Fransico’s Shoe Dreams

Jimmy Dao

MIS 325 Section 1

http://cb-ot-devst06.ad.wsu.edu/MF31jimmy.dao/Homework/FinalProject.aspx

PROBLEM

A client by the name of Fransico has come to me discussing his dreams of one day owning a big shoe store. There are numerous problems he has run into when starting his small start-up business, a lack of structure, unorganized inventory, no form of viewing data, and much more. The dis-organization to Fransico involves the loss of inventory and the inability to keep track of what is in his inventory. Fransico is forgetful and forgets what shoes he has in his inventory, how much the shoe costs, the type of shoes, branding, and much more. Furthermore, a lack of structure has been a pitiful downfall for Fransico. He has no form of business structure in terms of a digital way, relating to the dis-organization problem. Fransico has no efficient way to input and keep track of his inventory and doesn't have a centralized platform to go and view necessary details. To dive deeper into viewing necessary details, Fransico has no way to formulate, keep track, and view statistics of his business. This means Fransico can't view a table that displays how many shoes are in the inventory, total cost of inventory, the brands of shoes, and much more. Also, he doesn't have another way to view the data in terms of a graphical way. These are the major problems that Fransico has ran into and discussed with me in his journey to owning a successful shoe store.

FRANSICO (THE CUSTOMER)

Before discussing the solution that I have derived for Fransico it is important to understand the clientele that I am working with. Upon discussion with Fransico there is a lot that I learned about him. From a very young age Fransico has been passionate about shoes, to where he has built up his collection to hundreds of shoes in the current day. He had an older brother that ran a small business in re-selling shoes to his friends, but that hobby quickly died out for his brother as he experienced the same problems. However, Fransico seemed very determined to pursue this dream of his and has reached out for help. Fransico is a knowledgeable person in the business world and how a business should be run. He was a former manager at Champ Sports (a shoe outlet) and understands how a business should be run. His passion for shoes has gone beyond collecting them and has transformed into a business mindset where he believes this idea can be profitable.

SOLUTION

Upon many meetings and discussions with Fransico, I have developed a solution that solves many of Fransico's business problems. I have developed a database and a website for Fransico and his business. Within the database it stores the information of the shoe's name, cost, brand, size, release date, and date acquired. Furthermore, the database also holds summary data to keep track of how many shoes are in the inventory, total cost of the inventory, quantity of shoes within each brand, and much more. Now within the webpage design it offers a lot for Fransico and in solving his problems. I have developed a page where Fransico can input the needed details of a shoe which he can then click a button to add it to the database I created for him. He has complete control of entering the information that is needed such as the name of the shoe, cost, etc. There is also another portion on the webpage that prompts him a 'manager view.' Within this manager view Fransico is able to see the statistics/data from the input of his inventory. There are data tables provided that show the quantity of shoes within his inventory, the total cost of said inventory and much more. Also, I've implemented forms of error checking for Fransico just in case he messes up on entering the information needed for adding a shoe to the database. Furthermore, there is a security feature which is a login feature that a username and password is provided for Fransico just in case for any security breaches. All in all, the solution I have provided for Fransico solves many if not most of his business problems that he has expressed to me.

BENEFITS

With the solution provided to Fransico comes with many benefits that can help his business. To start off, with the login feature provided the possible danger of a security breach or someone accessing the webpage has dwindled down, as there needs to be a unique username and password entered to access the tools provided. Furthermore, when adding a shoe to the database there is a form of double-checking/error checking provided for Fransico. There is a button click procedure that allows Fransico to confirm the details of the shoe he is entering into the database, upon satisfying the fields required, another button appears that allows him to officially enter that shoe into the database. This is to help Fransico avoid making any errors but also making sure he inputs the correct details. Having the database and data tables provided in the solution offers Fransico a better way to keep track of things. It provides more structure, organization, and less mental stress as technology is providing a form of memory storage. In addition, there is a feature provided where Fransico can edit or delete a shoe from the database. With the slight chance that Fransico still makes an error in the double-checking stage, there is still a way to go back in and make desirable edits. This will benefit Fransico because upon making errors, there are not set and stone. Without a form/way to edit these errors Fransico would find himself back at square one, so, by being to edit the database and its rows combats that problem.

FUNCTIONALITY

This webpage offers a lot of tools that provide an overhaul of functionality. Upon opening the webpage the user is prompted with a login feature. To access the rest of the webpage and its functionality, the user must enter in a correct username and password. In this instance I have created a username (mis) and password (325) that is provided for the purpose of this assignment. After entering the right login credentials, the user is presented with three link buttons, 'Information', 'Add a Shoe', 'Summary Analytics'. Within the information view it's simply an overview of the business situation, Fransico's background, and other related information.

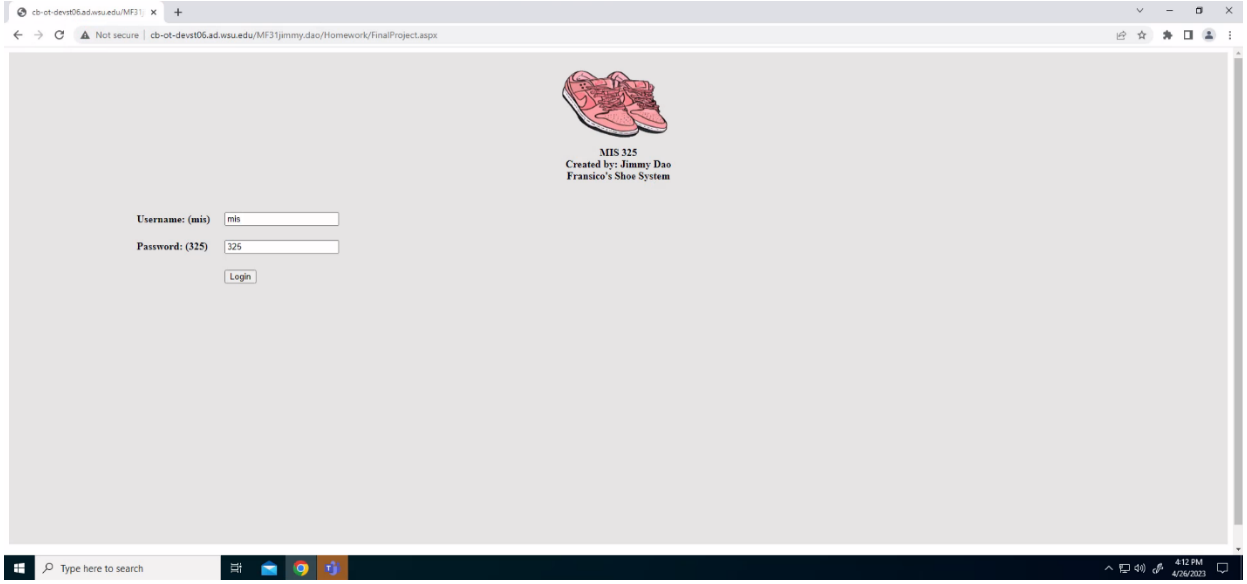
Moving forward, the next link button is 'Add a Shoe', once the user clicks this link it leads them to the main portion of the webpage. This is where Fransico is going to enter the details needed for each shoe he is going to add to the database. At first glance, there isn't much on this webpage except for a radio button list with three options. Once the user selects 'Add a Shoe' the webpage will refresh itself, and an overhaul of features will be provided. Now the user is presented with textboxes, drop down lists, buttons, and radio button lists.

To make this easier to understand, I'll go down the webpage in the appropriate order. To start off, shoe name textbox is provided and this is where the user simply types/inputs the name of the shoe they are adding. For example, Fransico just obtained a pair of Kobe VI basketball shoes, so, in the shoe name textbox he would simply type out "Kobe VI." Now, moving down the webpage, there is a drop-down list that prompts the user to select a brand that the shoe belongs to. Continuing the example, Fransico would choose "Nike," as Kobe's are a Nike branded shoe. Then, the next area that user is prompted to fill out is the shoe type and the shoe's condition. So, in this case Fransico would select "Athletic," and "New." Moving forward, there are four more textboxes that take in user input which would be the size, cost, release date, and the date acquired of the shoe. So, Fransico would input 10 as the size, $165 for the cost, 4/2/20 for the release date, and 4/23/23 for the acquired date of the shoe. Now, the user is ready to click on the button that says "Confirm Shoe Selection," which is a form to double check that all the details are corrected. Upon clicking the button, the textbox outputs all the user entry details for the user to overlook and make sure everything seems right. It is also important to note, if a user leaves a field empty and clicks the button, the website will not output anything but rather it gives an error message on the top of the webpage. Now, after a successful confirmation, a new button appears and it states "Add to Inventory." Which, will simply take all the fields of entered information and then transfer it into the database, which a grid view is present on the webpage to see all the shoes added to the inventory. Discussed here is how Fransico or a user can utilize the functionality to add a shoe to the database/inventory system.

