Internet Systems Programming: Term Project

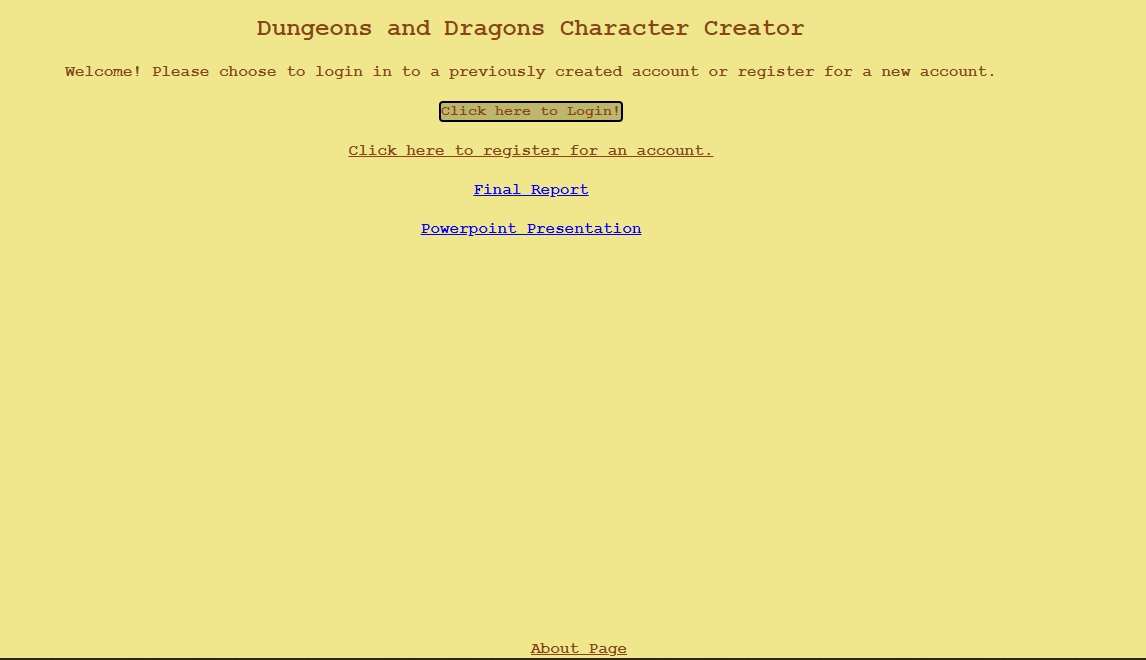
Final Report: Dungeons & Dragons Character Creator

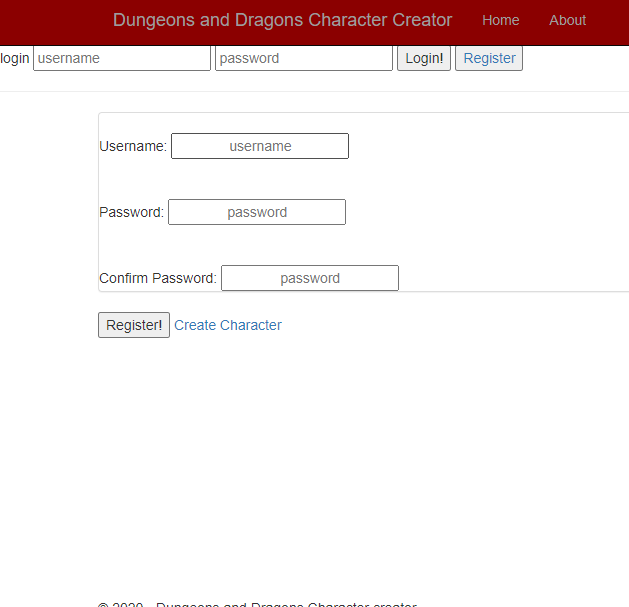
By: Harvey Petersen, Faith Sekerak, Sean Watson

**Implementation:**

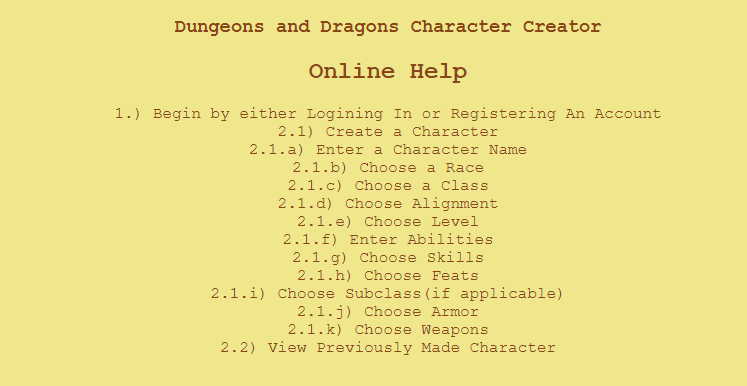
Our app is a series of web pages built using the ASP.Net framework and utilizes a Microsoft Azure SQL Database. The database has 27 different tables that are roughly divided into two categories - data from the rules of Dungeons and Dragons regarding character creation, and tables storing characters that the user has created and which values from the rules tables apply to the character. Our web pages use code behind pages in C# that manage event handlers, interfacing with the SQL database, and event driven business logic. Our login system utilizes cookies to track which user is logged in and access the characters that user has created. The main display of the web pages are made with HTML.

