Spring 2015

Kristen Harris COP 4710

Phi Tran

Jasmine Rivera

University of South Florida

Project 4: An Online Enterprise Information System

Index

Vision and Scope …………………………………………………………………………...…… 2

Ring Web Application ………………………………………………………………...… 2

Database Design …………………………………………………………………………………. 3

Entity-Relationship Diagrams……………………………………………………………. 3

Ring Application…………………………………………………………………………………. 4

Interface.................................……………………………………………………………. 4

Application Logic..................……………………………………………………………. 6

Database.................................……………………………………………………………. 6

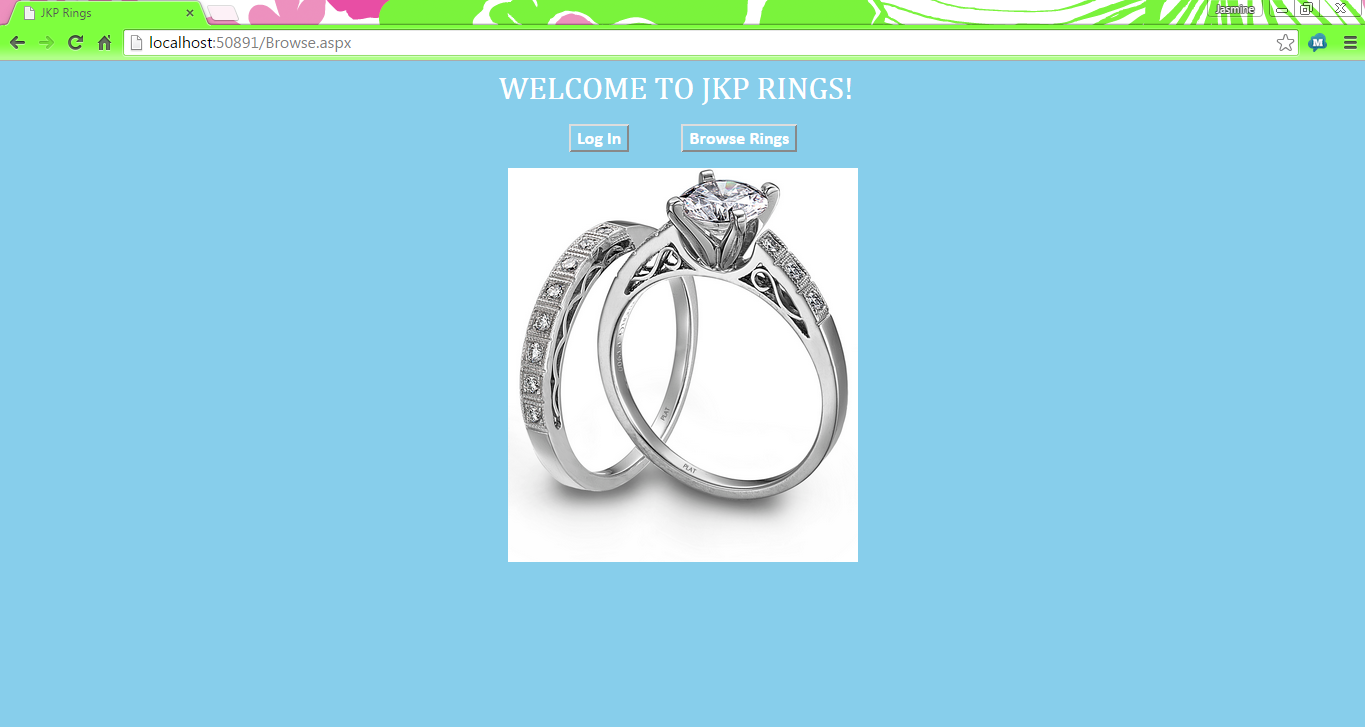
1. Ring Web Application
2. Vision Statement

The ring web application is a website which will allow a customer to design a ring. The customer will be able to personalize rings to their specifications based on gems, metal, karats and size. This website is intended to relieve the stress of ring buying that many go through by allowing you to completely customize a ring to your specifications. The website will provide suggestions based on what the user is particularly interested in. When the customer is satisfied with their ring, they will be able to purchase their beautiful piece of jewelry.