Now, utilizing the edit a row/shoe or deleting a row/shoe is a simple functionality to grasp. Using the radio button list at the top of the webpage, Fransico or the user can simply to enter either edit or delete mode. So, within the ‘Edit a Shoe,’ view the webpage is further transformed. Now, a drop down list appears, which is filled with the ShoeID’s all of the shoes that have been entered into the database. Here, the user can select a shoe that they want to modify. For example, say Fransico wants to edit a shoe, he would simply select the edit option and proceed to the drop down list to select a shoe. So, say he selects the shoe he added into the database from the previous example. All the webpage fields for shoe details would be automatically filled in with the corresponding shoe selected from the drop down list. From, there Fransico or the user can make their desirable edits. After, they are satisfied with their edits a new button is introduced that replaces the add to inventory button. The update inventory button is solely to update that single row within the database, in other words this button will update that one shoe’s information/details. Furthermore, the delete functionality is even more simple. All the user has to do is select the delete mode, utilize the drop down list to select the shoe they want to delete, and proceed to the new button ‘DELETE A SHOE’, which will execute SQL commands to remove that row of data from the database.

Lastly, the last functionality that is provided within this webpage would be the summary data. The user can navigate to the summary data page by clicking on the ‘Summary Analytics’, link button at the top of the webpage. Within this view of the webpage, three data tables are displayed with analytical data provided to analyze. These tables are automatically updated when the user adds a shoe to the database from the previous functionality. So, there isn’t much required of the user to do anything to make these tables update with new numbers.

WEBPAGE

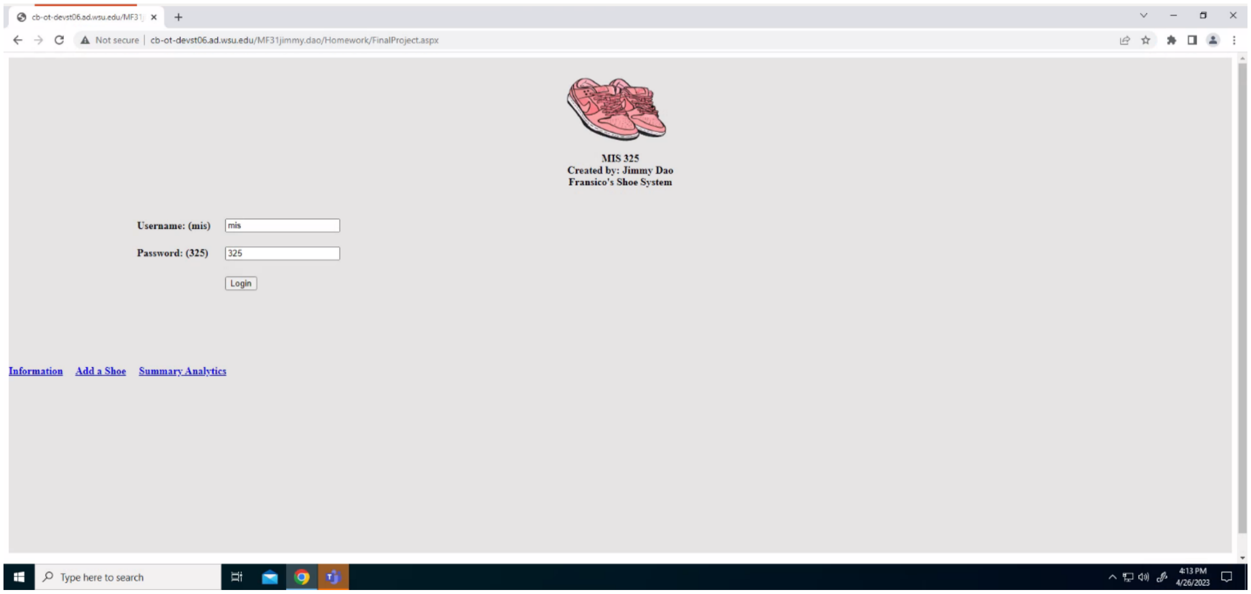

cb-ot-devst06.ad.wsu.edu/MF31j
Not secure | cb-ot-devst06.ad.wsu.edu/MF31jimmy.dao/Homework/FinalProject.aspx
Type here to search
4:12 PM
4/26/2023

MIS 325
Created by: Jimmy Dao
Fransico's Shoe System

Username: (mis) mis

Password: (325) 325

Login


cb-ot-devst06.ad.wsu.edu/MF31j
Not secure | cb-ot-devst06.ad.wsu.edu/MF31jimmy.dao/Homework/FinalProject.aspx
Type here to search
4:13 PM
4/26/2023

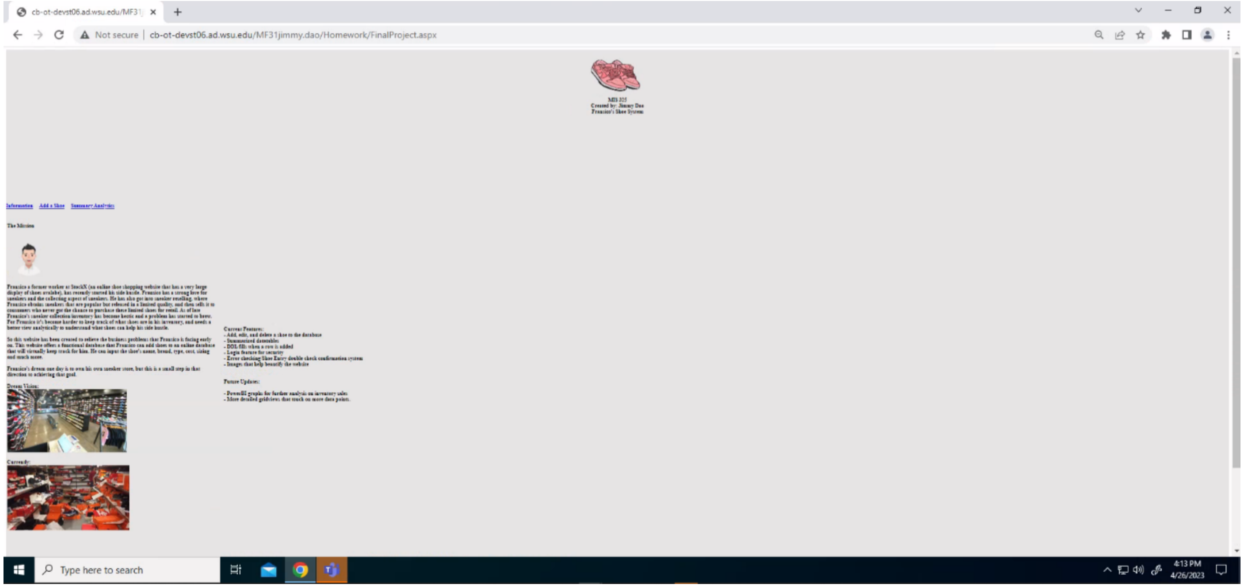
MIS 325
Created by: Jimmy Dao
Fransico's Shoe System

Username: (mis) mis

Password: (325) 325

Login

Information Add a Shoe Summary Analytics


cb-ot-devst06.ad.wsu.edu/MF31j
Not secure
cb-ot-devst06.ad.wsu.edu/MF31jimmy.dao/Homework/FinalProject.aspx
Type here to search
4:13 PM
4/26/2023

MIS 325
Created by: Jimmy Dao
Fransico's Shoe System

Information Add a Shoe Summary Analytics

The Mission

Fransico a former worker at StockX (an online shoe shopping website that has a very large display of shoes avalible), has recently started his side hustle. Fransico has a strong love for sneakers and the collecting aspect of sneakers. He has also got into sneaker reselling, where Fransico obtains sneakers that are popular but released in a limited quality, and then sells it to consumers who never got the chance to purchase these limited shoes for retail. As of late Fransico's sneaker collection inventory has become hectic and a problem has started to brew. For Fransico it's become harder to keep track of what shoes are in his inventory, and needs a better view analytically to understand what shoes can help his side hustle.

So this website has been created to relieve the business problems that Fransico is facing early on. This website offers a functional database that Fransico can add shoes to an online database that will virtually keep track for him. He can input the shoe's name, brand, type, cost, sizing and much more.

Fransico's dream one day is to own his own sneaker store, but this is a small step in that direction to achieving that goal.

