**Conestoga College - ACS & IT**

**Programming Microsoft Web Technologies – PROG2230**

**Assignment #2: Introductory Data-driven ASP.NET Core MVC Web Apps**

**Due: October 15, 2021 (by Midnight)**

**Total marks: 40 Worth: 6%**

**Introduction**

The goal of this assignment is to use the concepts you learned for migrations and extend them further. You will also be introduced to Query Strings (routing segments) as well as TempData.

**First, a word of caution**

I remind you to make sure you do your own work on this assignment and resist any urge to copy code from any other source - e.g. your classmates, the web, etc. Not only is this the only way to learn how to program but also everyone’s solution will be run through [Moss](https://theory.stanford.edu/~aiken/moss/) to check for academic integrity violations. There is zero-tolerance for such violations and any encountered with be dealt with in accordance with [Conestoga’s policy](https://lib.conestogac.on.ca/academic-integrity/penalties). I also remind you that if you are not typing syntactically correct code in yourself you are not learning to program! Finally, I’ll note that these assignments are very representative of what will be expected of you on exams so it is very much in your interest to ensure that you are capable of doing them on your own.

**Hints**

The general idea is that there will be time to work on these questions in class and, if necessary, I can offer hints if I see that you are struggling with certain parts. Also, you should try to pace out your work on this assignment over the coming weeks.

**How will it be graded?**

I’ll say from the outset that there is absolutely no reason to not get 100% - the emphasis here is simple: get your hands dirty coding to solve basic problems and, if needed, I will do what I can to help steer you towards a working solution.

Accompanying each assignment will be an Excel marking sheet that details how your grade will be calculated so you obviously want to make sure that you are doing everything as it’s laid out there.

**What/how to submit?**

Zip up your entire solution into ***one zip file*** and submit that file to the eConestoga dropbox for the assignment. You can submit multiple solutions but only the latest (i.e. most recent) one will be looked at and evaluated.

**What will you build?**

Your client, an investor, wants to UPGRADE the app you created in AS1 to keep track of the stock transactions that they make in their personal investment portfolio. There are several upgrades the investor has asked for you to accommodate:

For this assignment, you need to MODIFY your AS1 ASP.NET Core MVC (or use the solution) web app:

**Generic**

* ~~Modify the header of the app to include the investor’s company logo. You may create a logo and company name for the investor, or download one from the internet. It must appear on ALL pages and the logo must be included in the project. Clicking on the logo will take you to the default page.~~
* ~~Also created a nav bar next to the logo with three (or four) options; “Home” and “Add Transaction”, “Transactions” and “Companies”; which go to the appropriate pages. Home and Transactions pages may be the same page.~~
* ~~Modify ALL of the dropdowns in all of the Create/Update pages you created, instead of the keys behind display, display the names. ie; “Sell”, “Buy”. You may have already done this in your AS1 if you manually created your controllers / pages. If you scaffold (will cover this in class), you must do this.~~
* ~~Create a TempData variable named “message” on your Shared Layout BEFORE the @Renderpage div. Modify all of your CRUD (Create/Read/Update/Delete) operations to show a message when a save is successful (or not successful) when being processed by the server. Make sure you add Colour / Class to show Success.~~

**Transaction**

* ~~Modify your TransactionRecord Controller to allow sorting by Company Name via a link in the table header for the listings.~~
* ~~Use a QueryString (static and dynamic segments within your routing configuration) to send the sort parameters to the controller combined with OrderBy or OrderByDescending.~~
* ~~When first opened, the table is sorted by Company Name alphabetically in ASC order. When clicked, it is sorted by DESC. When clicked again, it is back to ASC.~~
* ~~You \*MUST\* Use a QueryString, have the controller do the sort, and send the data back. Do NOT use a Javascript sort.~~
* ~~HINT: A Viewbag could be used to adjust the sorting based on previous sorts~~

**Company Model / Controller**

* ~~Create a new “Company” Model that splits the company data and company stock data and migrate/update your database. When complete, you will have 2 models (in regards to this):~~

|  |  |
| --- | --- |
| ~~Company Model~~   * ~~CompanyID~~ * ~~Name~~ * ~~Address~~ * ~~TickerSymbol~~ | ~~Transaction Model~~   * ~~TransactionModelID~~ * ~~TransactionTypeID~~ * ~~SharePrice~~ * ~~Qty~~ * ~~CompanyID~~ |

