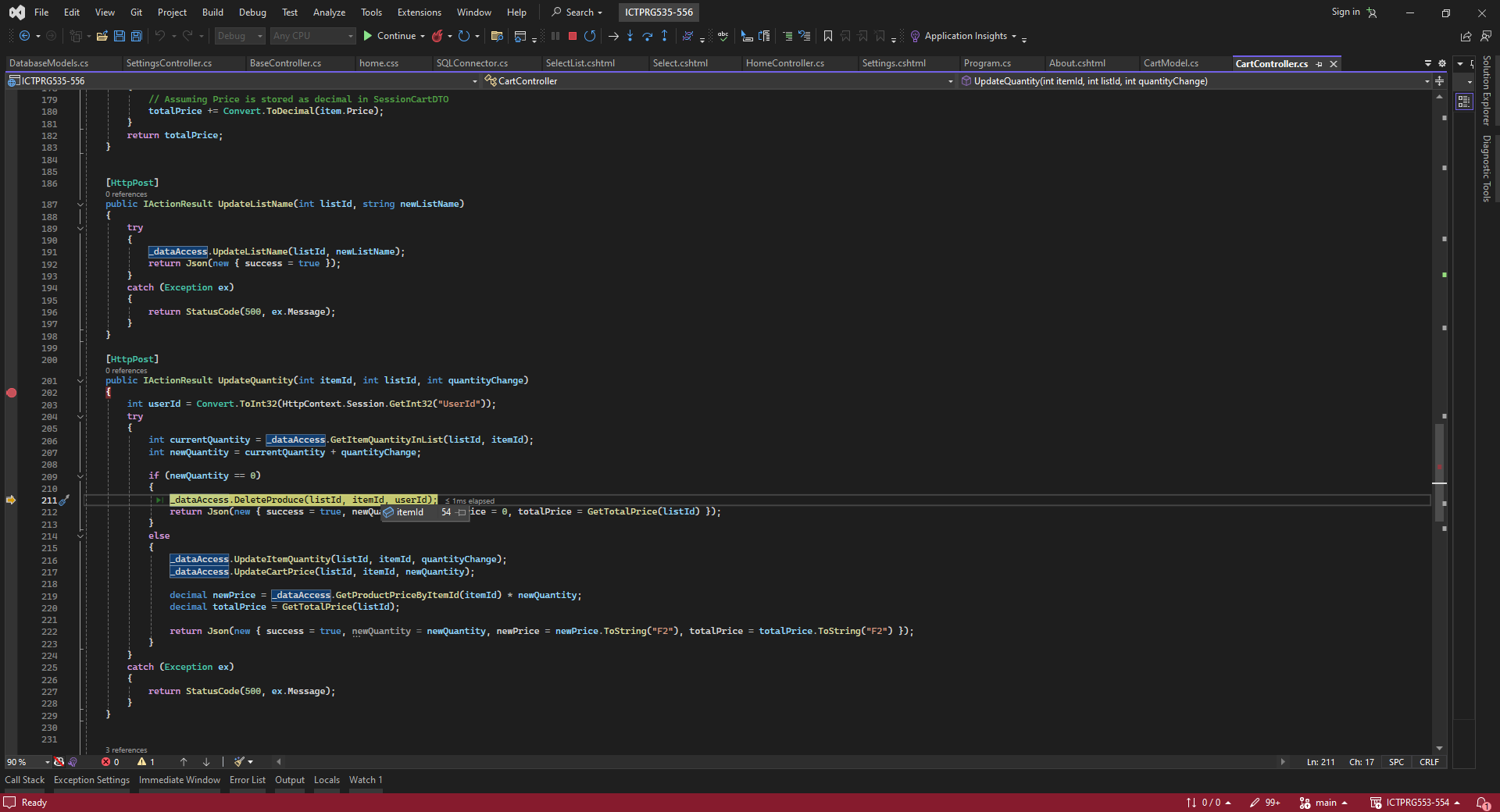
**New List**



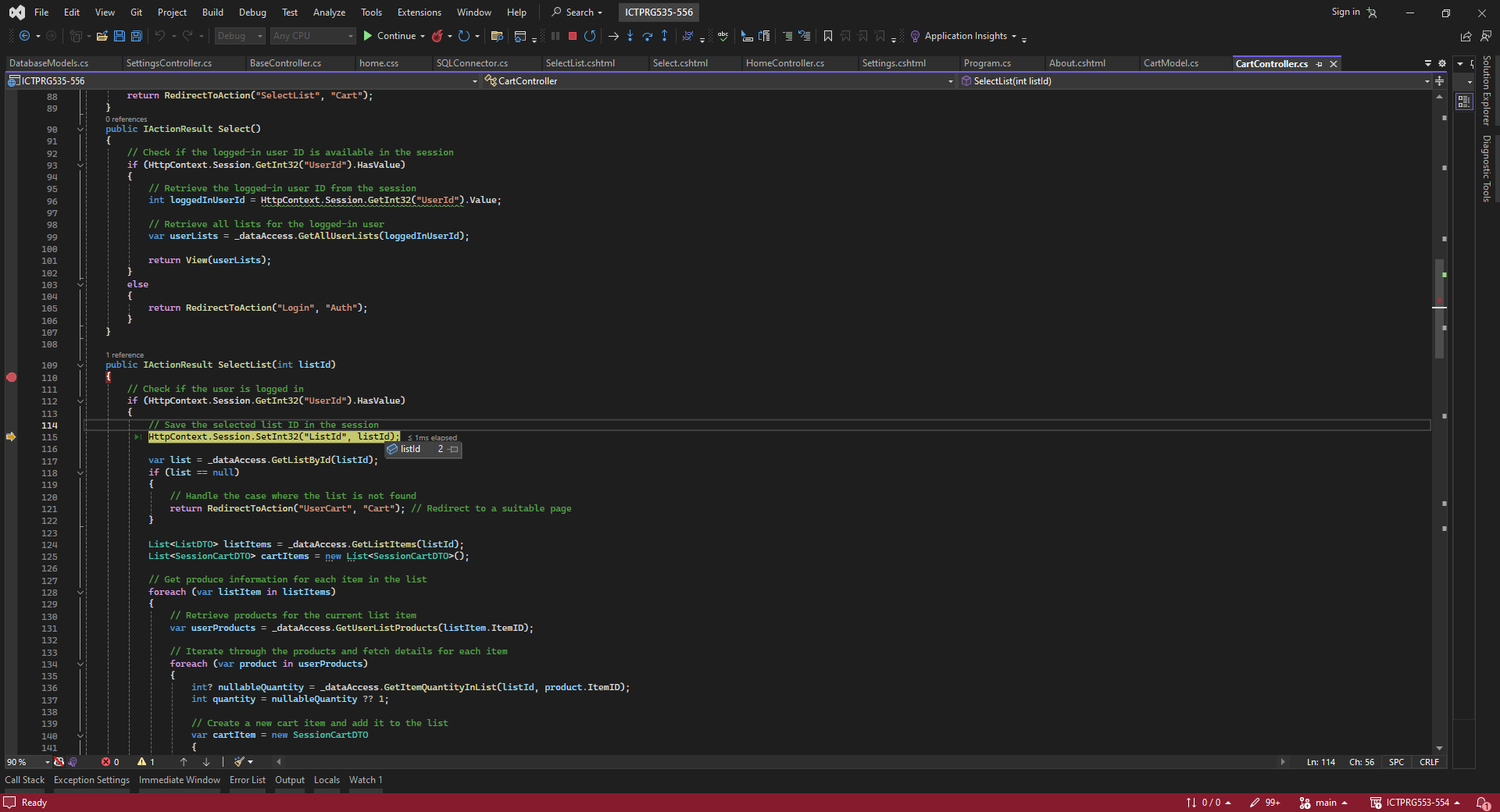
