|  |
| --- |
|  |
| Wandile’s Airliner Collection Web Application|Coursework 1 |
| SET09103, Advanced Web Technologies |
|  |
| **Wandile Sobopha, 40113996** |
|  |

[Introduction 2](#_Toc466977848)

[Design 2](#_Toc466977849)

[Critical Evaluation of Web Application 4](#_Toc466977850)

[References and Resources 6](#_Toc466977851)

Introduction

This report discusses the web application implemented for the coursework. It will go through the details of the design, possible enhancements that could be made to the app and an evaluation of the implementation.

The web app is called ‘Wandile’s Airliner Collection’ and it is a catalogue of aircraft that are used by airlines in commercial service, hence the word ‘airliner’. In this application, users can look through the top airliners of today and see facts about each of the aircraft. It is possible to browse, and find particular airliners by their attributes; Number of engines, number of seats, range, airline operators and by the body size of each of the aircraft.

The implementation of this collection was inspired by a personal interest in aircraft and the airline industry, and also by some sites that show a collection of photographs of airliners taken by individuals around the world [1] [2] [3] [4].

This is where the idea came from – just like I collect posters of the aircraft, I wanted to create a web app where users can browse through different airliners and see quick facts about them.

In line with the module, the Flask framework, written in Python has been used to implement the application. HTML, CSS and Jinja2 are all used together within Levinux to present the application.

Design

Once the decision had been taken that the online catalogue would be of aircraft, the first step taken in the design process was to establish how the application would be appealing to a user – What kind of content a user might be interested in and what functionality would best enable a user to be able to find items within the collection. Looking at existing airliner sites, I knew that I wanted to have a picture collection of the airliners. The idea then developed into having a collection of pictures but with some information to go along with each of the pictures.

I decided that the best way to categorise the collection was by the aircraft manufacturer. From this, the idea of having a navigation bar with links to the main aircraft manufacturers arose. The idea of the navigation and its links was foundation of the URL structure of the app. It was already known that a landing page would be needed for the app but at this time I had not decided on what content it would have.

From the ideas that had been developed, several designs of the interface and different pages of the app were sketched on paper, to get ideas of the possible layout and to determine the functionality that the app would have in order to include use of requests, redirects, responses, static files, templates etc, as covered in the course material.

After that the structure of the application was created within Levinux by creating the main directory for the application and then all the relevant directories within it to house the templates, static files etc.

From the paper designs, it was clear that the application was going to be mainly image based. This was the most suitable and aesthetic way of presenting the collection of airliners. It was decided that a suitable data size of to include in the app prototype in order to demonstrate its functionality would be at least three models under each aircraft manufacturer and a maximum of six models. The manufacturers and models used would be those which are most used in the airline industry, e.g. under Tupelov the models: Tu-204, Tu-154 and Tu-144 and under Boeing: 737, 747, 757, 777 and 787 as the manufacturer with more models being used for commercial service.

The HTML for each of the pages was then constructed once the sketches had been finalised. The aim was to make use of templates as much as possible especially for the pages were the layout would be the same, such as for the different manufacturer pages. Once the HTML was written up and the application had a skeleton for the data to fit into, the data was inputted, the images being downloaded from various sources online and the facts/figures on each of the aircraft models also being sourced online [5].

The ‘Browse’ page was implemented last as it required all of the other data to be inputted into the rest of the pages. The idea around the ‘Browse’ page was for it to be a home/landing page that the user arrives at when they first open the app, and from the this page they should be able to see all of the data within the app, ‘browsing’ it for it using certain criteria or attributes. The users can get different results in the catalogue of airliners depending on the options they choose and submit for drop down lists. This makes the Browse page functionally, a useful page for a user.

Enhancements

There are a couple of features that were not implemented because of time constraints or because of not having the technical ability/knowledge to implement them.

