**Intro to PHP**

**Final Project Summary**

**Student:** Reeve Jarvis

**Course code:** DGL-113 DLU1

**Instructor:** Brad Best

**Due Date:** 12/08/2021

**Submitted:** 12/08/2021

**USAGE INSTRUCTIONS:**

1. Create a new database in phpMyAdmin named: rjarvis\_simpsons\_archive

2. Import rjarvis\_simpsons\_archive.sql dump file located in repository

3. Load webpage: localhost/dgl-123-finalproject-rjarvis

For this project, I have decided to go with Option A. I wish I could have had the time to work on a more bespoke project idea, but time constraints held me back from this. I am planning to explore what we have learned and apply it to a personal project of mine in my own time. With that said, I feel like I have applied the techniques learned in this class to produce a feature-rich replication of the example webpage provided. I have reverse engineered the site, creating a near-exact clone, and taken the liberty to add some extra features that I felt would be beneficial.

My first step was to reverse engineer the site. I did this by inspecting and accessing the source code from my browser dev-tools. This allowed me to gather the produced html code, CSS stylesheet, and image files I needed. I copied these files into my repository and examined them to brainstorm how I might replicate this in a more dynamic way. I produced some simple php scripts to output the character list based on the local json data we were provided. I also wrote the required code to output character cards dynamically using this data.

To add to the overall functionality, I decided to add some audio clips for the initial characters provided. To make the application more dynamic and useful, I created a database to store the character data and filled it with the character information we were provided (**SEE ABOVE FOR USAGE INSTRUCTIONS)**. I created functions to update the local json character data from this database, to keep a current locally stored copy of this data. I thought this may be useful to have and populate the characters from. After deciding to connect this application to a database, I thought that a useful new feature would be the ability to add new characters to the list. To accomplish this, I added a new page with another form for the user to fill out. It includes fields for all the character attributes we were provided, and requires that a first name, last name, and an image file be uploaded. That brings me to my next feature, file uploading and storage. For my add-character feature, I thought it would not make sense to be able to add characters without a visual representation of them (since this is the way all other character cards are presented). I implemented a file upload script that would validate, and handle image files submitted by the user. If required fields are filled out, and a valid image file is provided… the image will be added to the repository image folder, the character details will be added to the database (empty fields will be replaced with null) and the user will be redirected to the homepage where they should see the newly added character in the list. To accurately link the image file I am storing the relative file path in the database. Local character json data is updated at the start of my main script so it is always current. When characters are selected, and the show characters button is pressed, I am dynamically generating character cards using the stored data, outputting all attributes that are not null.

For the unit tests, I created some tests in a separate file to test my functions. I had trouble making sense of how to properly implement this test for some of my features, but I left some work to show my attempts. I hope this satisfies the requirements. I was very diligent throughout my coding and tried to squash any issues I encountered while I did my work. I also tried to add validation throughout my code to ensure proper usage. I hope I was able to cover this, but I expect there may be a few missed details. Overall I feel like I put a lot of effort into this assignment, and I am happy with my work.

My general goal was to try and increase the functionality of this application as much as possible using the techniques we have covered in this course. I had some fun coming up with solutions to my ideas. I encountered some difficulties along the way, trying to figure out how to properly handle the image storage, as well as with the database connection. I handled this by doing some extra research online through stack overflow and browsing through posted ideas. Whenever I got stuck, I also went over our course material as a refresher to try and make sense of what may be going wrong. In the end I was able to accomplish most of what I set out to do with this assignment. I feel like I was able to achieve a good finished product in the end. I think the user interface is straightforward enough to the user, and that I applied the concepts we learned in this course to create code that is readable and for the most part concise.

If I were to have the time to improve it, I would handle the database and table creation from within my php code. However, this is not something we covered the details of in this course so I would have to investigate this further. There may be some underlying issues with my code, but I hope nothing too drastic. I hope that you can view and operate my application correctly. I have left instructions regarding the database creation at the top of this document, as well as the readme and in the head of most pages. I based my database implementation off the assignment we were given. I tested that it would work on other computers in my home, but I hope this translates to you. If not, I hope you can see the effort I put in to this project. I look forward to hearing any problems you may have encountered.