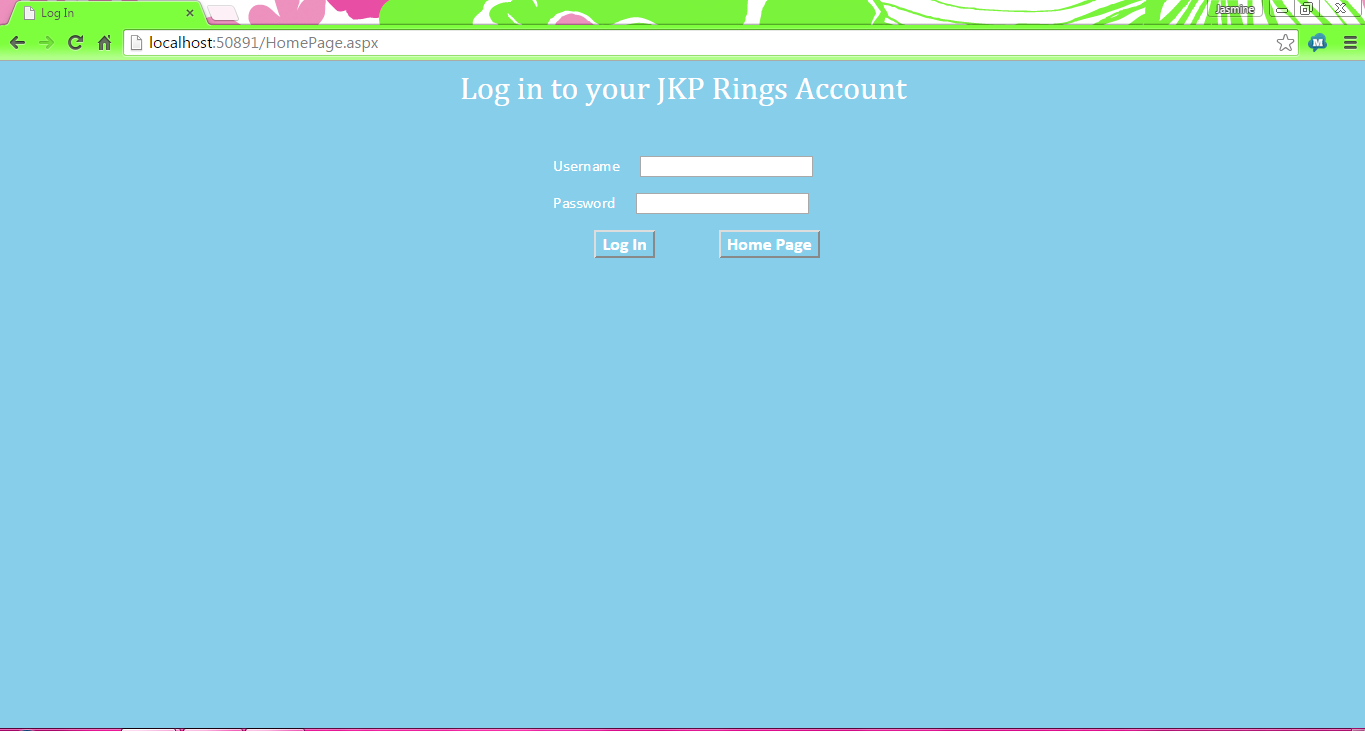
1. Major Features

* Customer can create an account and access and edit their user ID, password, name, address, payment information as well as order IDs
* Customer is able to create a ring based on personal specifications
* Ring is customizable based on gem, metal, size, karats and cost
* Customer is able to purchase their customized ring
* Customer and employee are assigned different privileges
* Employee is able to keep track of sales
* Hashes for the passwords and MAC for the order forms to protect customer