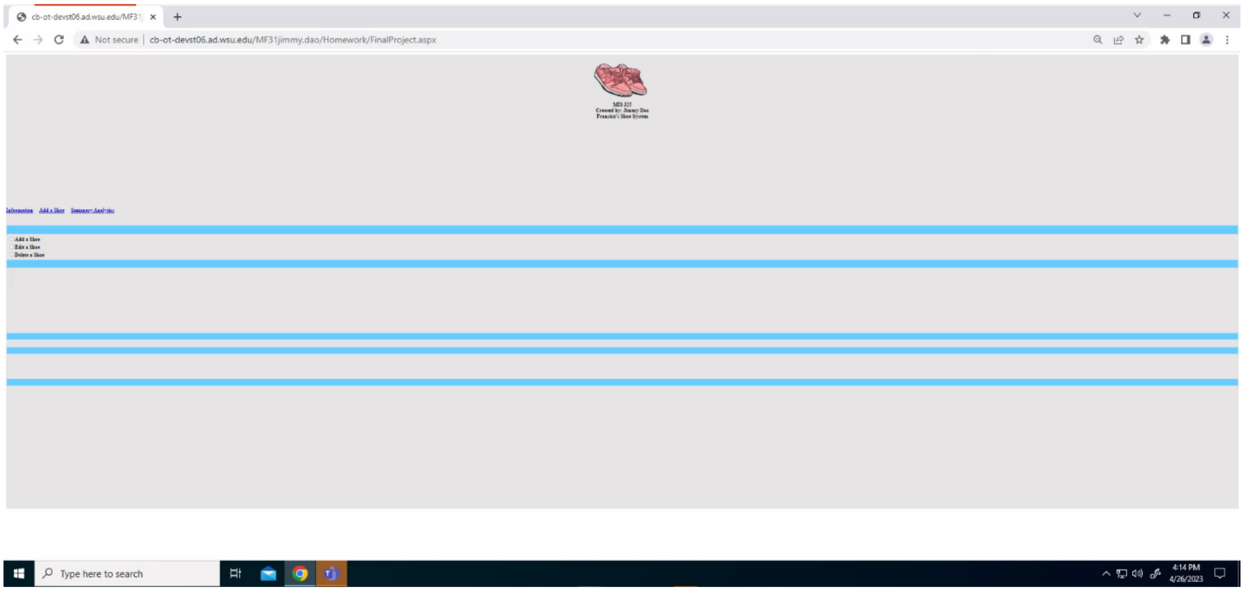
Dream Vision:

Currently:

Current Features:
- Add, edit, and delete a shoe to the database
- Summarized datatables
- DDL fills when a row is added
- Login feature for security
- Error checking Shoe Entry double check confirmation system
- Images that help beautify the website

Future Updates:

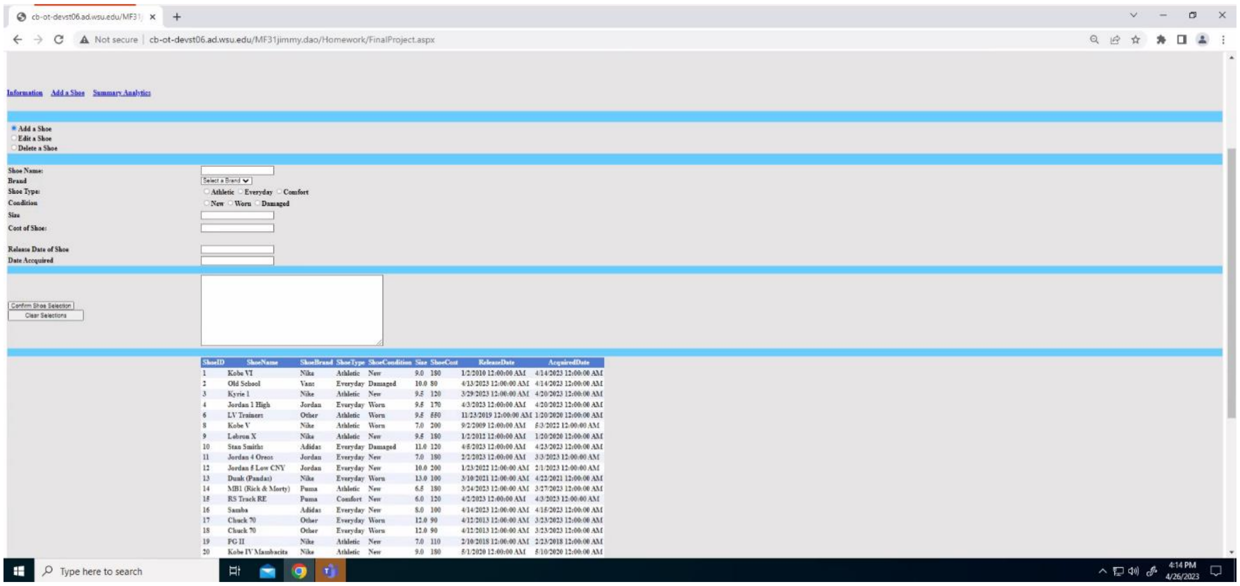
- PowerBI graphs for further analysis on inventory sales
- More detailed gridviews that touch on more data points.


cb-ot-devst06.ad.wsu.edu/MF31j
Not secure
cb-ot-devst06.ad.wsu.edu/MF31jimmy.dao/Homework/FinalProject.aspx
Type here to search
4:14 PM
4/26/2023

MIS 325
Created by: Jimmy Dao
Fransico's Shoe System

Information Add a Shoe Summary Analytics

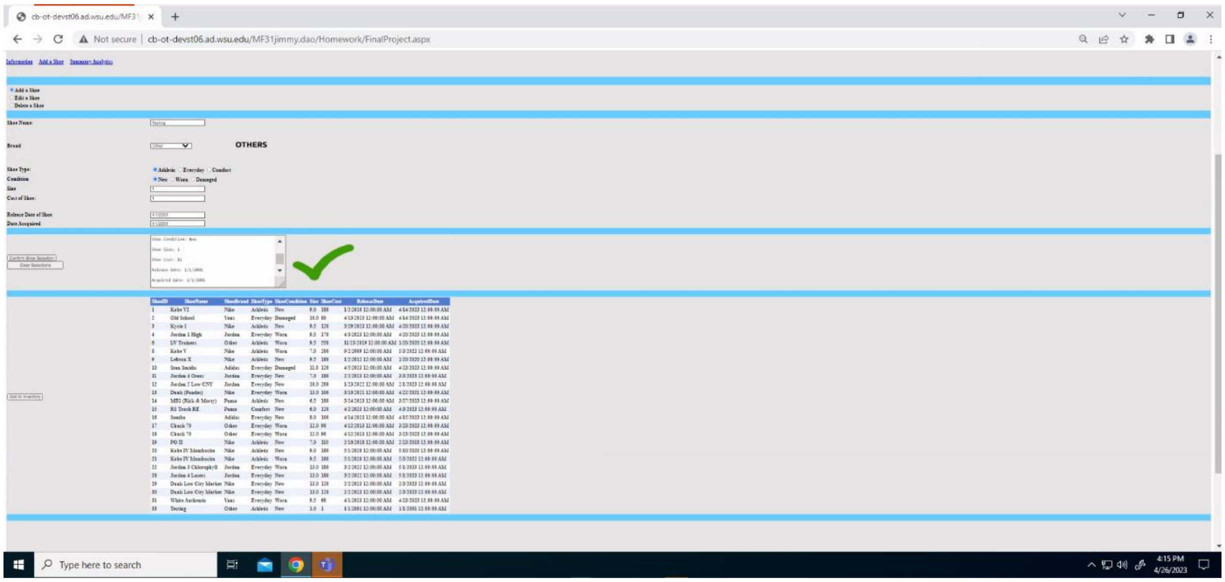
Add a Shoe
Edit a Shoe
Delete a Shoe



● Add a Shoe
○ Edit a Shoe
○ Delete a Shoe

Shoe Name:	
Brand	Select a Brand
Shoe Type:	○ Athletic ○ Everyday ○ Comfort
Condition	○ New ○ Worn ○ Damaged
Size	
Cost of Shoe:	
Release Date of Shoe	
Date Acquired	