The enhancements that I would make to the application are as follows:

* Use of templates and inputting data into pages more efficiently: The aircraft manufacturer pages – ‘Boeing’, ‘Airbus’ etc, all have the same layout and present data in the same way. The data was individually inputted on each of the HTML pages for these categories, however what would have been more efficient was if the data was rendered dynamically – having a set template for the layout and then inserting the images and text depending on which manufacturer page was accessed, this would eliminate manually coding in the images and relevant facts into the HTML for each model under the respective manufacturer.
* Similarly to the previous point, the ‘Browse’ page contains a lot of code – mainly if statements and blocks of code that is then executed depending on the option which is selected in the drop down list. This makes the page slow to load. The improvement that could be made to this again is perhaps through the use of templates and to render that code dynamically so that it doesn’t have to be manually inputted on the page, and the page doesn’t have to load all this code that it might not use.
* A page for each of the models that has more information about the model, such as more pictures of the model and a short description of the model. Likewise for the airlines that are mentioned as the operators of the models. The images and quick facts were initially intended as teasers that had links to further information about the models and the airlines, however it was realised that this would need a bigger time than planned.
* A search bar at the in the header – This would allow the user to type any keyword and be able to view the relevant aircraft/airlines that matched the search criteria. This would make the web application more user-friendly and present another way of narrowing down data.

Critical Evaluation of Web Application

The application fulfils the aim of the coursework. It is a functioning catalogue web catalogue of airliners which are sorted by manufacturer. The app has a way of finding specific models based on certain parameters, using dropdown forms. The app can request the information from the form in the HTML and then return a response depending on the option selected, therefore showing the relevant models on the ‘Browse’ page.

The application makes use of Flask aspects covered in the course material. It has a carefully designed URL hierarchy, making use of routing with each of the routes returning the appropriate responses. Templates have been used to generate HTML pages for the application however, the use of templates could have been used more efficiently to make better reuse of code – There are parts where the same code has been used across other pages. This was a technical constraint, where I failed to use the correct method to successfully achieve dynamically rendering code depending on what manufacturer page was opened.

The application has a good directory structure; all the relevant files were put in relevant directories such as the static files and templates, so the application was able to successfully read the files needed.

Overall, the application is quite simple but fulfils the purpose of it being a catalogue. A good sample size of aircraft was used to demonstrate how it works. One thing to mention about the design is that it may feel more like a website rather than a web application due to having the typical header at the top and a navigation bar. With more time perhaps the application could be improved and include more complex features and the code could be written more efficiently.

Personal Evaluation

When starting this application, I felt very ambitious. I had the great idea of the type of catalogue I wanted to design and I was determined to implement the design that I had in mind using what I had learned in the course material. I think that was a good starting point, as it set me up with an ethic to succeed in what I was about to start.

Through building the application, I was able to get to know certain aspects of the course material very well as I needed them in my application. At first, I did not understand the code demonstrated. It took me some practice to fully understand and make use of the material in my application. I have improved my Linux skills though using Levinux and I have improved in areas such as git through learning how to set it up and then using the right commands to push my work to a repository.

I was particularly struggled in reusing blocks of code. I feel like the idea that I had of rendering content depending on which page was opened might have been more complex than I would understand. I made an effort to try and make it work but I also lost a lot of time in trying to make this one thing work. When it proved too difficult, I chose the simpler method I has in mind which was just to rewrite the code where it was needed.

Overall, I think that I have done quite well in building the application. I did meet the aim of creating a catalogue and I was able to meet the deliverables. I struggled with time keeping, to make sure that the application would be built within the certain time frame given. I think that I could improve on this by having a project plan of what I will work on and allocate a time limit to each section of the project. However, I have learned a lot of new things technically through this project, especially at times where I was stuck and had to find solutions online.

References and Resources

1. <http://www.airliners.net/>
2. <http://www.jetphotos.net/>
3. <https://www.planespotters.net>
4. <http://www.airlinerphotos.com/>
5. <https://www.wikipedia.org/>

<https://www.sitepoint.com/a-basic-html5-template/>

<http://thetravelinsider.info/airplanetypes.htm>

<https://en.wikipedia.org/wiki/Wide-body_aircraft>

<https://en.wikipedia.org/wiki/Narrow-body_aircraft>

<http://www.airpartner.com/en/aircraft-guide/>

<http://www.freeiconspng.com/uploads/plane-icon-png-images--pictures--becuo-8.png>

<https://realpython.com/blog/python/primer-on-jinja-templating/>

<http://stackoverflow.com/questions/4620672/copy-and-paste-content-from-one-file-to-another-file-in-vi>

<http://jinja.pocoo.org/docs/dev/templates/#extends>

<http://stackoverflow.com/questions/7537439/how-to-increment-a-variable-on-a-for-loop-in-jinja-template>

<http://jinja.pocoo.org/docs/dev/tricks/>

<http://stackoverflow.com/questions/33355159/how-can-i-dynamically-render-images-from-my-images-folder-using-jinja-and-flask>