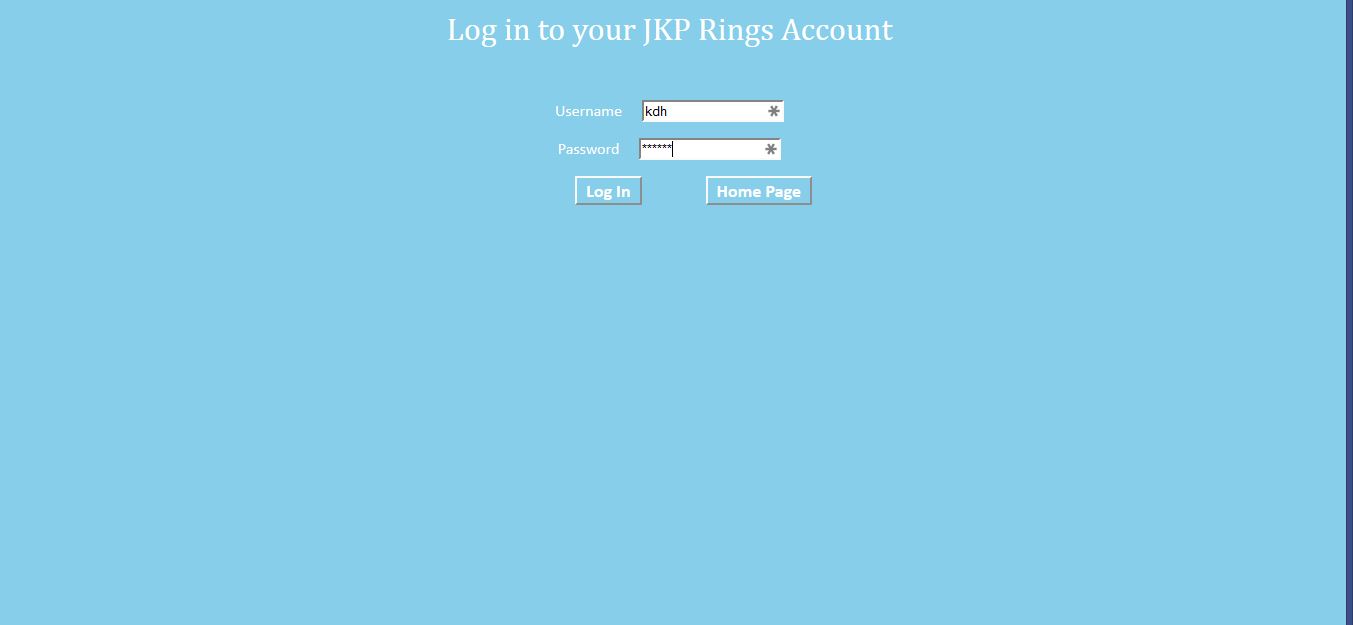
1. Database Design  
   The ER diagram below provides an introduction to our database. They are provided for reference to understanding our system throughout this report.
2. Entity-Relationship Diagrams
3. The Ring Web Application
4. Interface