ShoeID	ShoeName	ShoeBrand	ShoeType	ShoeCondition	Size	ShoeCost	ReleaseDate	AcquiredDate
1	Kobe VI	Nike	Athletic	New	9.0	180	1/2/2010 12:00:00 AM	4/14/2023 12:00:00 AM
2	Old School	Vans	Everyday	Damaged	10.0	80	4/13/2023 12:00:00 AM	4/14/2023 12:00:00 AM
3	Kyrie 1	Nike	Athletic	New	9.5	120	3/29/2023 12:00:00 AM	4/20/2023 12:00:00 AM
4	Jordan 1 High	Jordan	Everyday	Worn	9.5	170	4/3/2023 12:00:00 AM	4/20/2023 12:00:00 AM
6	LV Trainers	Other	Athletic	Worn	9.5	550	11/23/2019 12:00:00 AM	1/20/2020 12:00:00 AM
8	Kobe V	Nike	Athletic	Worn	7.0	200	9/2/2009 12:00:00 AM	5/3/2022 12:00:00 AM
9	Lebron X	Nike	Athletic	New	9.5	180	1/2/2012 12:00:00 AM	1/20/2020 12:00:00 AM
10	Stan Smiths	Adidas	Everyday	Damaged	11.0	120	4/5/2023 12:00:00 AM	4/23/2023 12:00:00 AM
11	Jordan 4 Oreos	Jordan	Everyday	New	7.0	180	2/2/2023 12:00:00 AM	3/3/2023 12:00:00 AM
12	Jordan 5 Low CNY	Jordan	Everyday	New	10.0	200	1/23/2022 12:00:00 AM	2/1/2023 12:00:00 AM
13	Dunk (Pandas)	Nike	Everyday	Worn	13.0	100	3/10/2021 12:00:00 AM	4/22/2021 12:00:00 AM
14	MB1 (Rick & Morty)	Puma	Athletic	New	6.5	180	3/24/2023 12:00:00 AM	3/27/2023 12:00:00 AM
15	RS Track RE	Puma	Comfort	New	6.0	120	4/2/2023 12:00:00 AM	4/3/2023 12:00:00 AM
16	Samba	Adidas	Everyday	New	8.0	100	4/14/2023 12:00:00 AM	4/15/2023 12:00:00 AM
17	Chuck 70	Other	Everyday	Worn	12.0	90	4/12/2013 12:00:00 AM	3/23/2023 12:00:00 AM
18	Chuck 70	Other	Everyday	Worn	12.0	90	4/12/2013 12:00:00 AM	3/23/2023 12:00:00 AM
19	PG II	Nike	Athletic	New	7.0	110	2/10/2018 12:00:00 AM	2/23/2018 12:00:00 AM
20	Kobe IV Mambacita	Nike	Athletic	New	9.0	180	5/1/2020 12:00:00 AM	5/10/2020 12:00:00 AM



● Add a Shoe
○ Edit a Shoe
○ Delete a Shoe

Shoe Name:	Testing
Brand	Other
Shoe Type:	● Athletic ○ Everyday ○ Comfort
Condition	● New ○ Worn ○ Damaged
Size	1
Cost of Shoe:	1
Release Date of Shoe	1/1/2001
Date Acquired	1/1/2001

OTHERS

Shoe Condition: New

Shoe Size: 1

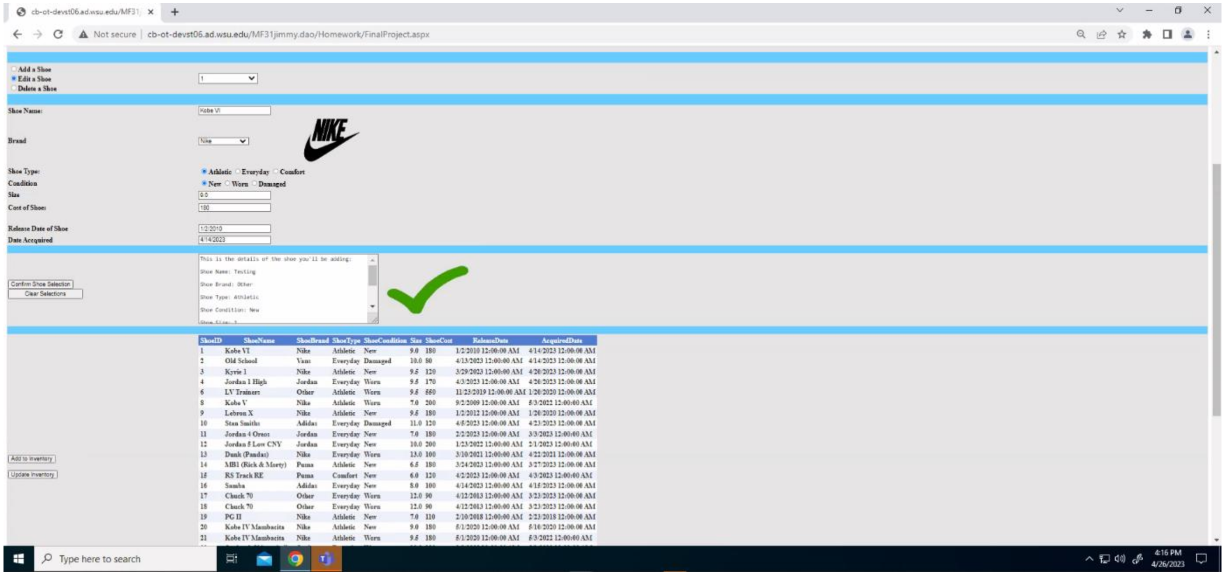
Shoe Cost: $1

Release Date: 1/1/2001

Acquired Date: 1/1/2001

✓

ShoeID	ShoeName	ShoeBrand	ShoeType	ShoeCondition	Size	ShoeCost	ReleaseDate	AcquiredDate
1	Kobe VI	Nike	Athletic	New	9.0	180	1/2/2010 12:00:00 AM	4/14/2023 12:00:00 AM
2	Old School	Vans	Everyday	Damaged	10.0	80	4/13/2023 12:00:00 AM	4/14/2023 12:00:00 AM
3	Kyrie 1	Nike	Athletic	New	9.5	120	3/29/2023 12:00:00 AM	4/20/2023 12:00:00 AM
4	Jordan 1 High	Jordan	Everyday	Worn	9.5	170	4/3/2023 12:00:00 AM	4/20/2023 12:00:00 AM
6	LV Trainers	Other	Athletic	Worn	9.5	550	11/23/2019 12:00:00 AM	1/20/2020 12:00:00 AM
8	Kobe V	Nike	Athletic	Worn	7.0	200	9/2/2009 12:00:00 AM	5/3/2022 12:00:00 AM
9	Lebron X	Nike	Athletic	New	9.5	180	1/2/2012 12:00:00 AM	1/20/2020 12:00:00 AM
10	Stan Smiths	Adidas	Everyday	Damaged	11.0	120	4/5/2023 12:00:00 AM	4/23/2023 12:00:00 AM
11	Jordan 4 Oreos	Jordan	Everyday	New	7.0	180	2/2/2023 12:00:00 AM	3/3/2023 12:00:00 AM
12	Jordan 5 Low CNY	Jordan	Everyday	New	10.0	200	1/23/2022 12:00:00 AM	2/1/2023 12:00:00 AM
13	Dunk (Pandas)	Nike	Everyday	Worn	13.0	100	3/10/2021 12:00:00 AM	4/22/2021 12:00:00 AM
14	MB1 (Rick & Morty)	Puma	Athletic	New	6.5	180	3/24/2023 12:00:00 AM	3/27/2023 12:00:00 AM
15	RS Track RE	Puma	Comfort	New	6.0	120	4/2/2023 12:00:00 AM	4/3/2023 12:00:00 AM
16	Samba	Adidas	Everyday	New	8.0	100	4/14/2023 12:00:00 AM	4/15/2023 12:00:00 AM
17	Chuck 70	Other	Everyday	Worn	12.0	90	4/12/2013 12:00:00 AM	3/23/2023 12:00:00 AM
18	Chuck 70	Other	Everyday	Worn	12.0	90	4/12/2013 12:00:00 AM	3/23/2023 12:00:00 AM
19	PG II	Nike	Athletic	New	7.0	110	2/10/2018 12:00:00 AM	2/23/2018 12:00:00 AM
20	Kobe IV Mambacita	Nike	Athletic	New	9.0	180	5/1/2020 12:00:00 AM	5/10/2020 12:00:00 AM
21	Kobe IV Mambacita	Nike	Athletic	Worn	9.5	180	5/1/2020 12:00:00 AM	5/3/2022 12:00:00 AM
22	Jordan 3 Chlorophyll	Jordan	Everyday	Worn	10.0	180	3/2/2022 12:00:00 AM	5/1/2023 12:00:00 AM
28	Jordan 4 Lasers	Jordan	Everyday	New	10.0	180	3/2/2022 12:00:00 AM	5/1/2023 12:00:00 AM
29	Dunk Low City Market	Nike	Everyday	New	10.0	120	2/2/2023 12:00:00 AM	2/3/2023 12:00:00 AM
30	Dunk Low City Market	Nike	Everyday	New	10.0	120	2/2/2023 12:00:00 AM	2/3/2023 12:00:00 AM
31	White Authentic	Vans	Everyday	Worn	9.5	60	4/1/2023 12:00:00 AM	4/23/2023 12:00:00 AM
33	Testing	Other	Athletic	New	1.0	1	1/1/2001 12:00:00 AM	1/1/2001 12:00:00 AM


cb-ot-devst06.ad.wsu.edu/MF31
Not secure | cb-ot-devst06.ad.wsu.edu/MF31jimmy.dao/Homework/FinalProject.aspx
Confirm Shoe Selection
Clear Selections
Add to Inventory
Update Inventory
Type here to search
4:16 PM
4/26/2023