**Delete List**



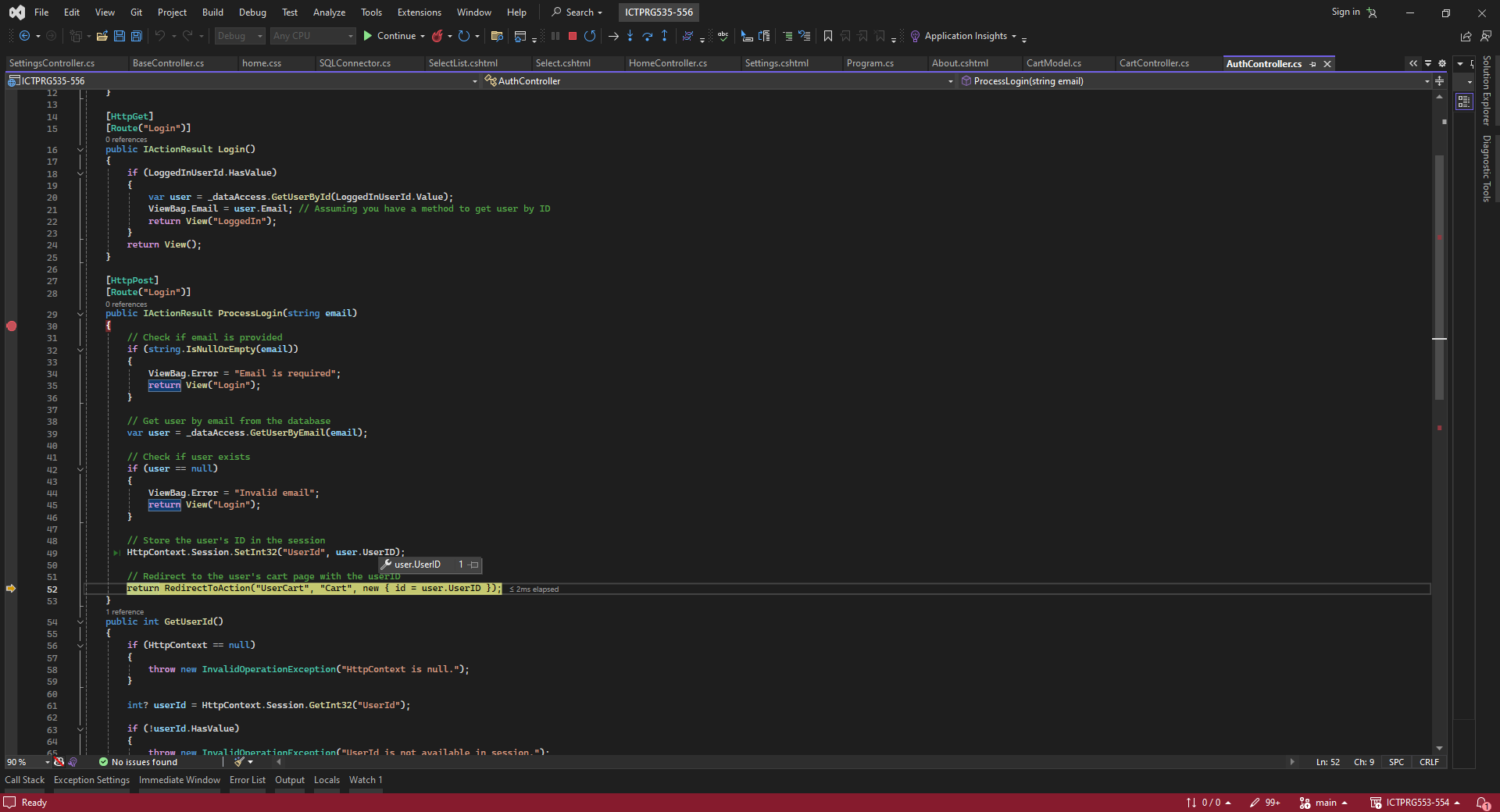
**Delete item in list**

****

