Introduction to Web Science

Assignment 4

PD Dr. Matthias Thimm
thimm@uni-koblenz.de

Ipek Baris Schlicht
ibaris@uni-koblenz.de

Kenneth Skiba
kennethskiba@uni-koblenz.de

Institute of Web Science and Technologies
Department of Computer Science
University of Koblenz-Landau

Submission until: 08.12.2020, CEST 23:59

1 XML 26 Points

Consider the following XML document:

```
?xml version="1.1" encoding="UTF-8" ?>
<book>
    <author><publisher>Macmillan</author></publisher>
    <title>Alice Adventures in Wonderland</title>
    <year>1865</year>
</book>
```

1.1

Solution:
The above xml document is not valid for the below mentioned reasons:

1. In the declaration line, '<' is missing. The correct way is
   <?xml version="1.1" encoding="UTF-8" ?>
2. The element tags for author and publisher are not closed in order. The correct order should be
   <author><publisher>Macmillan</publisher></author>

1.2 Well formed XML

Consider the following information about the best-selling individual books[1]:

Name	Author(s)	Original language	First published	Approximate sales	Genre
The Communist Manifesto	Karl Marx and Friedrich Engels	German	1848	> 500 million	Political philosophy
A Tale of Two Cities	Charles Dickens	English	1859	200 million	Historical fiction
Harry Potter and the Philopher's Stone	J. K. Rowling	English	1997	120 million	Fantasy, mystery
The Lion, the Witch and the Wardrobe	C.S. Lewis	English	1950	85 million	Fantasy
The Adventures of Pinocchio	Carlo Collodi	Italian	1881	> 80 million	Fantasy

Your task is to encode the table above into a well formed XML document.

[1] Based on https://en.wikipedia.org/wiki/List_of_best-selling_books

2 HTML and CSS 27 points

The aim of this task to learn and practice the basics of HTML and CSS. You are required to implement a website that presents your resume. The layout of the website as seen Figure 1. The most left section is **Experiences**, the middle is the section for **Education** and the most right section is **Skills** followed by **References**.

Perform the following steps:

1. Create a HTML file called `index.html`, and a CSS file called `style_display`. In CSS, define the items referring the sections and header and use only `inline-block` for positioning the items.
2. Create a CSS file called `style_float.css`. In CSS, define the items referring the sections and header and use only `float` for positioning. Do not alter the contents in `index.html`.
3. Create a HTML file called `index_table.html`, and a CSS file called `style_table.css`. This time, you are supposed to implement the layout by using tables in HTML. However, for styling the cells and the properties of the table, you need to edit the `style_table.css`.

Three steps that you will implement should output same layout and content. However, small margins between the results could be neglected. The background color of the sections is `#ccc`.

For this task, don't use frameworks such as bootstrap. You are only allowed to use standard HTML and CSS items.

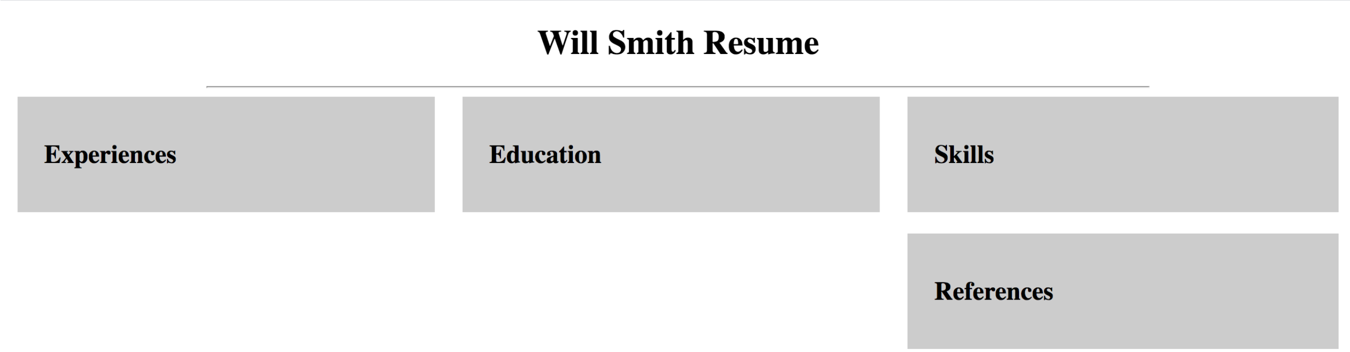

Figure 1: The website page you will implement

3 Dynamic Web Content 27 points

You are provided with simpleapp.zip that contains a simple HTTP web server that we have designed for you. It has a simple HTTP registration page (localhost:5000/form) that asks you to enter the first name and email ID to register.

Your task is to write a python script client.py that automates the process of registering. The simpleapp.zip contains contacts.txt and alternate_contacts.txt that have the different names and email IDs that you need to register automatically through your script.

In the the template folder, you will find an html form whose actions you need to automate through your script. The script should look for error messages if the entry that you are trying to register is already in the system or not. If no error message, your entry gets stored in the server. The script should save all the responses from the server into lists and count the number of successful and unsuccessful attempts to register.

We give the following code snippet as starting code. When you execute this code, you will have a compiled errors because it is incomplete. Please replace "TODO" with your codes and implement the methods in the __main__. First you need to run the client script for contacts.txt, and then after restarting your server you need to run it for alternate_contacts.txt. You should get the following error message from the server:

```
sqlite3.Warning: You can only execute one statement at a time.
```

Please briefly explain, what we try to achieve here and what could be the fix for preventing such message from the server side.

```python
if __name__ == '__main__':
    URL = "TODO"
    fname = sys.argv[0]
    contacts = read_file(fname)
    "TODO"
    for contact in contacts:
        status, text = register(contact=contact, url=URL)
        print(f"Status code from the app {status}")
        print(f"Text is {text}")
        list_size, contains_success_msg = parse_response(text)
        "TODO"
    total_request = len(contacts)
    assert error_count + success_count == total_request
    plot_timeline(list_sizes, output_file="list_size_timeline.png")
    plot_bar(success_count, error_count, output_file="results.png")
```

You will need to install the required libraries by using command pip install -r requirements.txt. Then, you will need to run server_start.sh that starts the server. Please keep in mind that when you will test your code, you need to refresh the database and restart your server by running server_start.sh.

You **are allowed** to use the following libraries for your implementations: `matplotlib`, `regex`, `requests`, `sys`.