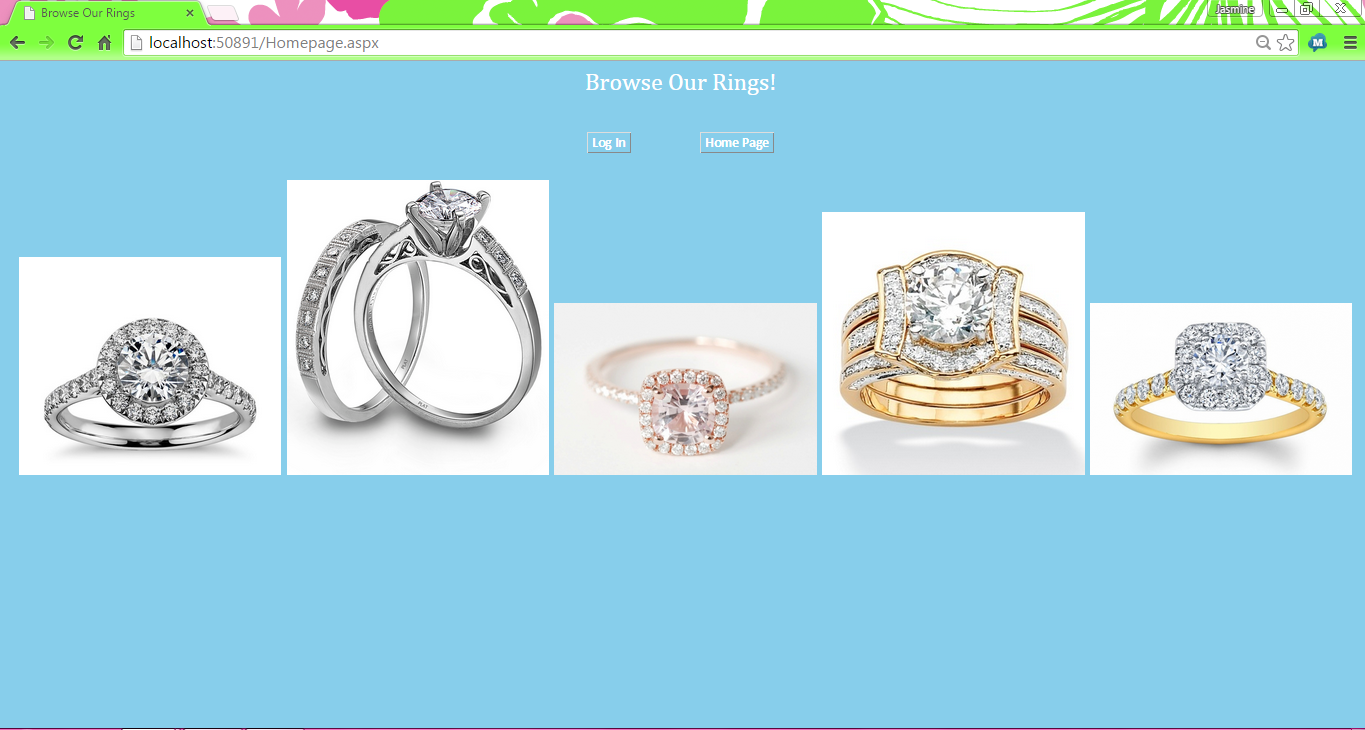
* Home Page

The homepage will be our landing when accessing our site. You will have the option to either log in to your account or browse our rings. When the Log In button is clicked, you will be directed to the Log -In screen (figure below). If the Browser Rings button is clicked, then you will be directed to the Browse page of the website.

* Login Screen

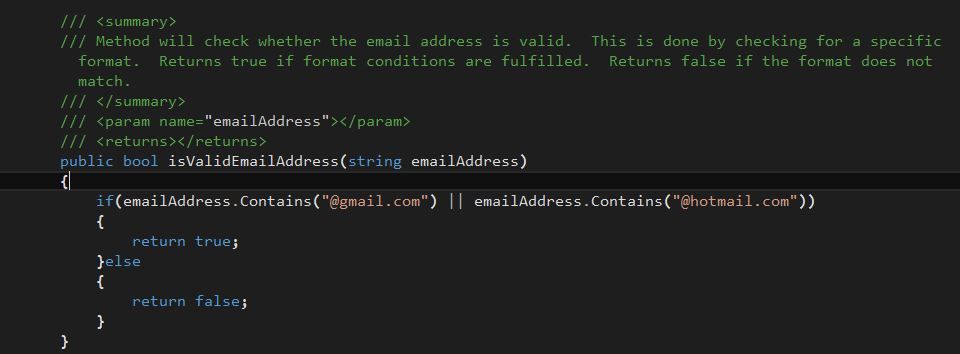
The log in screen will allow you to enter your username and password to log in to your account in which you can access your account information. The image below displays how a user would enter login credentials (username and password). As indicated by the figure below, the password is obscured so no one else would be able to see it in plain text.



* Browse Screen  
  The browse screen allows you to browse through our rings. At this stage you will go through and pick out your criteria to best fit your needs to one of our rings.

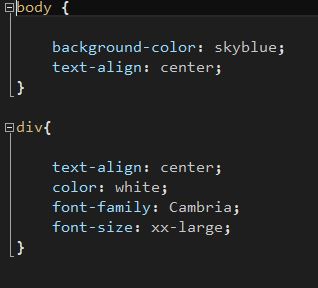
1. Application Logic

* Email Validator



The email validator, written in C#, checks to see if the user's email address has a valid format. The only parameter of the method is a string value that is the user's email address. Once the email address is passed to the method, there is a conditional if statement that checks whether the string contains a valid format. If the format matches, then it returns true. Otherwise, it returns false.