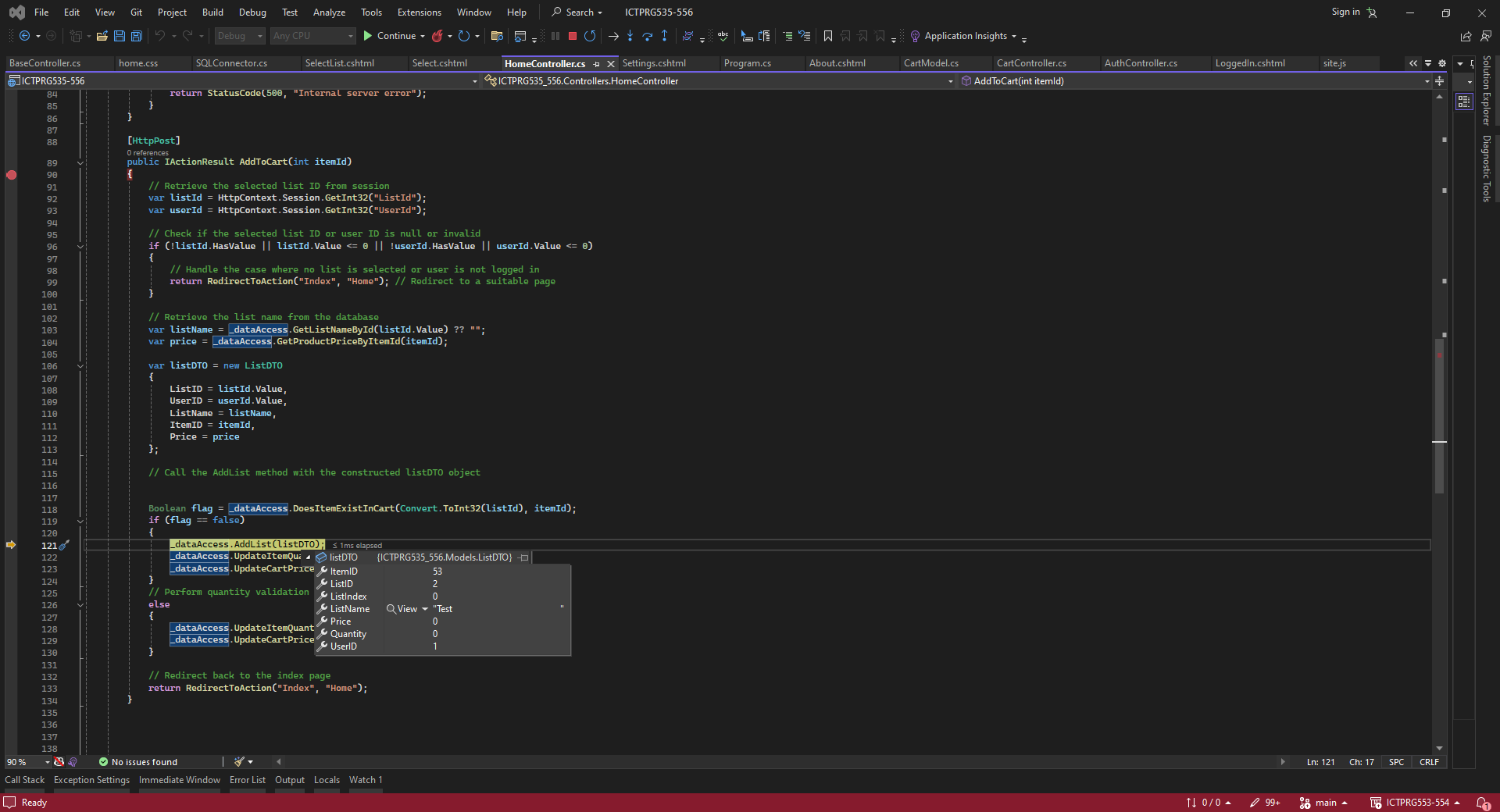
**Select List**



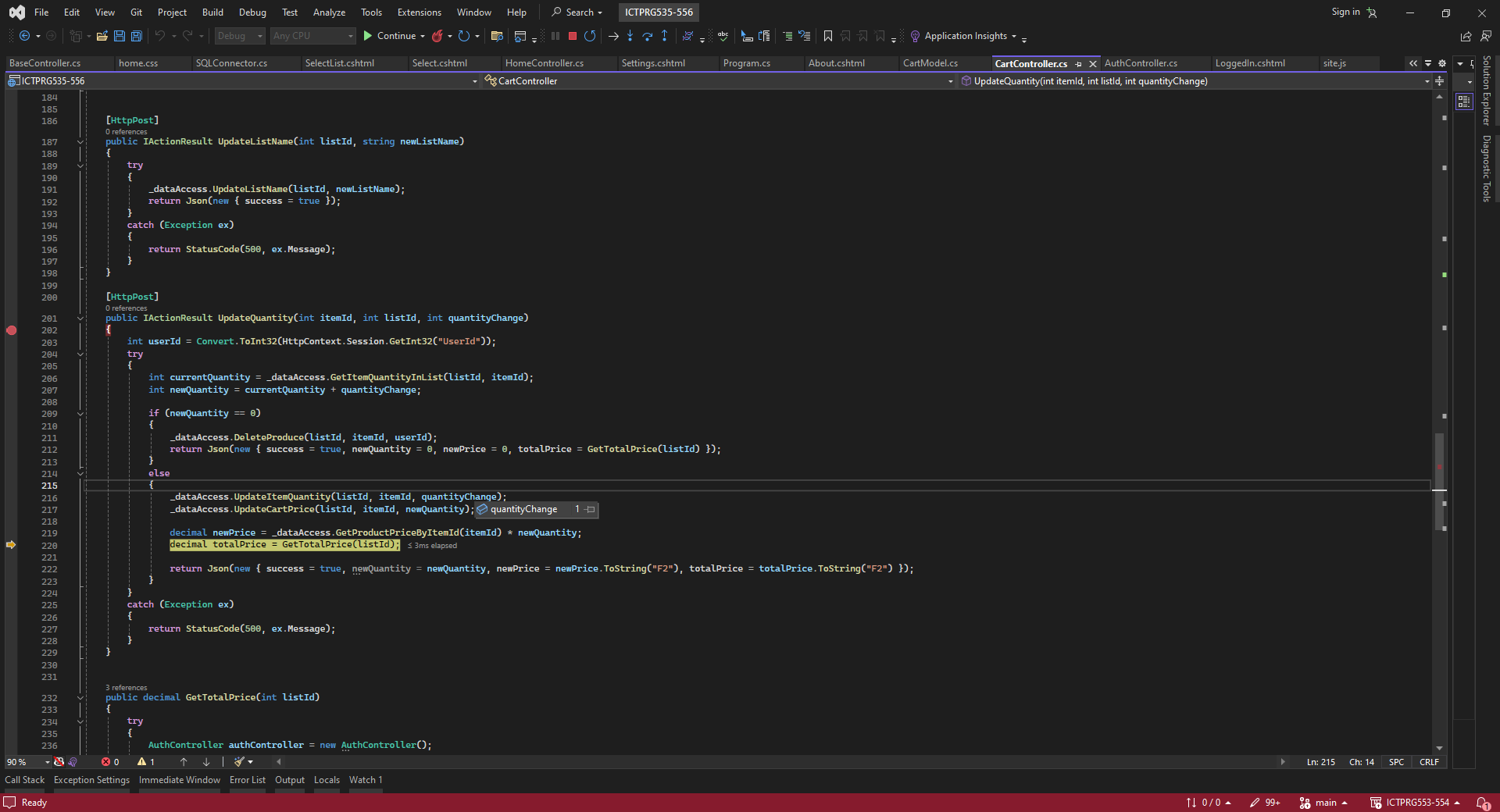
**Login**

****

**Add to cart**

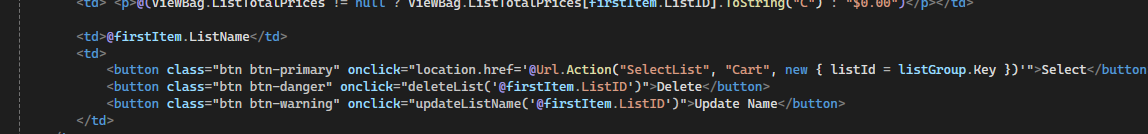


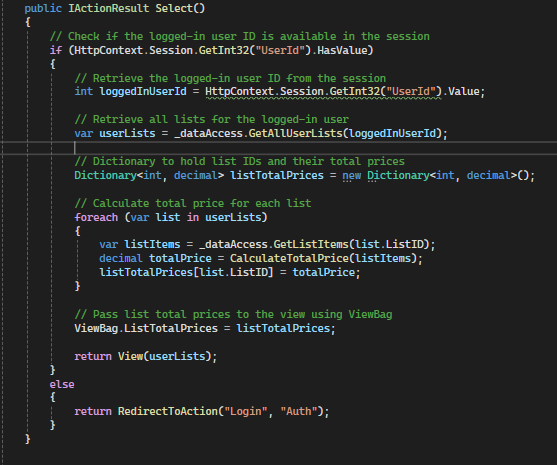
