2016

Edinburgh Napier
University

40110560

[PLANESWALKERS' GUIDE]

This document contains the report for first coursework - a web application for the Advanced Web Technologies module.

Contents

Introduction – Planeswalkers' Guide

As a big fan of the card game – Magic the Gathering, I wanted to create an application that would help people familiarise themselves with some of the main personalities around which the plot of the "Multiverse" is based.

The Planeswalkes's Guide is a web catalogue created to provide information about some of the main characters of the world of Magic the Gathering. At the moment there are around 20 planeswalkers in the system, with the option for expansion. More information can be added to the JSON file containing the planeswalkers' data.

The app is written in Python utilising the Flask framework with the help of Jinja2 templating, HTML, JavaScript and Bootstrap framework.

Design and Implementation

As suggested in the workbook, the Bootstrap Framework was a great starting point and provided some the functionality regarding the looks of the website.

The landing page of the website consists of a slideshow and some information about the general story of Magic the Gathering. The slideshow is generated in the jinja template file by looping through the randomised contents of the 'images/backgrounds' folder. This way everytime the page is loaded everything but the first picture in the slideshow is random. The template also uses the names of the picture it iterates through in order to create links which point to the corresponding character's page.

The Planeswalkers' page contains an image for each of the characters which is a link to their individual pages. The jinja template takes the contents of the 'images/walkers' directory as a list and loops through them, generating the divs with the images and links.

At first I tried using Python dictionaries to store the information for each character, which worked great but separating code from data is always good practice. Thus I moved the data from the dictionaries into a separate JSON file. Now the application reads that JSON files and converts their contents into Python dictionaries.

The individual pages for each Planeswalker reuse a single jinja template. The views.py file has a function which checks the URL /planeswalkers/* (* represents anything) for a name of a planeswalker. It compares the last part of the URL against a list of the names of each planeswalker which are contained in the JSON file. If the url contains one of the names from the list it redirects the user to the corresponding Planeswalker page. The template goes through the dictionary for the planeswalker and displays the information related to them. The data for the planeswalkers also contains the names of the images for different worlds that character has visited (displayed in another slideshow) and their connections to other characters.

If an invalid URL is entered by the user, they are redirected to a custom error page for error – 404.

Even though the application uses Bootstrap some additional work had to be done in order to make the website completely responsive for most devices.

Enhancements

Data

At the moment the application relies on several JSON files for its contents. By adding additional data and information about the different planes(worlds) in the story the user experience would be improved. The information about the planes could contain their history, the different races that live on each plane like elves, orcs, goblins etc.

Data Storage

The Planeswalkers' Guide would greatly benefit from the inclusion of a database. This would allow easier access to more data. More information about each planeswalker could be stored in it as well as information for the different worlds of Magic the Gathering. Another aspect of the website that would get improved by the addition of a database is that it would be possible for a search to be implemented.

Comments

The implementation of comments would give a chance for the users to interact with the page and leave their opinion which would improve the application's social aspect. Disqus would be a great technology to include comments under each of the characters' pages.

Configuration

The application can be improved significantly by the addition of a configuration file and a log file in which everything would be recorded.

Critical Evaluation

Overall, the website covers most of the topics which were discussed in the workbook provided in the module. I feel like the application is way too simple and I could have added more content and/or additional functionally to make it a bit more complex. It runs on Python using the Flask framework. The app uses a variable in its URL hierarchy in order to display the correct Jinja2 template for the planeswalker's page. I think it turned out pretty good that I was able to utilise a single template and feed it different information in order to generate the pages.

In addition, I used tinyjpg.com in order to compress the size of all the images. This greatly improved the time that was needed for the pages to load all of the information.

At the moment the application handles 404 errors by redirecting the user to a custom page. It doesn't handle any other errors. The application is also missing a log file and a configuration file.

Personal Evaluation

When I started working on this assignment I wasn't familiar with Python at all and had to figure a way to store the information for my web app. The first thing that worked for me were dictionaries which were easy to work with and I could have a dictionary of dictionaries (2d dictionaries) which allowed me to store the information for each character and display their information when a condition was met. At a later stage I was informed that JSON and Python dictionaries have an almost identical structure which made it easier for me to separate my data from my code.

Python was fairly easy to get used to as the syntax is very straightforward though different the languages I was comfortable with like Java, C#.As I had some web development experience with Drupal thanks to my placement I felt confident that I would be able to satisfy the requirements of this coursework even though I had no experience with Flask or Python prior to starting this module.

By constantly revisiting the course workbook I was able to overcome the challenges of working with Jinja and Flask. The official documentation for both also came in handy when I had troubles figuring out if certain things were possible. For example – "if statements" in Jinja2.

References:

Bootstrap:
https://bootstrapdocs.com/v3.3.5/docs/

Jinja2:
http://jinja.pocoo.org/docs/dev/

Information about Planeswalkers:
http://magic.wizards.com/en

Stabdard CSS media queries:
https://scotch.io/tutorials/default-sizes-for-twitter-bootstraps-media-queries

Gallery tutorial:
https://www.youtube.com/watch?v=yO_XNCsBIsg

Nested Loops in Flask:
https://teamtreehouse.com/community/nested-loops-in-flask-how-to-iterate-and-make-nested-lists

Custom Error pages:
http://flask.pocoo.org/docs/0.11/patterns/errorpages/

Image used for the /Planeswalkers/ page:
http://gamessaga.com/magic-the-gathering-best-planeswalker-for-mtg-players/

Images for slideshows and individual planeswalker pages and the favicon:
http://magic.wizards.com/en