ASSIST Software

1 Tipografiei Street | 720043, Suceava, Romania

Homework for “Weekends”

Subject: Web Scraper

Document Version: 1.0

Author: Victor JEMAN (Head of Front-End Technologies)

Overview

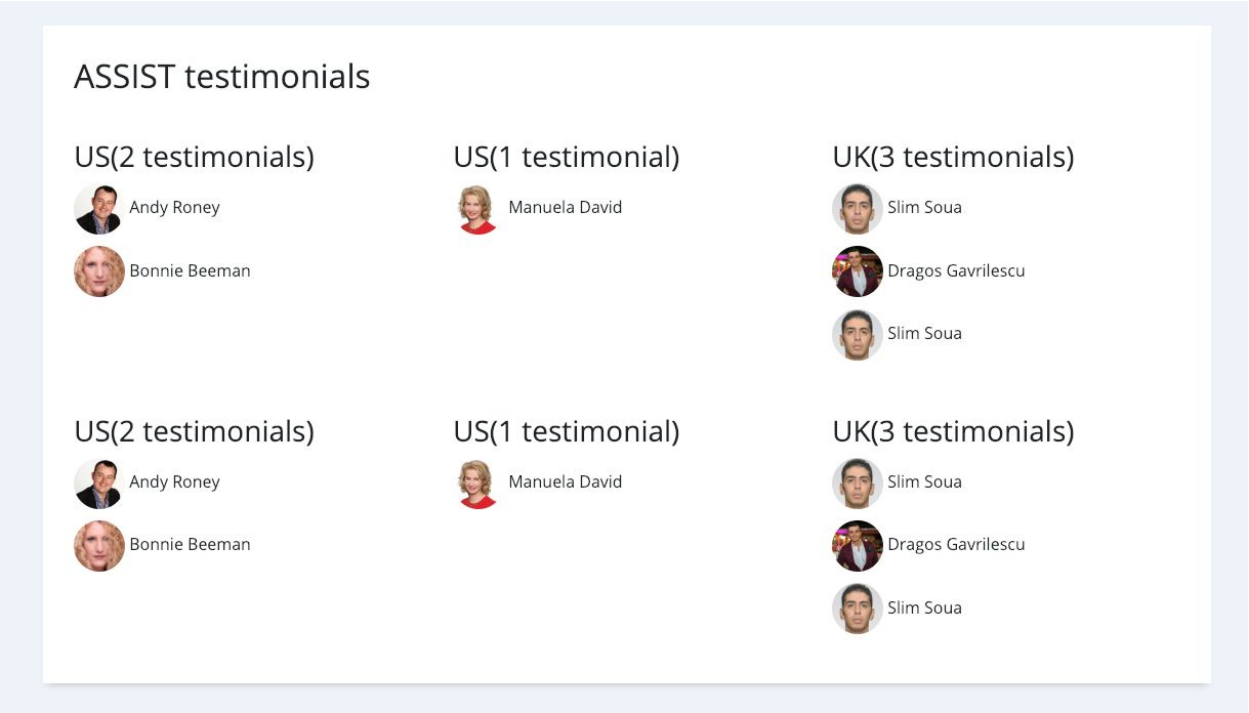
We want to build a web scraper which could tell us how many testimonials the ASSIST website has and sort them by countries. Our web scraper should have a basic UI that would display nicely all the required information.

Assignment

1. Implement a basic web scraper in **node**.
2. Use only the following dependencies: cheerio, express, request-promise
3. Using the scraper to **extract the HTML** from https://assist-software.net/testimonials.
4. From **first testimonials** page, extract all the testimonials.
5. Sort the testimonials by **country**.
6. If an author belongs to multiple countries, display him under each country.
7. Show the **amount** of testimonials for each country.
8. For each testimonial show the **picture** and the **name** of the **author**.
9. Make use of latest ECMAScript specifications.
10. Upload your solution to **Github**.
11. **EXTRA POINTS:** Split the functionality in multiple files.
12. **EXTRA POINTS:** Sort the testimonials from **all pages**(4 pages), not only the first page.
13. **EXTRA POINTS:** Write some unit tests for your functionality.

Tips

1. You can see here a basic UI example for this assignment



2. You may need a basic node server to serve the above dashboard once you sorted the information in columns
3. Learn more about web scraping.

How can I review my code?

Take a look at this **Code Review Checklist** before you hand your application for review

- ❏ The code works
- ❏ The code is easy to understand
- ❏ Follows coding conventions
- ❏ Names are simple and if possible short
- ❏ Names are spelt correctly
- ❏ There are no usages of 'magic numbers’
- ❏ All variables are in the smallest scope possible
- ❏ There is no commented out code
- ❏ No code can be replaced with library functions
- ❏ Required logs are present
- ❏ Frivolous logs are absent
- ❏ Debugging code is absent
- ❏ Code is not repeated or duplicated
- ❏ No complex/long boolean expressions
- ❏ No empty blocks of code
- ❏ Catch clauses are fine grained and catch specific exceptions
- ❏ Loops have a set length and correct termination conditions
- ❏ Blocks of code inside loops are as small as possible

- ❏ Design patterns if used are correctly applied