* ~~Create a new CompanyController, along with a set of CRUD (Create Read Update Delete) pages for the Company model.~~
* ~~Update your Transaction Edit / Create to incorporate the new Company model.~~
* ~~When a record is added, updated or deleted, the application should display the full list of Companies with the appropriate message~~.
* ~~(eg. “CompanyName has been deleted.” or “CompanyName has been added”)~~
* ~~Modify your Seed Data appropriately. You may even want to erase your database and re-run the Update-Database command after you do this migration. That will save you from updating the old data manually, which isn’t required.~~

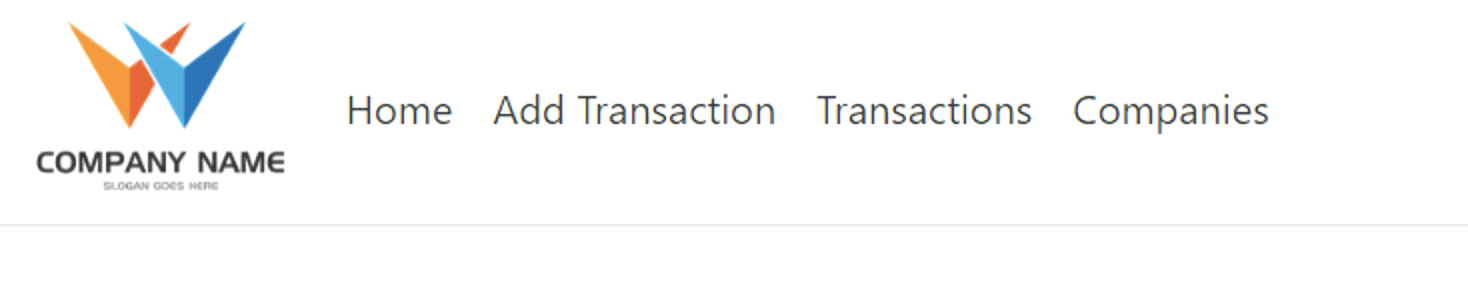
**Transaction + Company**

* ~~Add a link in your ACTION column on your new Company Listing page (Index), that links to the Transactions page. The page should ONLY show transactions for the specific company the user is clicking on. You will have to modify your TransactionController, manually create an Action (aka, function) and a view - that limits the listings to the company only.~~
* ~~At the top of this new page page, the Company Name and Ticker Symbol should be listed in <h2> tags, showing the user that they are currently looking at only Transactions for the company they clicked on. (Multiple ways of doing this. Think how you pass a SelectList)~~
* ~~In your TransactionRecord listing, change the Company Name into a link that goes to the edit page of that particular company. This would be the opposite of the above~~

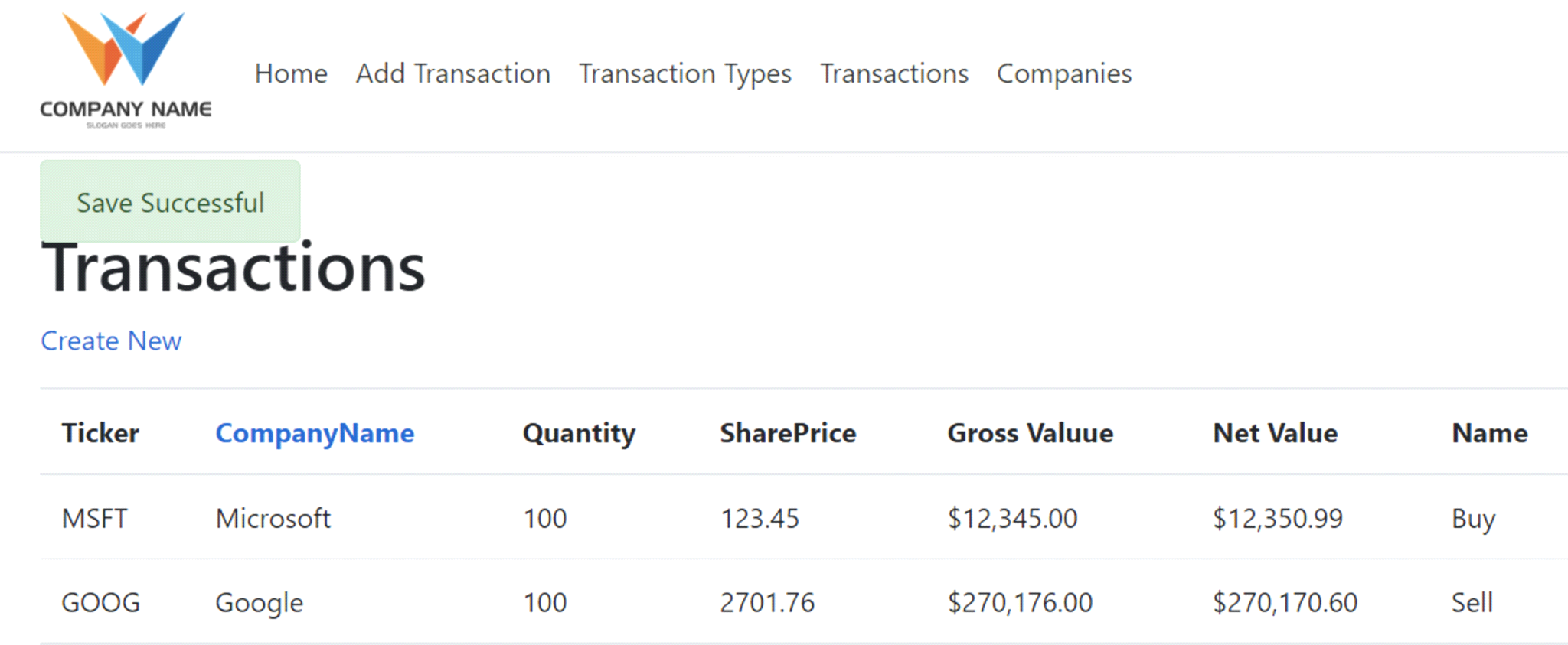
**~~HINT~~**~~: You should probably use a QueryString to send the ID of the company to the Transaction List, get the data from the database and send it back out to the same view you already created.~~

