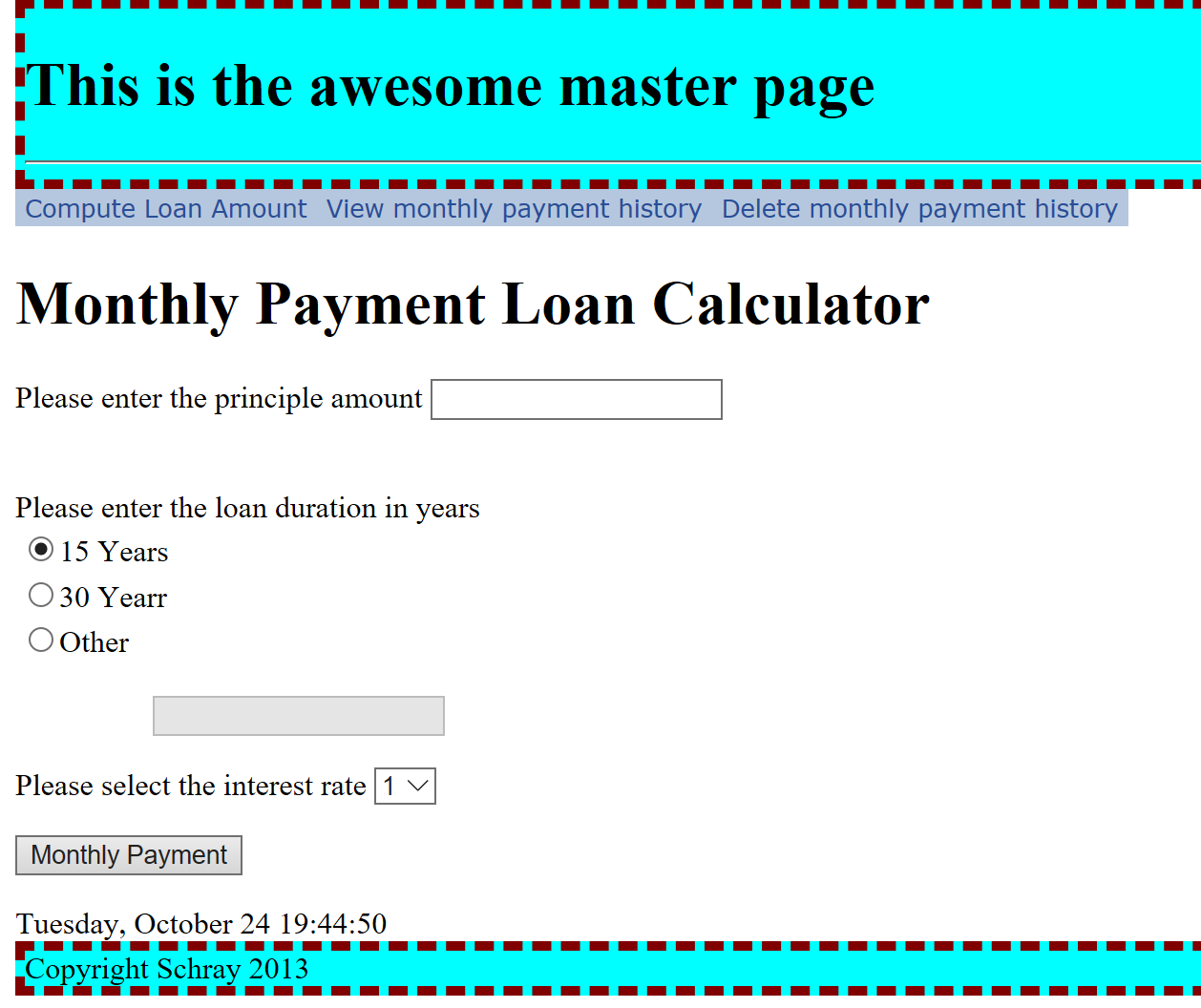
**ITMD 463/563 Project 4**

* **Goal:** Write an ASP.NET web application using Master Pages and Content pages. The app will be a mortgage calculator application that calculates the monthly payment for a loan based on the principal, number of years, and interest rate. You should be able copy, and leverage components from project 3 in this new master pages project.
* **Relevant Example:** See my lecture on MasterPages.
* **Deliverable:** Submit a zip file of your .Net Solution file on BlackBoard using this naming convention: Pro4-Smith.zip (replace Smith by your last name).
* **May the UI well** to make your application clear to the user. Something better than this. Also make the dropdown for rates more complete (offer more choices from .25% to 10% by .25 percent, so .25%, .50%, .75%, … 9.75 and finally 10%).
* Note the use of the ScriptManager to allow AJAX components (updates without a full page postback) for the timer and for radiobuttons and the optional years textbox.
* **Note in the example there are THREE pages: One for the loan info, one for viewing past loan entered, and one to clear the loans file.**
* You’ll be staying with the file-based IO version of the application



* **Detailed Specs:**   
  1. Input the principal, number of years, and interest rate as string variables.
  2. Convert each of the variables in the preceding step to double or decimal using the double.TryParse or decimal.TryParse method. Prompt the user to reenter any illegal input.
  3. Compute the monthly payment using this formula:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| m | = | |  | | --- | | p r / 1200.0 | |  | | 1 - (1.0 + r / 1200.0) -12.0 n | |

p = principal (dollars), n = number of years, r = interest rate (percent), m = monthly payment (dollars).   
   
Use the .Net function call Math.pow(x, y) to compute xy (x raised to the y power).

// use the function to calculate the monthly payment

private double ComputeMonthlyPayment(double principal, double years, double rate)

{

double monthly = 0;

double top = principal \* rate / 1200.00;

double bottom = 1 - Math.Pow(1.0 + rate / 1200.0, -12.0 \* years);

// http://www.bankrate.com/calculators/mortgages/loan-calculator.aspx

monthly = top / bottom;

//Console.WriteLine();

//Console.WriteLine("With a principl of ${0}, duration of {1} years and a interest rate of {2}% the monthly loan payment amount is {3:$0.00}", principal, years, rate, monthly);

return monthly;

}

* 1. Output the amount of the monthly payment.

Hints and questions to ask yourself before handing this in:

* Is the Window title set to something meaningful?
* Things to think about
  + You can use If (!IsPostback) in page\_load to see if you processing a post back.
* Make sure to use a RadioButtonList for your radio button needs
  + On the RadioButtonList make sure to set AutoPostback property is set to true (this way when change a radio value is changed you get a post back. Why? You need to enable the Textbox is they picked other.
  + Use the RadioButtonList selection changed event to respond to changes to the radiobutton lists selection (remember you'll have to enable and disable the Textbox)
  + Rather than a combobox you’ll be using a DropDownList.
* Did I break the functionality out in such a way that I would reuse part this this again. For example, the computation components or the conversion from a string component again easily (hint: like in the next assignment)?
* Did I format the monthly payment as a proper money amount (e.g. $2.91 instead of 2.90813453)?
* Check the output with this site <http://www.bankrate.com/calculators/mortgages/loan-calculator.aspx>
* Please use these naming standards for future projects: <http://blogs.msdn.com/b/mschray/archive/2013/05/06/practical-naming-standards-for-c.aspx>
* Is the output clear?
* **Grading:** Per syllabus