○ Add a Shoe
● Edit a Shoe
○ Delete a Shoe

1

Shoe Name: Kobe VI

Brand: Nike

Shoe Type: ● Athletic ○ Everyday ○ Comfort

Condition: ● New ○ Worn ○ Damaged

Size: 9.0

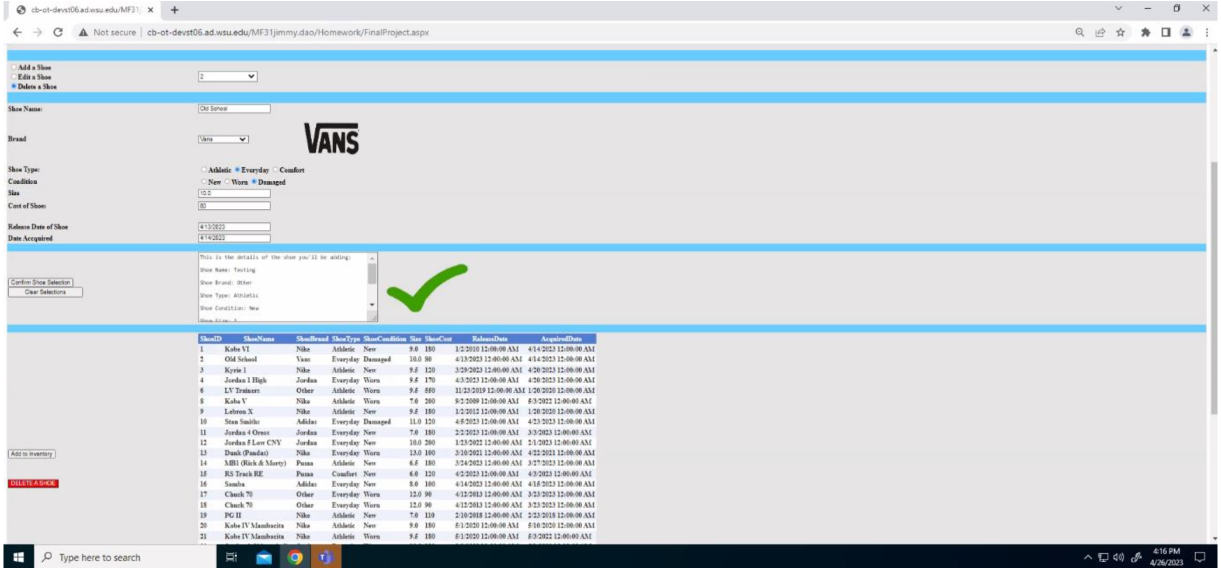
Cost of Shoe: 180

Release Date of Shoe: 1/2/2010

Date Acquired: 4/14/2023

```
This is the details of the shoe you'll be adding:

Shoe Name: Testing

Shoe Brand: Other

Shoe Type: Athletic

Shoe Condition: New
```

ShoeID	ShoeName	ShoeBrand	ShoeType	ShoeCondition	Size	ShoeCost	ReleaseDate	AcquiredDate
1	Kobe VI	Nike	Athletic	New	9.0	180	1/2/2010 12:00:00 AM	4/14/2023 12:00:00 AM
2	Old School	Vans	Everyday	Damaged	10.0	80	4/13/2023 12:00:00 AM	4/14/2023 12:00:00 AM
3	Kyrie 1	Nike	Athletic	New	9.5	120	3/29/2023 12:00:00 AM	4/20/2023 12:00:00 AM
4	Jordan 1 High	Jordan	Everyday	Worn	9.5	170	4/3/2023 12:00:00 AM	4/20/2023 12:00:00 AM
6	LV Trainers	Other	Athletic	Worn	9.5	560	11/23/2019 12:00:00 AM	1/20/2020 12:00:00 AM
8	Kobe V	Nike	Athletic	Worn	7.0	200	9/2/2009 12:00:00 AM	5/3/2022 12:00:00 AM
9	Lebron X	Nike	Athletic	New	9.5	180	1/2/2012 12:00:00 AM	1/20/2020 12:00:00 AM
10	Stan Smiths	Adidas	Everyday	Damaged	11.0	120	4/5/2023 12:00:00 AM	4/23/2023 12:00:00 AM
11	Jordan 4 Oreos	Jordan	Everyday	New	7.0	180	2/2/2023 12:00:00 AM	3/3/2023 12:00:00 AM
12	Jordan 5 Low CNY	Jordan	Everyday	New	10.0	200	1/23/2022 12:00:00 AM	2/1/2023 12:00:00 AM
13	Dunk (Pandas)	Nike	Everyday	Worn	13.0	100	3/10/2021 12:00:00 AM	4/22/2021 12:00:00 AM
14	MB1 (Rick & Morty)	Puma	Athletic	New	6.5	180	3/24/2023 12:00:00 AM	3/27/2023 12:00:00 AM
15	RS Track RE	Puma	Comfort	New	6.0	120	4/2/2023 12:00:00 AM	4/3/2023 12:00:00 AM
16	Samba	Adidas	Everyday	New	8.0	100	4/14/2023 12:00:00 AM	4/15/2023 12:00:00 AM
17	Chuck 70	Other	Everyday	Worn	12.0	90	4/12/2013 12:00:00 AM	3/23/2023 12:00:00 AM
18	Chuck 70	Other	Everyday	Worn	12.0	90	4/12/2013 12:00:00 AM	3/23/2023 12:00:00 AM
19	PG II	Nike	Athletic	New	7.0	110	2/10/2018 12:00:00 AM	2/23/2018 12:00:00 AM
20	Kobe IV Mambacita	Nike	Athletic	New	9.0	180	5/1/2020 12:00:00 AM	5/10/2020 12:00:00 AM
21	Kobe IV Mambacita	Nike	Athletic	Worn	9.5	180	5/1/2020 12:00:00 AM	5/3/2022 12:00:00 AM


cb-ot-devst06.ad.wsu.edu/MF31
Not secure | cb-ot-devst06.ad.wsu.edu/MF31jimmy.dao/Homework/FinalProject.aspx
Confirm Shoe Selection
Clear Selections
Add to Inventory
DELETE A SHOE
Type here to search
4:16 PM
4/26/2023

○ Add a Shoe
○ Edit a Shoe
● Delete a Shoe

2

Shoe Name: Old School

Brand: Vans

Shoe Type: ○ Athletic ● Everyday ○ Comfort

Condition: ○ New ○ Worn ● Damaged

Size: 10.0

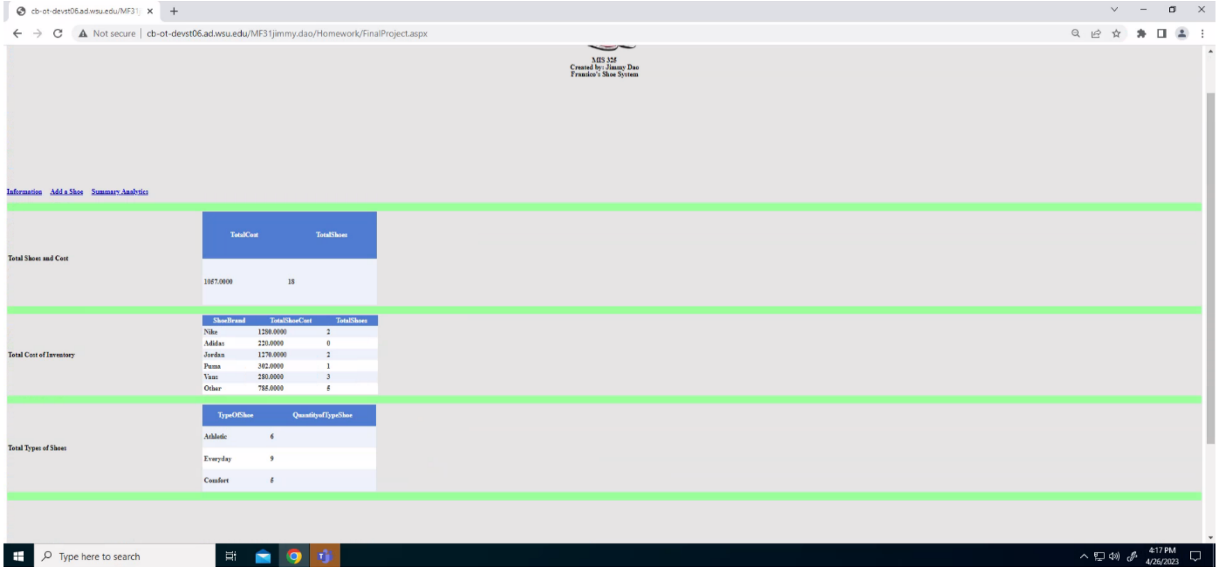
Cost of Shoe: 80

Release Date of Shoe: 4/13/2023

Date Acquired: 4/14/2023

```
This is the details of the shoe you'll be adding:

Shoe Name: Testing

Shoe Brand: Other

Shoe Type: Athletic

Shoe Condition: New
```