**Screenshot Examples**

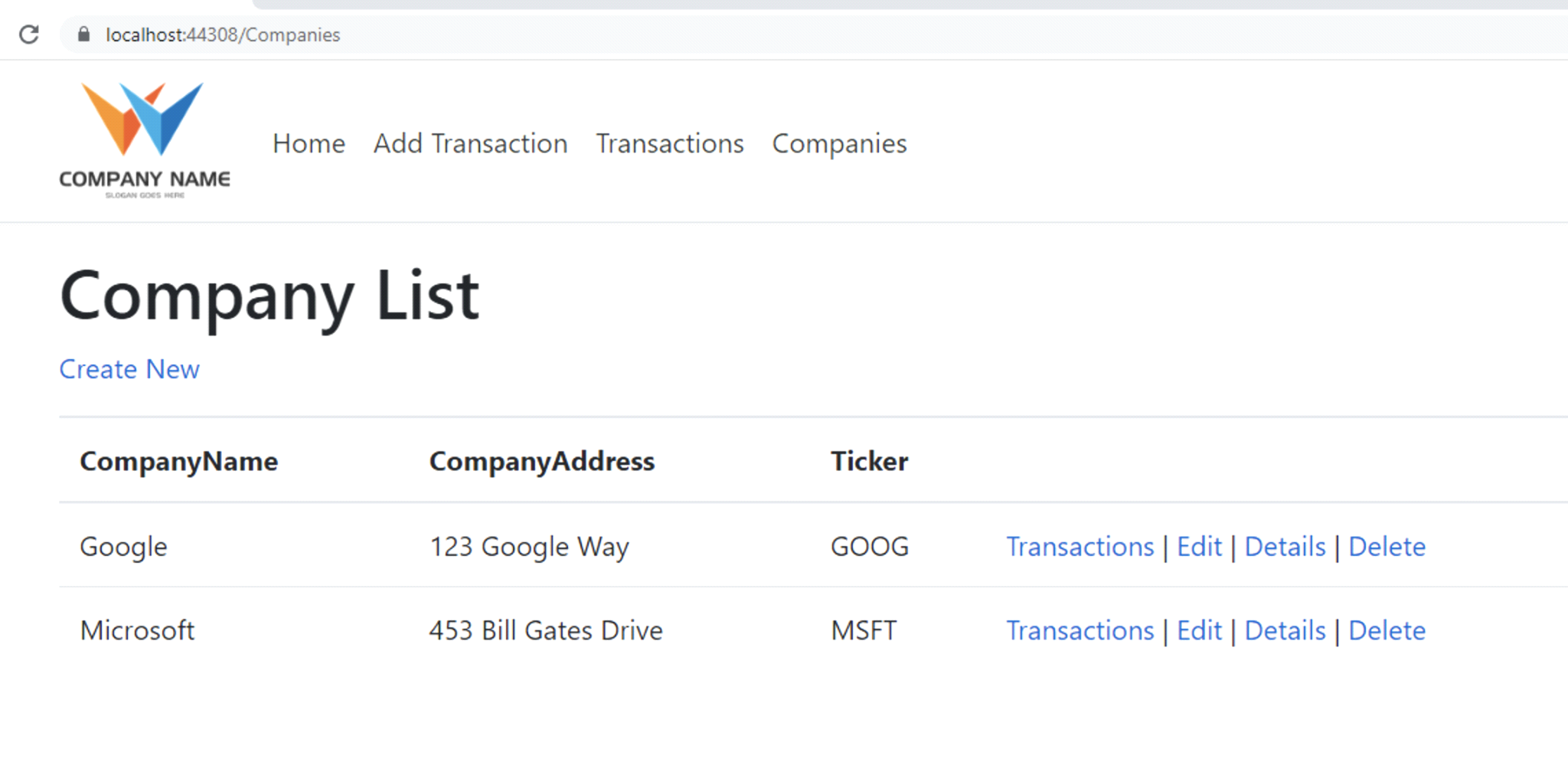
Navigation Bar

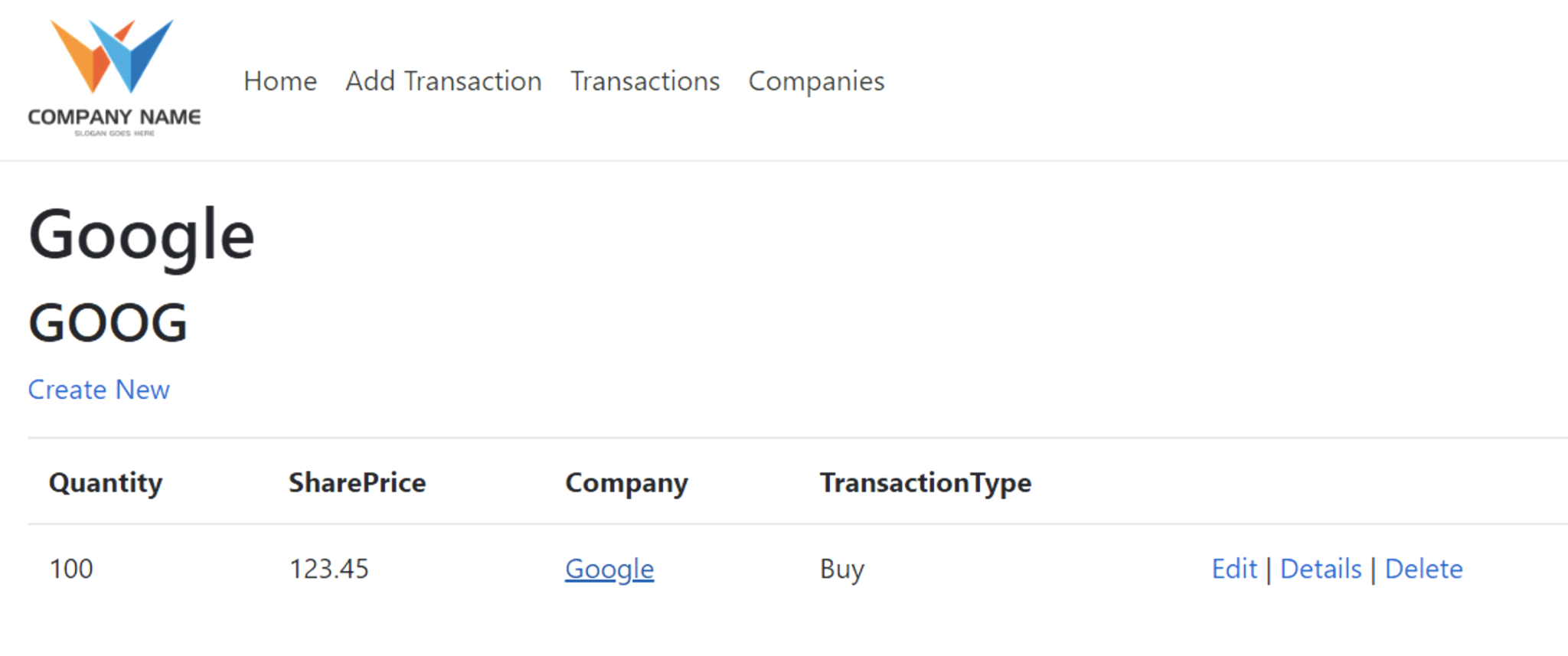


TempData after Save + Sort



Transaction + Company





**Database Diagram**

Your database should look something similar to this now:

