5/4/2018

Matt Wall

N0677885

Internet Applications Programming

Dynamic Web Applications

Table of Contents

[Introduction 2](#_Toc513232629)

[Functionality 2](#_Toc513232630)

[Any Site Visitor 2](#_Toc513232631)

[Logged-in User 2](#_Toc513232632)

[Admin 2](#_Toc513232633)

[Design Areas to Consider 3](#_Toc513232634)

[Database 3](#_Toc513232635)

[Styling 3](#_Toc513232636)

[Security 4](#_Toc513232637)

[Declaration 6](#_Toc513232638)

# Introduction

During this dynamic web applications project, I have created a database driven website for a restaurant using ASP.Net. The specification had several criteria, which I believe I have completed, including features such as having system administrators being able to perform account management.

## Functionality

### Any Site Visitor

Any visitor to the website has the ability to view menu items, as well as add them to their basket. However, in order to proceed to payment they must be logged in. Additionally, when trying to view certain pages, such as ‘My Profile’ or ‘My Orders’ they will be redirected to a log in page, rather than being able to view them as there would be no information there for them.

### Logged-in User

Logged in users have the same functionality as any visitors, however they also have extra features. For example, they can view the information on their ‘My Profile’ page, as well as updating their username or password. Additionally, users will be able to look through their past orders, and proceed to checkout with current items in their basket. Furthermore, logged in users can also add or remove items from their favourites when on the menu page.