<https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-ii-templates>

<http://damyanon.net/flask-series-templating/>

<http://stackoverflow.com/questions/6181935/how-do-you-create-different-variable-names-while-in-a-loop-python>

<http://stackoverflow.com/questions/9610393/looping-over-subset-in-jinja>

<https://learnpythonthehardway.org/book/ex32.html>

<http://stackoverflow.com/questions/13850519/margin-on-inline-block-element>

<http://www.w3schools.com/html/html_form_elements.asp>

<https://learnpythonthehardway.org/book/ex51.html>

<http://stackoverflow.com/questions/11556958/sending-data-from-html-form-to-a-python-script-in-flask>

<http://stackoverflow.com/questions/1976651/multiple-level-template-inheritance-in-jinja2>

<http://stackoverflow.com/questions/37116835/python-flask-request-form-not-working?noredirect=1&lq=1>

<https://www.tutorialspoint.com/python/python_lists.htm>

<http://stackoverflow.com/questions/12502646/access-multiselect-form-field-in-flask>

<http://stackoverflow.com/questions/19578613/posting-data-on-flask-via-form-is-giving-400-bad-request>

<http://stackoverflow.com/questions/19830607/how-do-i-redisplay-the-same-dropdown-list-with-the-previous-selection-instead-of>

<https://upload.wikimedia.org/wikipedia/commons/c/c6/SAS_Scandinavian_Airlines_Boeing_737-800_%28LN-RRJ%29_taking_off_from_Stockholm_-_Arlanda_Airport.jpg>

<https://upload.wikimedia.org/wikipedia/commons/2/22/Icelandair_Boeing_757-256_Wedelstaedt.jpg>

<https://upload.wikimedia.org/wikipedia/commons/1/15/All_Nippon_Airways_Boeing_787-8_Dreamliner_JA801A_OKJ_in_flight.jpg>

<https://upload.wikimedia.org/wikipedia/commons/c/c4/British_Airways_G-BNLU-2008-09-13-YVR.jpg>

<https://upload.wikimedia.org/wikipedia/commons/f/f5/Emirates.b777-300.a6-emv.arp_lhr.jpg>

<https://upload.wikimedia.org/wikipedia/commons/4/43/Delta_Air_Lines_B767-332_N130DL.jpg>

<https://en.wikipedia.org/wiki/Airbus_A318#/media/File:Airfrance.a318-100.f-guga.arp.jpg>

<https://upload.wikimedia.org/wikipedia/commons/8/81/A318.jpg>

<https://upload.wikimedia.org/wikipedia/commons/5/54/A7-ALA_%2819397355348%29.jpg>

<https://upload.wikimedia.org/wikipedia/commons/4/40/Airbus_A321_-_myAustrian_%281%29.jpg>

<https://upload.wikimedia.org/wikipedia/commons/a/a9/Egyptair.a330-200.su-gci.arp.jpg>

<https://upload.wikimedia.org/wikipedia/commons/e/eb/D-AIHD-2008-09-13-YVR.jpg>

<https://upload.wikimedia.org/wikipedia/commons/d/d6/Air_France_Airbus_A380-800_F-HPJB.jpg>

<https://upload.wikimedia.org/wikipedia/commons/5/53/J-AIR_CRJ200ER%28JA206J%29_%283504044273%29.jpg>

<https://upload.wikimedia.org/wikipedia/commons/1/12/CRJ-900_Air_Nostrum_EC-JTU_01.jpg>

<https://upload.wikimedia.org/wikipedia/commons/a/a0/Bombardier%2C_BD-500_CSeries_CS300%2C_C-FFDK_-_SIAE_2015_%2818887460245%29.jpg>

<https://upload.wikimedia.org/wikipedia/commons/d/d9/Flybe_dash8_g-jecl_takeoff_manchester_arp.jpg>

<https://upload.wikimedia.org/wikipedia/commons/a/ad/Polish_Air_Force_Tupolev_Tu-154_Dmitry_Karpezo-2.jpg>

<https://upload.wikimedia.org/wikipedia/commons/8/88/RIAN_archive_566221_Tu-144_passenger_airliner.jpg>

<https://upload.wikimedia.org/wikipedia/commons/d/d7/RAEerj145.jpg>

<https://upload.wikimedia.org/wikipedia/commons/d/d8/EI-RDB_Embraer_175_Alitalia_BCN.jpg>

<https://upload.wikimedia.org/wikipedia/commons/4/41/Flybe-man.jpg>

<http://www.smartcockpit.com/images/xl/EMBRAER-190-195.jpg>