**Quantity Change**

****

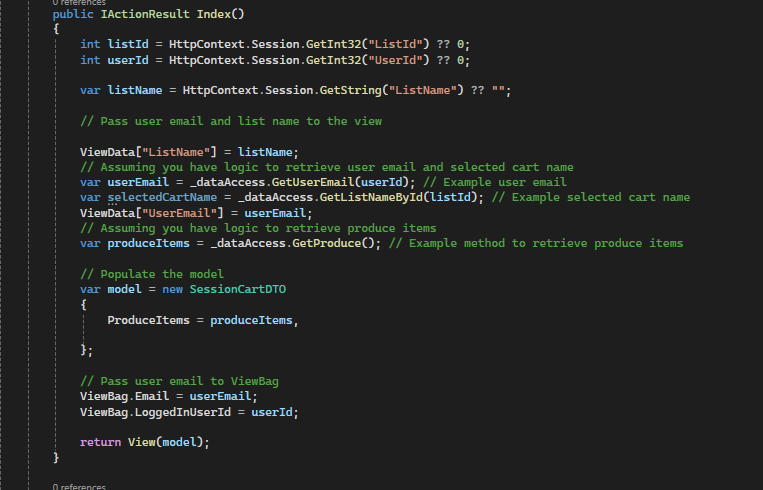
## Model, View + Controller connection Screenshots

* Screenshot of Code demonstrating navigation from one view to another (using controller methods)