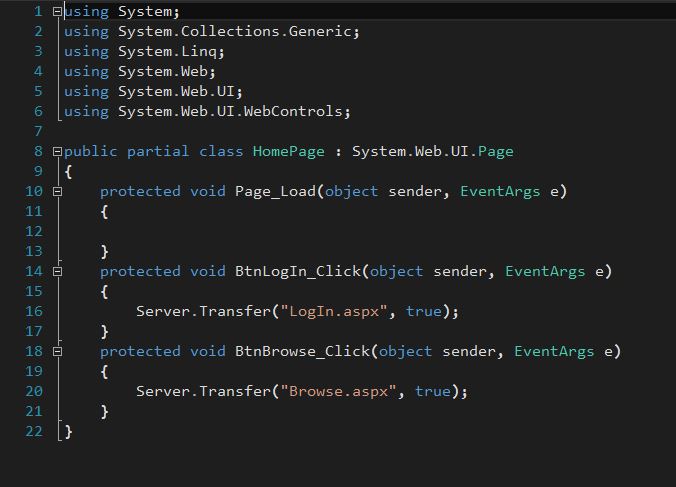
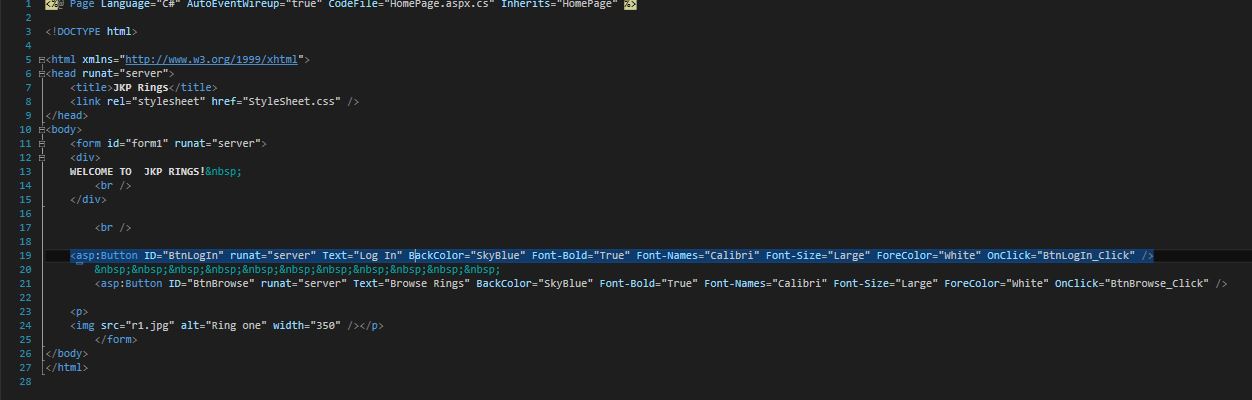
* Cascading Style Sheet (CSS)



The above image is the cascading style sheet used to set the format of each web document. In this case, the css is formatting web documents with the following:

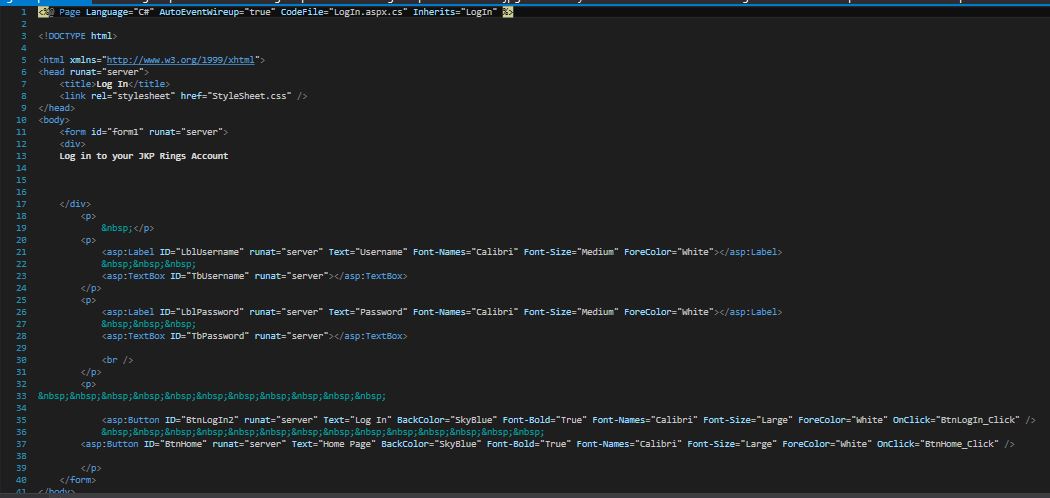
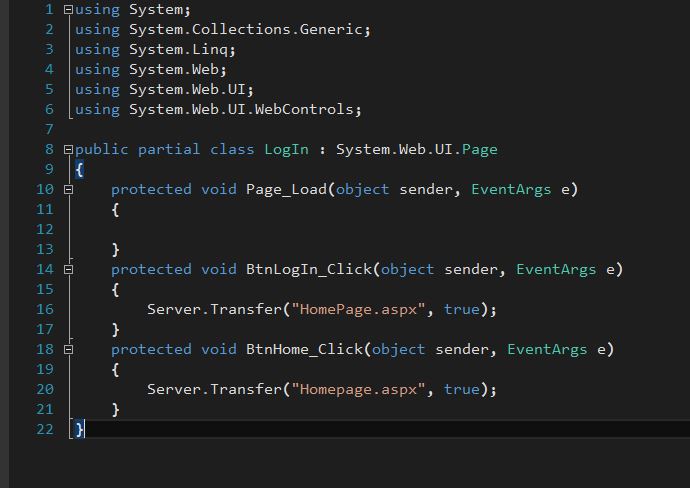
* + Sky blue background color
  + Centered text alignment
  + White text color
  + Cambria font
  + XX-Large font size
* Home Page