ShoeID	ShoeName	ShoeBrand	ShoeType	ShoeCondition	Size	ShoeCost	ReleaseDate	AcquiredDate
1	Kobe VI	Nike	Athletic	New	9.0	180	1/2/2010 12:00:00 AM	4/14/2023 12:00:00 AM
2	Old School	Vans	Everyday	Damaged	10.0	80	4/13/2023 12:00:00 AM	4/14/2023 12:00:00 AM
3	Kyrie 1	Nike	Athletic	New	9.5	120	3/29/2023 12:00:00 AM	4/20/2023 12:00:00 AM
4	Jordan 1 High	Jordan	Everyday	Worn	9.5	170	4/3/2023 12:00:00 AM	4/20/2023 12:00:00 AM
6	LV Trainers	Other	Athletic	Worn	9.5	560	11/23/2019 12:00:00 AM	1/20/2020 12:00:00 AM
8	Kobe V	Nike	Athletic	Worn	7.0	200	9/2/2009 12:00:00 AM	5/3/2022 12:00:00 AM
9	Lebron X	Nike	Athletic	New	9.5	180	1/2/2012 12:00:00 AM	1/20/2020 12:00:00 AM
10	Stan Smiths	Adidas	Everyday	Damaged	11.0	120	4/5/2023 12:00:00 AM	4/23/2023 12:00:00 AM
11	Jordan 4 Oreos	Jordan	Everyday	New	7.0	180	2/2/2023 12:00:00 AM	3/3/2023 12:00:00 AM
12	Jordan 5 Low CNY	Jordan	Everyday	New	10.0	200	1/23/2022 12:00:00 AM	2/1/2023 12:00:00 AM
13	Dunk (Pandas)	Nike	Everyday	Worn	13.0	100	3/10/2021 12:00:00 AM	4/22/2021 12:00:00 AM
14	MB1 (Rick & Morty)	Puma	Athletic	New	6.5	180	3/24/2023 12:00:00 AM	3/27/2023 12:00:00 AM
15	RS Track RE	Puma	Comfort	New	6.0	120	4/2/2023 12:00:00 AM	4/3/2023 12:00:00 AM
16	Samba	Adidas	Everyday	New	8.0	100	4/14/2023 12:00:00 AM	4/15/2023 12:00:00 AM
17	Chuck 70	Other	Everyday	Worn	12.0	90	4/12/2013 12:00:00 AM	3/23/2023 12:00:00 AM
18	Chuck 70	Other	Everyday	Worn	12.0	90	4/12/2013 12:00:00 AM	3/23/2023 12:00:00 AM
19	PG II	Nike	Athletic	New	7.0	110	2/10/2018 12:00:00 AM	2/23/2018 12:00:00 AM
20	Kobe IV Mambacita	Nike	Athletic	New	9.0	180	5/1/2020 12:00:00 AM	5/10/2020 12:00:00 AM
21	Kobe IV Mambacita	Nike	Athletic	Worn	9.5	180	5/1/2020 12:00:00 AM	5/3/2022 12:00:00 AM


cb-ot-devst06.ad.wsu.edu/MF31j
Not secure | cb-ot-devst06.ad.wsu.edu/MF31jimmy.dao/Homework/FinalProject.aspx

MIS 325
Created by: Jimmy Dao
Fransico's Shoe System

Information Add a Shoe Summary Analytics

Total Shoes and Cost

TotalCost	TotalShoes
1057.0000	18

Total Cost of Inventory

ShoeBrand	TotalShoeCost	TotalShoes
Nike	1280.0000	2
Adidas	220.0000	0
Jordan	1270.0000	2
Puma	302.0000	1
Vans	280.0000	3
Other	785.0000	5

Total Types of Shoes

TypeOfShoe	QuantityofTypeShoe
Athletic	6
Everyday	9
Comfort	5

APPENDIX

Unique/New Features I added:

- Log in function
    - Dissappears when user selects a link button
- Images that appear for each brand and upon successful user entry
- Error checking that doesn’t allow user to add to database unless fields are correct

```
Imports System.Data
Imports System.Data.SqlClient
Partial Class Homework_FinalProject
    Inherits System.Web.UI.Page


    Public Shared con As New SqlConnection("Data source = cb-ot-devst06.ad.wsu.edu; initial catalog = MF31jimmy.dao; Persist Security Info = True;User ID = jimmy.dao;Password=d3982pw2")


    Public Shared gdaLogin As New SqlDataAdapter("SELECT * FROM RegisteredUsers", con)


    'Setting up for Shoe Inventory Database (Adding a row of data)
    Public Shared daShoeInv As New SqlDataAdapter("SELECT * FROM ShoeInventory", con)
    Public Shared cbShoeInv As New SqlCommandBuilder(daShoeInv)
    Public Shared dtShoeInv As New DataTable


    'Setting up to edit/retrieve one row of data

    Public Shared da1Shoe As New SqlDataAdapter("SELECT * FROM ShoeInventory WHERE ShoeID = @p1", con)

    Public Shared dt1Shoe As New DataTable


    'Setting up to have running totals for summary data

    Public Shared daTotalInvCost As New SqlDataAdapter("SELECT * FROM ShoeTotalSummary", con)

    Public Shared dtTotalInvCost As New DataTable

    Public Shared daTotalShoeBrand As New SqlDataAdapter("SELECT * FROM ShoeBrandSummary", con)

    Public Shared dtTotalShoeBrand As New DataTable

    Public Shared daTotalShoeType As New SqlDataAdapter("SELECT * FROM ShoeTypeSummary", con)

    Public Shared dtTotalShoeType As New DataTable

    Public Shared daTotals As New SqlDataAdapter("SELECT * FROM ShoeTotalCostSummary", con)

    Public Shared dtTotals As New DataTable
```

```
#Region "Init Setting up the Webpage"

    Private Sub Homework_FinalProject_Init(sender As Object, e As EventArgs) Handles Me.Init



        daShoeInv.FillSchema(dtShoeInv, SchemaType.Source)

        daTotalInvCost.FillSchema(dtTotalInvCost, SchemaType.Source)

        daTotalShoeBrand.FillSchema(dtTotalShoeBrand, SchemaType.Source)

        daTotalShoeType.FillSchema(dtTotalShoeType, SchemaType.Source)

        daTotals.FillSchema(dtTotals, SchemaType.Source)



        If dtShoeInv.Rows.Count > 0 Then

            dtShoeInv.Rows.Clear()

        End If



        If dtTotalInvCost.Rows.Count > 0 Then

            dtTotalInvCost.Rows.Clear()

        End If



        If dtTotalShoeBrand.Rows.Count > 0 Then
            dtTotalShoeBrand.Rows.Clear()
        End If

        If dtTotalShoeType.Rows.Count > 0 Then
            dtTotalShoeType.Rows.Clear()
        End If

        If dtTotals.Rows.Count > 0 Then
            dtTotals.Rows.Clear()
        End If

        Call UpdateDDLForShoe()

        Try

            daTotalInvCost.Fill(dtTotalInvCost)
            daShoeInv.Fill(dtShoeInv)
            daTotalShoeBrand.Fill(dtTotalShoeBrand)
            daTotalShoeType.Fill(dtTotalShoeType)
```

```
daTotals.Fill(dtTotals)



With gvShoeCollection

    .DataSource = dtShoeInv

    .DataBind()

End With




'Setting up the summary page gridviews

With gvSummaryCost

    .DataSource = dtTotalInvCost

    .DataBind()

End With


With gvSummaryBrand

    .DataSource = dtTotalShoeBrand

    .DataBind()

End With
```

```
            With gvSummaryType
                .DataSource = dtTotalShoeType
                .DataBind()
            End With

            With gvSummaryTotals
                .DataSource = dtTotals
                .DataBind()
            End With

        Catch ex As Exception
            Response.Write(ex.Message)
        End Try

    End Sub
#End Region

#Region "Update DDL for Shoes when Adding a Shoe"
```

```
'This sub is meant to update the DDL when the user adds a shoe to the database

Protected Sub UpdateDDLForShoe()



    If dtShoeInv.Rows.Count > 0 Then

        dtShoeInv.Rows.Clear()

    End If



    Try



        daShoeInv.Fill(dtShoeInv)





        With ddlEditShoe

            .DataSource = dtShoeInv

            .DataTextField = "ShoeID"

            .DataValueField = "ShoeID"

            .DataBind()

            .Items.Insert(0, "Select a Shoe")

        End With
        Catch ex As Exception
            Response.Write(ex.Message)
        End Try

    End Sub
#End Region

#Region "Login Feature"
    Protected Sub btnLogin_Click(sender As Object, e As EventArgs) Handles btnLogin.Click

        'When you want to pull 1 number (called a scalar) from a database table, use a SQL command
        Dim intNumberUsers As Integer
        Dim cmdPassword As New SqlCommand("SELECT Count(*) FROM [MF31jimmy.dao].[dbo].[RegisteredUsers] WHERE [UserID] = @UserID AND [Password] = @Password", con)

        With cmdPassword.Parameters
            .Clear()
    .AddWithValue("@UserID", txtUsername.Text)

    .AddWithValue("@Password", txtPassword.Text)

End With



Try

    If con.State = ConnectionState.Closed Then con.Open()

    intNumberUsers = cmdPassword.ExecuteScalar 'ExecuteScalar Returns 1 number



    'Enable the button on the form if the login is successful

    If intNumberUsers = 1 Then

        lbAddRow.Visible = True

        lbSummary.Visible = True

        lbInfo.Visible = True

    End If



    If intNumberUsers = 0 Then

        Response.Write("Invalid Login information")

        lbAddRow.Visible = False

        lbSummary.Visible = False
```

```
                lbInfo.Visible = False
            End If

            con.Close()
        Catch ex As Exception
            Response.Write(ex.Message)
        End Try

    End Sub
#End Region

#Region "Link Buttons that Change webpages"
    Protected Sub LinkButton1_Click(sender As Object, e As EventArgs) Handles lbInfo.Click
        MultiView1.ActiveViewIndex = 0
        btnLogin.Visible = False
        pUsername.Visible = False
        pPassword.Visible = False
        txtPassword.Visible = False
        txtUsername.Visible = False
```

```
End Sub


Protected Sub lbAddRow_Click(sender As Object, e As EventArgs) Handles lbAddRow.Click
    MultiView1.ActiveViewIndex = 1
    btnLogin.Visible = False
    pUsername.Visible = False
    pPassword.Visible = False
    txtPassword.Visible = False
    txtUsername.Visible = False
    ddlEditShoe.Visible = False
End Sub


Protected Sub lbSummary_Click(sender As Object, e As EventArgs) Handles lbSummary.Click
    MultiView1.ActiveViewIndex = 2
    btnLogin.Visible = False
    pUsername.Visible = False
    pPassword.Visible = False
    txtPassword.Visible = False
    txtUsername.Visible = False
    End Sub
#End Region


#Region "Error Checking but also Button that confirms entry"
    Protected Sub btnConfirmEntry_Click(sender As Object, e As EventArgs) Handles
btnConfirmEntry.Click
        'Error Checking
        If IsNumeric(txtShoeSize.Text) = False OrElse
            IsNumeric(txtShoeCost.Text) = False OrElse txtShoeName.Text = Nothing OrElse
            ddlShoeBrand.SelectedIndex = -1 OrElse
            rblShoeCondition.SelectedIndex = -1 OrElse rblShoeType.SelectedIndex = -1 OrElse
txtDateAcq.Text = Nothing OrElse
            txtReleaseDate.Text = Nothing Then

            Response.Write("Please Leave no Boxes un-filled")
            Exit Sub

        Else
```

```
            'If there are no errors then the btn to add a row and an image will appear

            btnAddRow.Visible = True

            imgConfirm.Visible = True


        End If


        'Outputting the user selection to a textbox for the user to view

        txtOutput.Text = "This is the details of the shoe you'll be adding: " & vbNewLine & vbNewLine

        txtOutput.Text &= "Shoe Name: " & txtShoeName.Text & vbNewLine & vbNewLine

        txtOutput.Text &= "Shoe Brand: " & ddlShoeBrand.SelectedItem.Text & vbNewLine & vbNewLine

        txtOutput.Text &= "Shoe Type: " & rblShoeType.SelectedItem.Text & vbNewLine & vbNewLine

        txtOutput.Text &= "Shoe Condition: " & rblShoeCondition.SelectedItem.Text & vbNewLine & vbNewLine

        txtOutput.Text &= "Shoe Size: " & CDec(txtShoeSize.Text) & vbNewLine & vbNewLine

        txtOutput.Text &= "Shoe Cost: $" & CDec(txtShoeCost.Text) & vbNewLine & vbNewLine

        txtOutput.Text &= "Release Date: " & CDate(txtReleaseDate.Text) & vbNewLine & vbNewLine
        txtOutput.Text &= "Acquired Date: " & CDate(txtDateAcq.Text) & vbNewLine & vbNewLine



    End Sub

#End Region


#Region "Button that Adds a Row"

    Protected Sub btnAddRow_Click(sender As Object, e As EventArgs) Handles btnAddRow.Click


        'Declaring variable for a new row

        Dim dr As DataRow = dtShoeInv.NewRow


        'Assigning column values to user input from webpage controls

        dr("ShoeName") = txtShoeName.Text

        dr("ShoeBrand") = ddlShoeBrand.SelectedItem.Text

        dr("ShoeType") = rblShoeType.SelectedItem.Text

        dr("ShoeCondition") = rblShoeCondition.SelectedItem.Text
        dr("Size") = CDec(txtShoeSize.Text)

        dr("ShoeCost") = CDec(txtShoeCost.Text)

        dr("ReleaseDate") = txtReleaseDate.Text

        dr("AcquiredDate") = txtDateAcq.Text


        dtShoeInv.Rows.Add(dr)


        Try 'The .Fill function/method saves the data to the database, and .Update (this is when the commmandbuilder wakes up and goes to work)

            'Data adapters do three things,

            daShoeInv.Update(dtShoeInv)


            Call UpdateSummaryData()


            'Clear the row of data from the array and then retrieve all the rows in the shopping list

            dtShoeInv.Rows.Clear()


            'Fetch all the rows the fill functions saves the data to the database

            'At this next line of code the red SQL SELECT statement is executed
```

```
            daShoeInv.Fill(dtShoeInv)

            gvShoeCollection.DataSource = dtShoeInv
            gvShoeCollection.DataBind()

        Catch ex As Exception
            Response.Write(ex.Message)
        End Try

        Call UpdateDDLForShoe()

    End Sub
#End Region

#Region "Clear Shoe Option Selections"
    'Button that will clear the user inputs to get ready to add a new row or to simply clear
    Protected Sub btnClearShoe_Click(sender As Object, e As EventArgs) Handles
btnClearShoe.Click
```

```
        txtShoeName.Text = Nothing

        txtDateAcq.Text = Nothing

        txtOutput.Text = Nothing

        txtReleaseDate.Text = Nothing

        txtShoeCost.Text = Nothing

        txtShoeSize.Text = Nothing

        rblShoeCondition.SelectedIndex = -1

        rblShoeType.SelectedIndex = -1

        ddlShoeBrand.SelectedIndex = -1

        imgBrand.Visible = False


    End Sub

#End Region



#Region "Edit, Delete or Add a Shoe View"

    Protected Sub RadioButtonList1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles rblEditShoe.SelectedIndexChanged

        'Add a shoe view

        If rblEditShoe.SelectedIndex = 0 Then
```

```
ddlEditShoe.Visible = False

btnUpdateRow.Visible = False

btnDelete.Visible = False

pBrand.Visible = True

pCondition.Visible = True

pCostOfShoe.Visible = True

pDateAcq.Visible = True

pReleaseDt.Visible = True

pShoeName.Visible = True

pShoeType.Visible = True

pSize.Visible = True

txtDateAcq.Visible = True

txtOutput.Visible = True

txtReleaseDate.Visible = True

txtShoeCost.Visible = True

txtShoeName.Visible = True

txtShoeSize.Visible = True

ddlShoeBrand.Visible = True

rblShoeCondition.Visible = True
```

```
    rblShoeType.Visible = True

    btnConfirmEntry.Visible = True

    btnClearShoe.Visible = True

    gvShoeCollection.Visible = True


End If


'Edit a shoe view

If rblEditShoe.SelectedIndex = 1 Then

    ddlEditShoe.Visible = True

    btnUpdateRow.Visible = True

    btnDelete.Visible = False

End If


'Delet a shoe view

If rblEditShoe.SelectedIndex = 2 Then

    ddlEditShoe.Visible = True

    btnUpdateRow.Visible = False

    btnDelete.Visible = True
```

```
        End If


    End Sub

#End Region


#Region "DDL Retrieving One Row"

    Protected Sub ddlEditShoe_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlEditShoe.SelectedIndexChanged


        'Retrieve the requested row of data for editing/updating

        'Clear out the last row of data that was updated

        If dt1Shoe.Rows.Count > 0 Then

            dt1Shoe.Rows.Clear()

        End If


        'Here we need to connect the SQL SELECT statement to the webpage, providing the number of the row to edit. Here we pull the value from the dropdownlist


        With da1Shoe.SelectCommand.Parameters
    .Clear()
    .AddWithValue("@p1", ddlEditShoe.SelectedValue)
    'Pass the .selectedvalue from the ddl via the parameter into the SQL SELECT statement
End With

Try 'Fetch just one row
    da1Shoe.Fill(dt1Shoe)

    With dt1Shoe.Rows(0)
        txtShoeName.Text = .Item("ShoeName")
        ddlShoeBrand.SelectedValue = .Item("ShoeBrand")
        rblShoeType.SelectedValue = .Item("ShoeType")
        rblShoeCondition.SelectedValue = .Item("ShoeCondition")
        txtShoeSize.Text = .Item("Size")
        txtShoeCost.Text = .Item("ShoeCost")
        txtReleaseDate.Text = .Item("ReleaseDate")
        txtDateAcq.Text = .Item("AcquiredDate")
    End With
```

```
        Catch ex As Exception
            Response.Write(ex.Message)
        End Try

    End Sub
#End Region

#Region "Button that Updates a Row"
    Protected Sub btnUpdateRow_Click(sender As Object, e As EventArgs) Handles btnUpdateRow.Click
        'While we use the parameterized SQL SELECT statement to retrieve one row of data for updating, we must use the non-parameterized data Adapter

        With dt1Shoe.Rows(0)
            'Here we are copying the newly updated values from the webpage into the row of data
            .Item("ShoeName") = txtShoeName.Text
            .Item("ShoeBrand") = ddlShoeBrand.SelectedItem.Text
            .Item("ShoeType") = rblShoeType.SelectedItem.Text
            .Item("ShoeCondition") = rblShoeCondition.SelectedItem.Text
            .Item("Size") = CDec(txtShoeSize.Text)
            .Item("ShoeCost") = CDec(txtShoeCost.Text)
            .Item("ReleaseDate") = txtReleaseDate.Text
            .Item("AcquiredDate") = txtDateAcq.Text

        End With


        Try
            'Save the updated row of data (Inside the datatable) back to the SQL database table
            daShoeInv.Update(dt1Shoe) '.Fill runs a SQL Select statement (Retrieve), .update (saves the data back)

            Call UpdateSummaryData()


            'Now fetch all the rows of data (Clear out the old data 1st)
            If (dtShoeInv.Rows.Count) > 0 Then
                dtShoeInv.Rows.Clear()
            End If
            daShoeInv.Fill(dtShoeInv)

            With gvShoeCollection
                .DataSource = dtShoeInv
                .DataBind()
            End With

        Catch ex As Exception
            Response.Write(ex.Message)
        End Try

        Call UpdateDDLForShoe()

    End Sub
#End Region

#Region "Button that Deletes a Row"
    Protected Sub btnDelete_Click(sender As Object, e As EventArgs) Handles btnDelete.Click
```

```
'We use a SQLCommand to execute a SQL DELETE


Dim cmdDeleteRow As New SqlCommand("DELETE FROM ShoeInventory WHERE ShoeID = @p1", con)

With cmdDeleteRow.Parameters

    .Clear()

    .AddWithValue("@p1", ddlEditShoe.SelectedValue)

End With




Try

    'Delete row of data

    'We have to open and close the connection

    If con.State = ConnectionState.Closed Then con.Open()

    'Now execute the delete SQL statement


    cmdDeleteRow.ExecuteNonQuery()

    con.Close()
    'Clear out full list of data displayed (clear the datatable)
    If (dtShoeInv.Rows.Count) > 0 Then
        dtShoeInv.Rows.Clear()
    End If
    'Refill the datatable with the new full list
    daShoeInv.Fill(dtShoeInv)


    'Display in the gridview
    With gvShoeCollection
        .DataSource = dtShoeInv
        .DataBind()
    End With

Catch ex As Exception
    Response.Write(ex.Message)
End Try

Call UpdateDDLForShoe()
```

```
    End Sub
#End Region

#Region "Sub that Updates Summary Data"
    'This will update the summary data on the Summary Analytics Page
    Protected Sub UpdateSummaryData()


        Dim cmdUpdateTotalInvCost As New SqlCommand("UPDATE ShoeTotalSummary SET
[TotalShoeCost] += @p1, [TotalShoes] += 1 WHERE [ShoeBrand] = @p2", con)

        Dim cmdUpdateTotalBrand As New SqlCommand("UPDATE ShoeBrandSummary SET
[TotalShoeBrand] += 1 WHERE [ShoeBrand] = @p1", con)

        Dim cmdUpdateTotalType As New SqlCommand("UPDATE ShoeTypeSummary SET
[QuantityofTypeShoe] += 1 WHERE [TypeOfShoe] = @p1", con)

        Dim cmdUpdateTotals As New SqlCommand("UPDATE ShoeTotalCostSummary SET
[TotalCost] += @p1, [TotalShoes] += 1", con)
```

```
'Adding the Running Totals
With cmdUpdateTotalInvCost.Parameters
    .Clear()
    .AddWithValue("@p1", CDec(txtShoeCost.Text))
    .AddWithValue("@p2", ddlShoeBrand.SelectedValue)
End With

With cmdUpdateTotalBrand.Parameters
    .Clear()
    .AddWithValue("@p1", ddlShoeBrand.SelectedValue)
End With

With cmdUpdateTotalType.Parameters
    .Clear()
    .AddWithValue("@p1", rblShoeType.SelectedValue)
End With

With cmdUpdateTotals.Parameters
```

```
    .Clear()
    .AddWithValue("@p1", CDec(txtShoeCost.Text))
End With

Try
    If con.State = ConnectionState.Closed Then con.Open()

    cmdUpdateTotalInvCost.ExecuteNonQuery()
    cmdUpdateTotalBrand.ExecuteNonQuery()
    cmdUpdateTotalType.ExecuteNonQuery()
    cmdUpdateTotals.ExecuteNonQuery()
    con.Close()

    Call RefreshGridViews()

Catch ex As Exception

End Try
```

```
    End Sub

#End Region



#Region "Refreshing the Summary GridViews"

    Protected Sub RefreshGridViews()



        'Clearing the rows

        If dtTotalInvCost.Rows.Count > 0 Then dtTotalInvCost.Rows.Clear()

        If dtTotalShoeBrand.Rows.Count > 0 Then dtTotalShoeBrand.Rows.Clear()

        If dtTotalShoeType.Rows.Count > 0 Then dtTotalShoeType.Rows.Clear()

        If dtTotals.Rows.Count > 0 Then dtTotals.Rows.Clear()




        'Filling the tables back up for summary data

        daTotalInvCost.Fill(dtTotalInvCost)

        gvSummaryCost.DataSource = dtTotalInvCost

        gvSummaryCost.DataBind()



        daTotalShoeBrand.Fill(dtTotalShoeBrand)
        gvSummaryBrand.DataSource = dtTotalShoeBrand

        gvSummaryBrand.DataBind()



        daTotalShoeType.Fill(dtTotalShoeType)

        gvSummaryType.DataSource = dtTotalShoeType

        gvSummaryType.DataBind()



        daTotals.Fill(dtTotals)

        gvSummaryTotals.DataSource = dtTotals

        gvSummaryTotals.DataBind()



    End Sub



#End Region



#Region "DDL to have shoe brand image pop up"

    'Simply here the user will make a selection of a shoe brand and the image will pop up depending on the brand
    Protected Sub ddlShoeBrand_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlShoeBrand.SelectedIndexChanged

        Select Case ddlShoeBrand.SelectedIndex
            Case Is = 1
                imgBrand.Visible = True
                imgBrand.ImageUrl = "nikelogo.png"
            Case Is = 2
                imgBrand.Visible = True
                imgBrand.ImageUrl = "adidas.png"
            Case Is = 3
                imgBrand.Visible = True
                imgBrand.ImageUrl = "jordan.png"
            Case Is = 4
                imgBrand.Visible = True
                imgBrand.ImageUrl = "puma.png"
            Case Is = 5
                imgBrand.Visible = True
                imgBrand.ImageUrl = "vans.png"
            Case Is = 6
                imgBrand.Visible = True
                imgBrand.ImageUrl = "others.png"
        End Select

    End Sub
#End Region


End Class
```