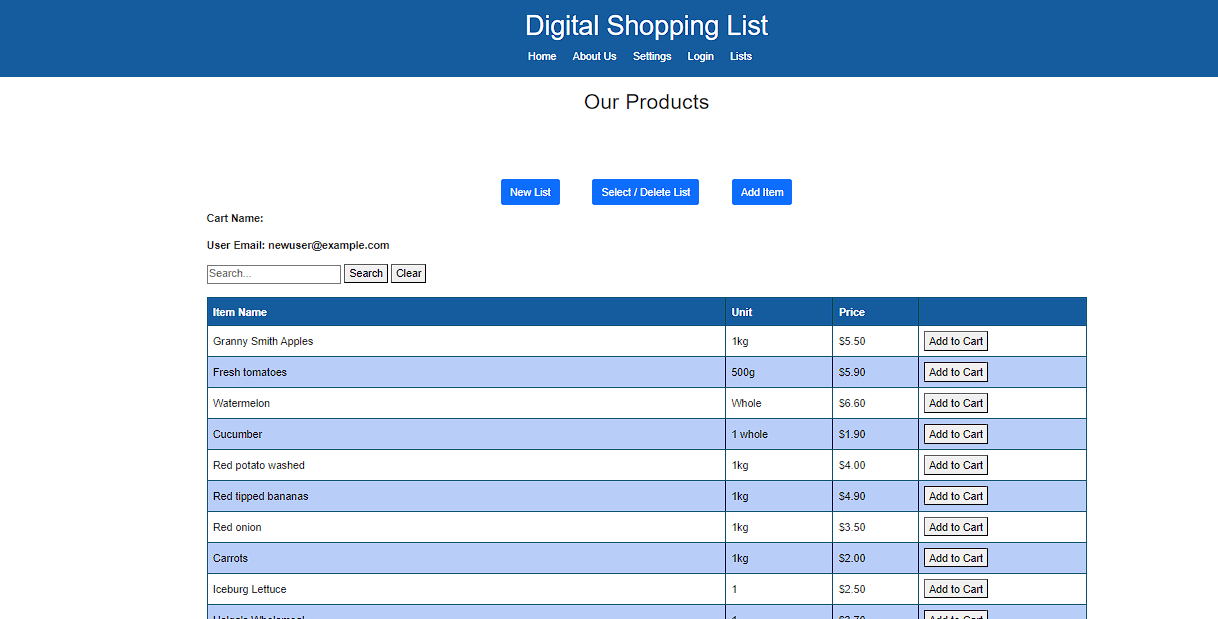




* Screenshot of source code of controller returning generated model data



* Screenshot of the view rendering the model data

****

## Client Email demonstrating developed application

From: ben@uptownit.com

Monday 08/05 12:34:56 PM

To: james.jones@shop.com

Hi James,

The application has now been completed. The application and its documentation have been sent to you, The application upon first running may need to have some additional frameworks installed, this should be automatically downloaded however feel free to let me know if there are any issues.

Regards,

Benjamin Moore

Software Developer

Uptown IT

## Feedback from the Client/Teacher

From: james.jones@shop.com

Monday 08/05 12:34:56 PM

To: ben@uptownit.com

Hi Ben,

Thankyou for sending the application over, however we have noticed that the dark mode setting doesnt stay after the user has left the page, can you please ensure that the user has this preference whenever they enter the site. Otherwise we are happy with the application and will start deployment.

Regards,

John Jones

General Manager

Uptown IT

## Adjustments made to the application to suit feedback



Java script was added to store the preference when selected. Now, as long as they dont delete their cookies the preference will stay.

# 

# Part 5 Handoff

## Client Email

From: ben@uptownit.com

Monday 08/05 12:34:56 PM

To: james.jones@shop.com

Hi James,

Thankyou for your feedback, I have now changed this so that the users dark mode preference will remain. Please note that depending on browsers it may not be saved permanently and only for users whom dont delete their cookies. This being said a majority of your users will not encounter an issue as they will be using a conventional cookie storing brower such as chrome. Upon the completion of this change the application has now completed development. Thankyou for your time throughout this project.

Regards,

Benjamin Moore

Software Developer

Uptown IT

## Sign off sheet

| **Project Name:** | Shopping List Application | | |
| --- | --- | --- | --- |
| **Project Manager:** | Troy Vaughn | | |
| **Start Date:** | 10/07/24 | | |
| **Completion Date:** | 19/07/24 | | |
| **Project Deliverables:** | | | |
| MVC .Net 8.0.4 Shopping Platform accessible through the browser and hosted on windows | | | |
| Programing documentation | | | |
| User interface design plan | | | |
| Testing reports | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
| *I acknowledge that I have submitted all the stated deliverables as per the Project Requirements and your instructions.* | | | |
| Benjamin Moore | | B.M | 19/07/24 |
| **Developer’s Name** | | **Signature** | **Date** |
| [Teachers Name] | |  |  |
| **Project Manager** | | **Signature** | **Date** |