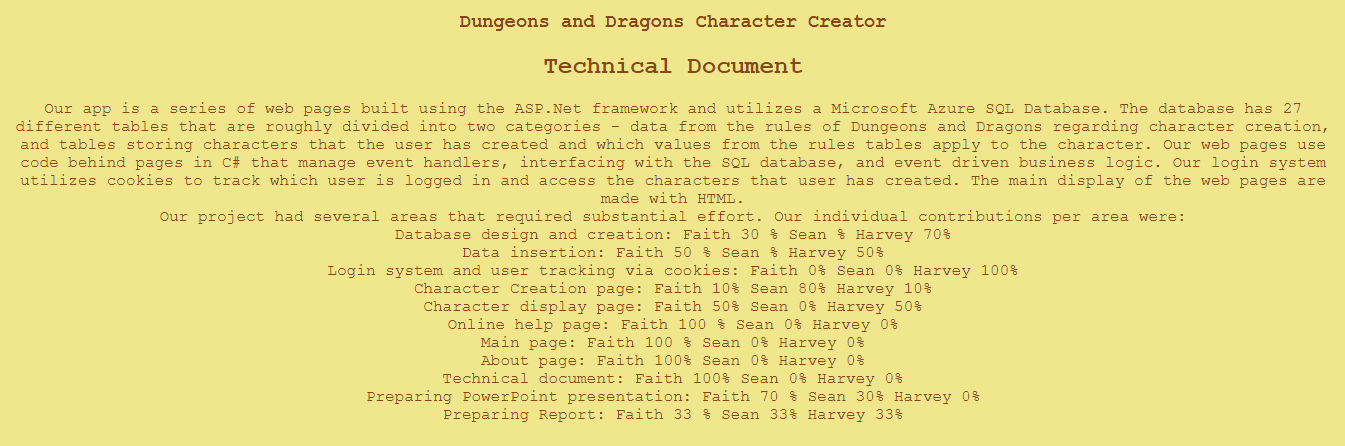
**Screen Shots:**

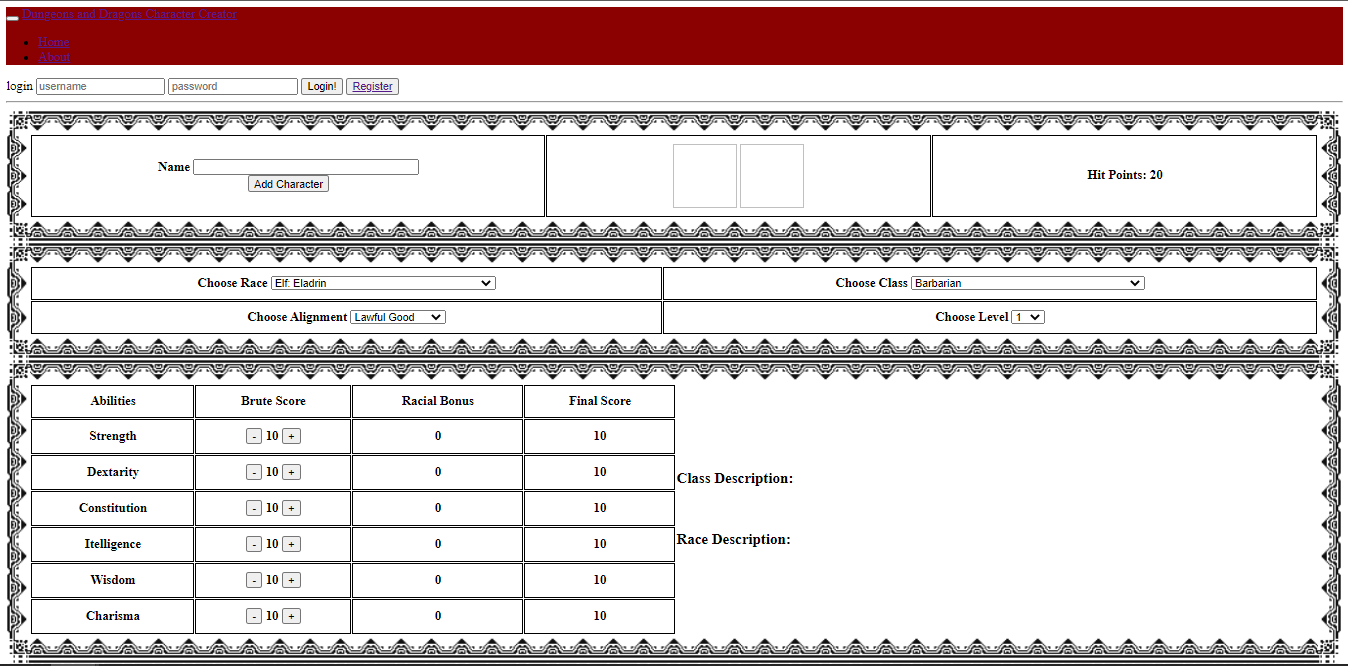
****

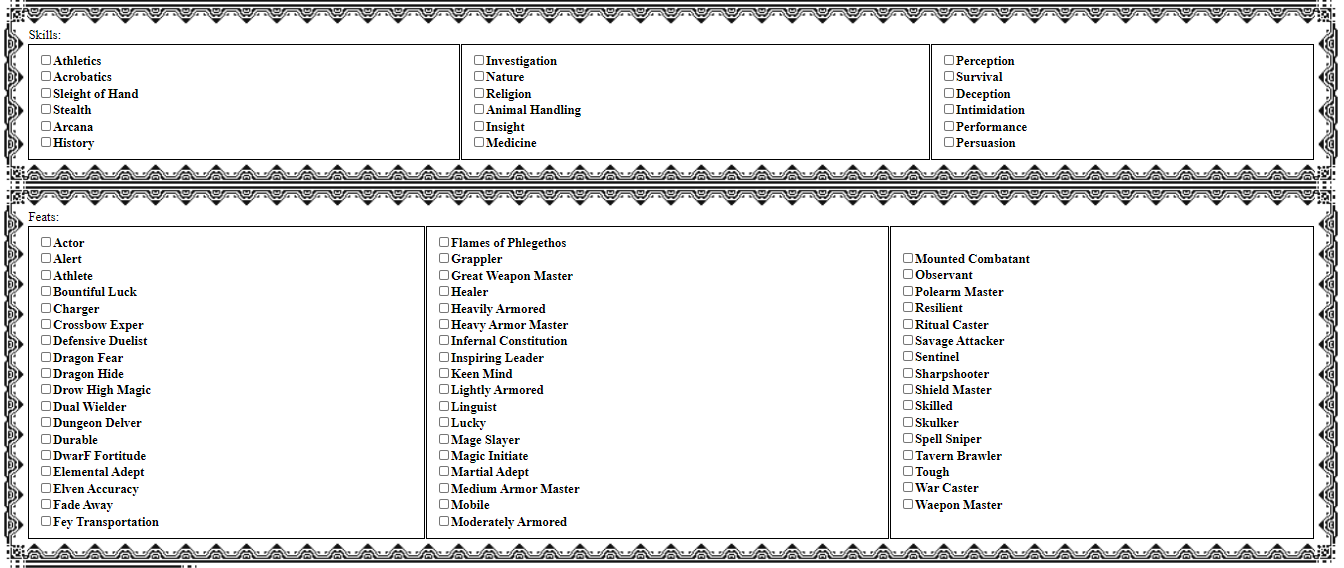
****

****

****

****

****

****

**Group Member Contributions:**

Our project had several areas that required substantial effort. Our individual contributions per area were:

Database design and creation: Faith 30 % Sean % Harvey 70%

Data insertion: Faith 50 % Sean % Harvey 50%

Login system and user tracking via cookies: Faith 0% Sean 0% Harvey 100%

Character Creation page: Faith 10% Sean 80% Harvey 10%

Character display page: Faith 50% Sean 0% Harvey 50%

Online help page: Faith 100 % Sean 0% Harvey 0%

Main page: Faith 100 % Sean 0% Harvey 0%

About page: Faith 100% Sean 0% Harvey 0%

Technical document: Faith 100% Sean 0% Harvey 0%

Preparing PowerPoint presentation: Faith 70 % Sean 30% Harvey 0%

Preparing Report: Faith 50 % Sean 0% Harvey 50%

Video: Sean 100%

**Lessons Learned:**

One of the major hurdles we faced was mass data entry. While enterprise solutions have sophisticated tools to enable access to large amounts of preexisting data, we were forced to pull data from multiple websites and create SQL insertion statements. This process ran over the time we budgeted for it and put the pinch on the time we had to develop tools to utilize the data. In future projects we would either cut back on the data used by our project, make more of an effort to find tools to expedite the entry of so much data, or budget more time to this part of the project.

Another obstacle our group faced was the learning curve associated with using SQL data in our ASP.NET application. Different methods exist for interacting with the SQL database in different ways, and finding the right way to read or write data was often challenging. Unfamiliarity with C# also led us to some brute force solutions that were suboptimal - our Display Character code behind page involved a lot of copying and pasting of code - given more time this would be a prime target for refactoring.

Finally, we encountered problems using the Azure cloud solution, due mostly to our unfamiliarity with it compared to other SQL servers. Some group members were unable to connect to our database through Visual Studio, and it took a lot of troubleshooting before we discovered that Azure offers a separate solution called Azure Data Studio. This was another frustrating time sink that we would have preferred to avoid.

**Possible Future Work:**

Our current login system stores passwords unencrypted in our SQL database. Given more time, and if our app was likely to see any use outside of an academic project, this would require a more sophisticated approach.

Several parts of our project feature repetitive or inefficient code due to time constraints and unfamiliarity with the technologies we chose to employ. One area for future work would be to refactor this code, with a particular focus on extracting methods.

The next step in our app’s development would be to introduce a system to level up characters that users enter. This would involve a new interface that allows users to modify existing data and guides them through the required changes at each level.