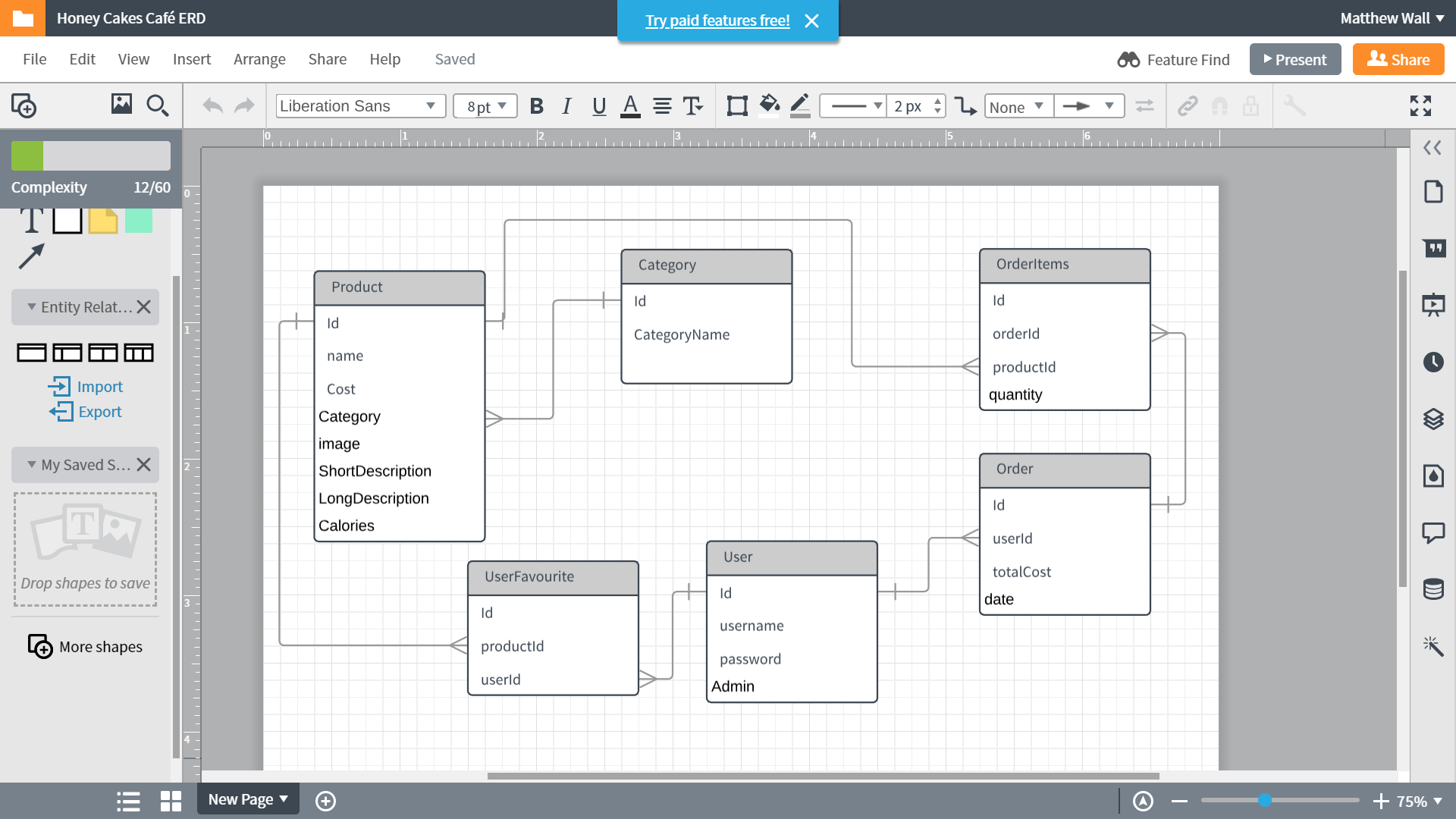
### Admin

In addition to the functions of a regular user, admin users are also able to update site content and update database information. This includes adding new users, providing admin access for existing users, adding menu items, and deleting orders. The pages available for the admin will only be able to be seen by a logged in user with admin privilege in the ‘User’ table.

# Design Areas to Consider

## Database

The planning of my database required a lot of planning to ensure that all elements would correctly connect to other tables. For example, one of the issues was that an order may have many items in it, however all the items can’t be stored in the order table. This meant using an extra table of ‘OrderItems’ to store each individual item in the order, with its quantity, and a foreign key reference to the order Id so that they could be linked. The following entity relationship diagram shows how the finalised database tables interact.



## Styling

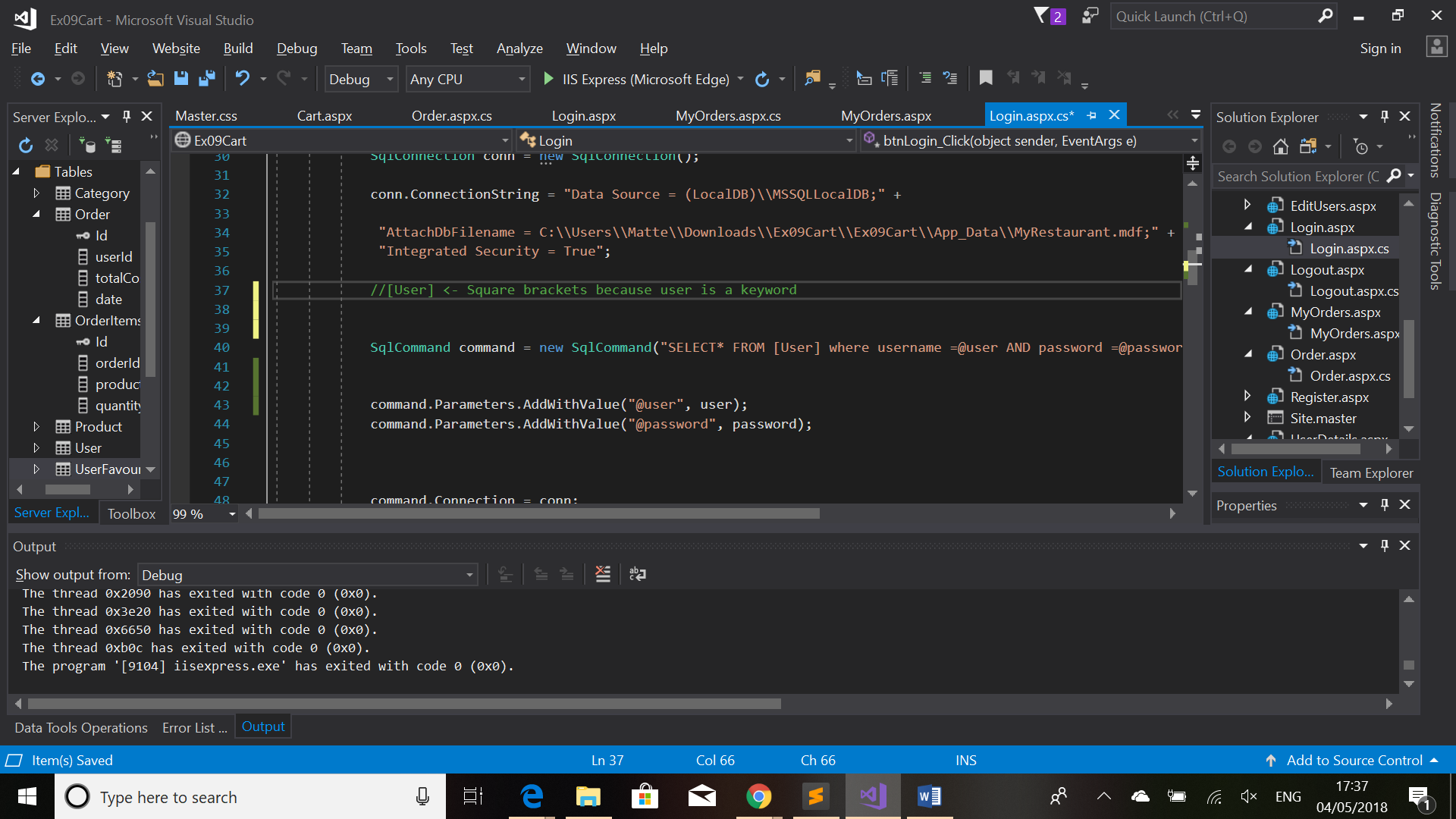
Unfortunately, due to time constraints and other areas of focus, I was unable to fully style the website as I would have liked. However, for many of the sections which I did style I used Bootstrap. Bootstrap is a CSS library which allows developers to use pre-set classes to easily create the layout and styling of a website. Additionally, the use of bootstrap helped to easily create a responsive design for many of the pages, ergo meaning the site will be more accessible as it will be displayed better on different sized devices. By using similar designs throughout the site it allows a more consistent and intuitive feel and general ease when using and navigating around the website.