The Homepage’s application logic can be seen in the next two images below. The first image is the code for the user interface of the webpage. The second image is the C# code that provides the events for each button on the Home page (Login and Browse).



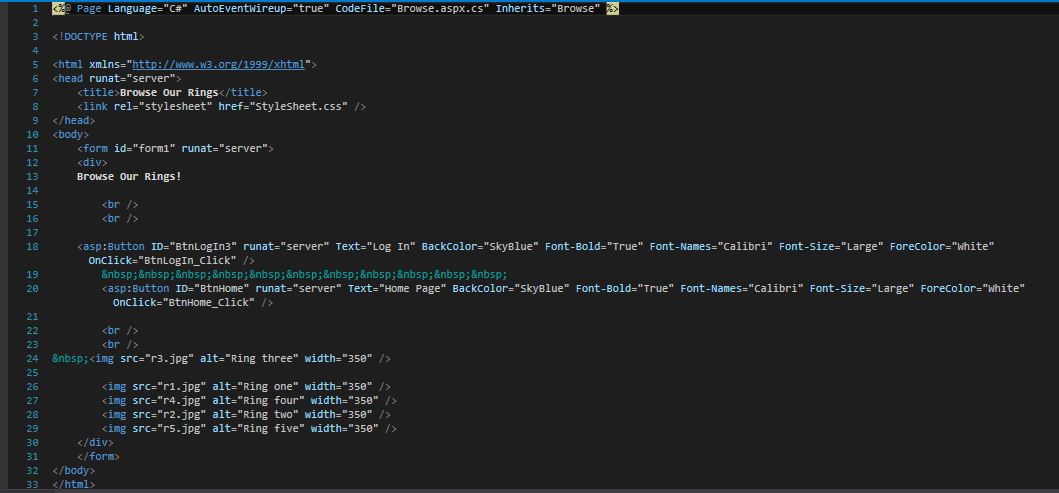
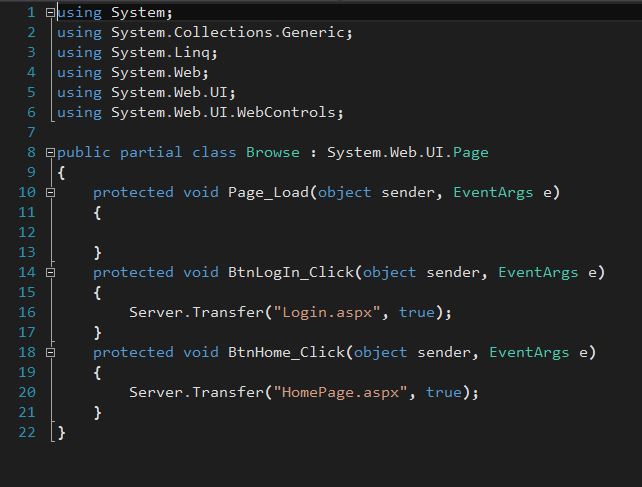
* LogIn

The Login screen’s application logic can be seen in the next two images below. The event driven C# code that sets up the buttons functionality can be seen from the first image. is the code for the user interface of the webpage. The next image is the code that sets up the user interface as well as the formatting, text, and images that are on the Login screen.



* Browse

The Browse screen’s application logic can be seen in the next two images below. The event driven C# code that sets up the buttons (Log In and Homepage) functionality can be seen from the first image. is the code for the user interface of the webpage. The next image is the code that sets up the user interface as well as the formatting, text, and images that are on the Browse screen.



1. Database

* Data Generation

The database we created is populated with four columns that consist of first names, last names, email addresses, and hashed passwords. There is a total of 1000 entries for each attribute. A python script was used to randomly generate each attribute of the data. Afterwards, the database was constructed in Catalyst (see below) where the data is now stored. The second image is a small sample of the 1000 entries of the database.