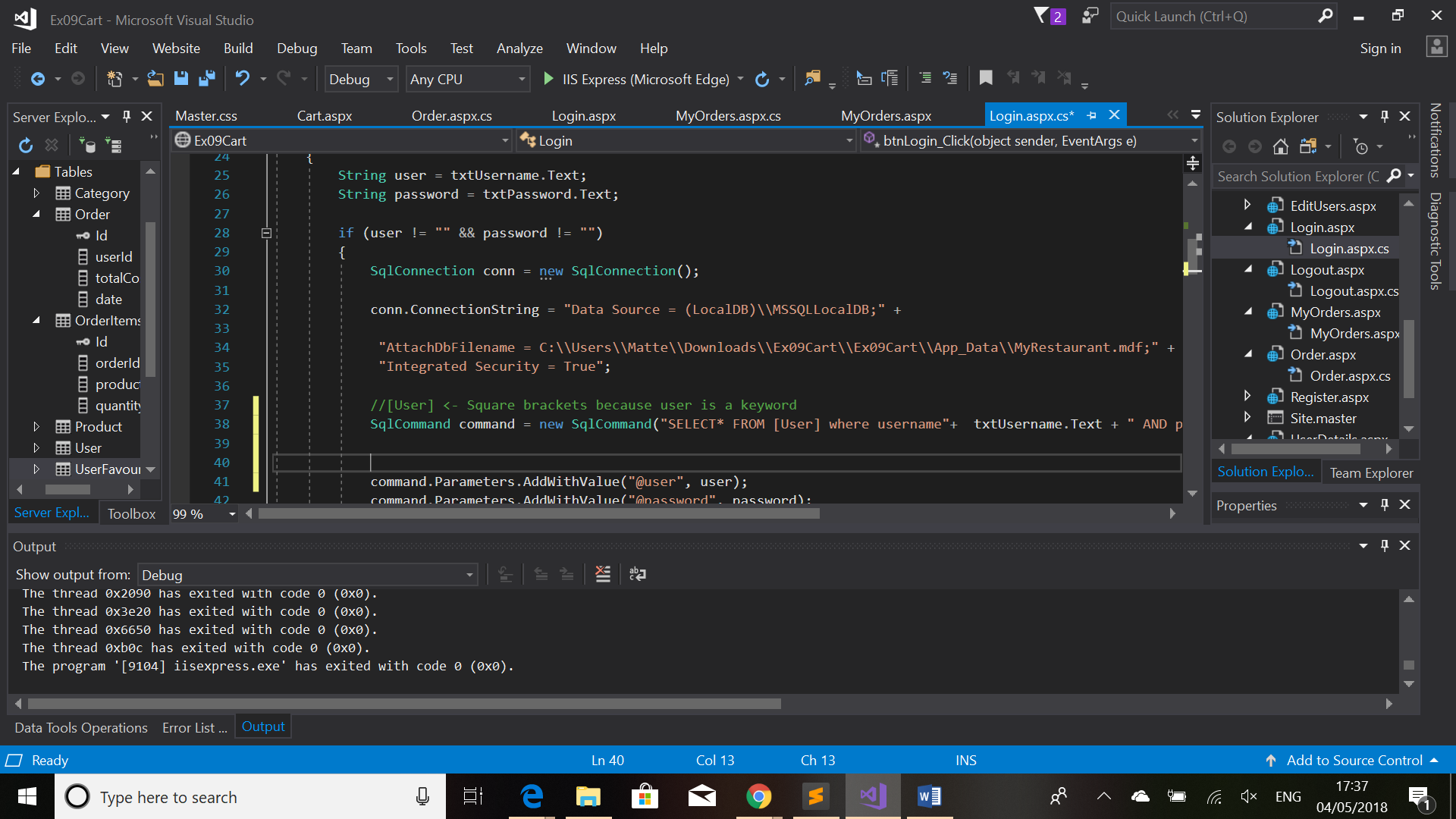
As stated in Nielsen’s 10 Usability Heuristics, “Dialogues should not contain information which is irrelevant or rarely needed” (Nielsen, 1995). By following this and ensuring that all information on the page is relevant, you can ensure that the visitors to the site will be able to find the information that they want, rather than having to search too hard or leave before finding it. This will likely lead to repeat visitors to the web page and users staying rather than searching elsewhere.

When relating to design, it is important to check that the site displays correctly in multiple browsers. For this, I used Chrome and Edge, as well as Chromes developer tools to ensure that the site would display as appropriate on a multitude of devices, including tablets and phones.

## Security

When creating a website which will be accessible by any members of the public, security is an aspect which must be taken into consideration. I have included several features to try and provide additional security for my website where possible. These include:

**SQL Injection Prevention:** Whenever there is an interaction with the database which requires user input, there will be a potential for an SQL injection attack. SQL Injection is where a user would insert malicious code in an area of the site that needs database interaction, e.g. entering a username, or inside a search function. To prevent this, anywhere the query parameters for SQL require direct input from a user, the function “Parameters.AddWithValues” is used (As seen below).

This is an alternative to directly putting the text into the SQL command, which could cause damage to the database (Such as deleting tables) if the user enters malicious code in place of an actual username or password.

**Restricting Users Privilege:** Some of the pages on the site are restricted to admin access. For this reason, anyone who is not logged will not have them appear in the navigation bar to click on. Additionally, should a user enter the URL directly to try and access a page they don’t have access to, they will be redirected when trying due to the session variable (Session[“Admin”]) being checked on Page\_Load.

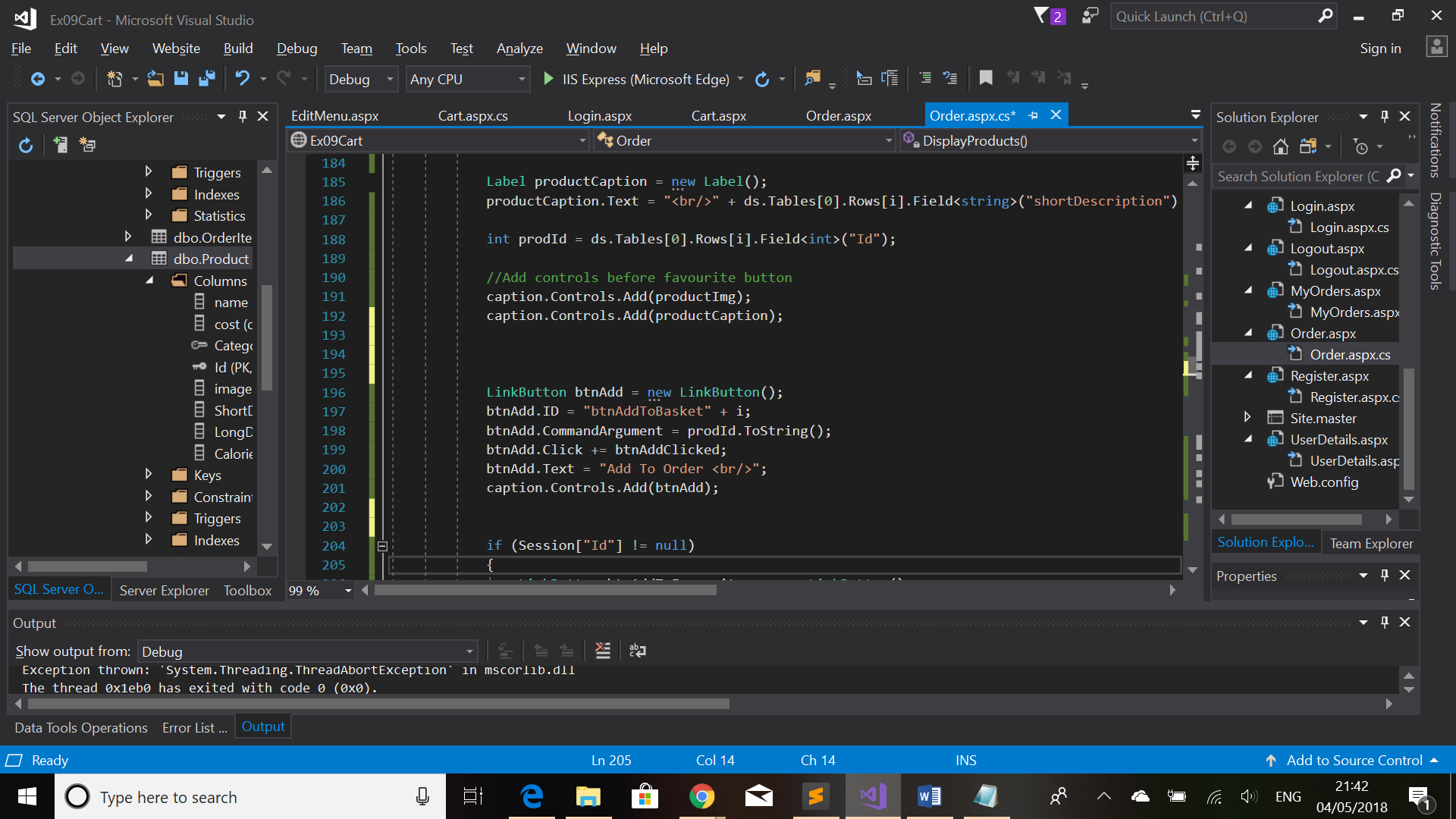
**Potential Security Improvements:** Potentially largest flaw with my websites security is that the passwords are stored in plain text. Given more time, I should have implemented the hashing of passwords so that if anyone gains access to them they will have to decrypt them before they can be used.

# Design Challenges

I encountered a myriad of challenges when creating this site which lead me to research into new ways of learning to do things to make them work

## Dynamic Table Creation

One of the biggest challenges I encountered was with creating a dynamic table. This has since been used in multiple areas of my site, however originally only appeared in the menu page. Due to wanting to add custom data and links within the table, I found that the method of using a GridView was not appropriate for my goal, so learned how to make a table dynamically with custom row classes in a C# function.   
The table creation required firstly creating the table in the aspx file, and then having a function to add new table cells to a new row, then adding those rows to the original table.

Another issue with this was creating buttons dynamically to relate to the individual table row being created. To counter this issue, I used asp Link Buttons when creating them to insert into tables, and added parameters to allow for individual functionality for each button. This image shows how the link button is created, and the product ID is then added as a command argument. This argument is used when the on-click function is called to determine the parameters required. This is now used in several areas of the site, including when adding items to favourites or to the users basket.

## Retrieving Database Information

Although I knew how to execute database queries upon starting this coursework, I did not know how to retrieve and use information, therefore I had to research to find an appropriate method. The method which I have used throughout my site is to store retrieved information in a dataset. This information can then be accessed by specifying the row, table and field, which makes it easy to use in a for loop where the table or row number can be incremented.

# Potential Improvements

The first potential improvement which I would make to the site would be to add an area for the customer to view their favourite products. Although there is an option for them to add them from the menu, there currently is no area for them to be viewed later. If I’d had more time, this area would have been included in the Profile (UserDetails.aspx) page.

Another potential improvement if I were to create this site again would be to use a function to open the database connection, rather than declaring it everywhere I have used it as it has led to make the code look messy. Although this isn’t an issue for any visitors of the site, the code has been unnecessarily repeated and makes reading it unclear.

# Declaration

I declare that all the work is my own (unless otherwise specified). All the images used, and some product descriptions, have been copied from online sources. This means they are not available to use commercially due to copyright infringement.

# References

Nielsen, J. (1995, 1 1). *10 Usability Heuristics for User Interface Design*. Retrieved from Nielsen Norman Group: https://www.nngroup.com/articles/ten-usability-heuristics/

# Bibliography

Nielsen, J. (1995, 1 1). *10 Usability Heuristics for User Interface Design*. Retrieved from Nielsen Norman Group: https://www.nngroup.com/articles/ten-usability-heuristics/

W3Schools. (n.d.). *Bootstrap 3 Tutorial*. Retrieved from W3Schools: https://www.w3schools.com/bootstrap/