

Gocomet Assignment

Instructions

1. Please ensure that you follow all the coding conventions.
2. When implementing this solution, please use git.
3. Create a Ruby on Rails application/python and Django, etc. (MVC Framework is must) and use **Relational Database**.
4. Please use twitter bootstrap for UI. (Bonus Points if you use React.js/Angular.js)
5. As a part of this evaluation you're required to provide a timeline schedule in a separate document (doc, txt, odt), structured into development phases with description and time spent on each phase.
6. You are also recommended to demonstrate your skills & expertise by enhancing the particular part of the web application (User Interface, Security, Backend, etc) that you hold expertise in.
7. The main stress will be placed on: code logic & structure, coding standard, timeline report, API usage and your innovation.
8. Any attempt at cheating, plagiarism or any other malpractice will result in immediate rejection of your application.

How to do

1. Divide the code in 3 phases
2. Phase 1 can have minimum functionality (Is should atleast run)
3. Phase 2 can have more advance features.
4. Phase 3 can have other things that you want to implement to show your skills

Prerequisite

1. MVC Framework like (Django/ROR etc)
2. Learn how to crawl/scrape webpages

Problem Statement

You need to create a Scraping/Crawling application that can crawl medium.com.

Medium.com Structure

- Medium.com has various **TAGS** such as ([cooking](#)/[culture](#)/[business](#)) etc.
- Each blog has many details you need to focus on creator, details, title, blog, tags, responses. For example there is a blog "[The future of Cyber Security in the Fintech Era](#)".
 - Creator: [Almog Ohayon](#)
 - Title: The future of Cyber Security in the Fintech Era
 - Details: 5 May, 5 min read
 - Blog: Content of blog(HTML)
 - Tags: Bitcoin, Blockchain, Ethereum, Cybersecurity, Fintech (you can find it at the bottom of the page)
 - Responses: In this case at the bottom of the page there is only one response by [Obeid Ruff](#), (responses are kind of comments)

Create an Admin Panel

NOTE: No use of libraries/gems are allowed, only use libraries if it is really required

You need to create a single page admin panel with following features (use ajax only) (150 points + 80 bonus)

1. There should be an input button where a user can input a tag name and on submit it should send a request to backend server. **[10 points]**
2. Upon receiving the request your crawler should crawl 10 latest blogs with that tags and parse the data like Details, title, blog, tags, responses. **[50 points]**
3. Show the data in frontend **[40 points]**
 - a. Show list of blogs with creator name, title, and details
 - b. While data is being crawled, in the frontend display the blogs that are crawled. It should be real time(without reloading)(e.g. Let say 3 blogs are crawled so it should display like this.)
 - i. <blog, title,creator, details>
 - ii. <blog2, title,creator, details>
 - iii. <blog3, title,creator, details>
 - iv. crawling..
 - v. pending..
 - vi. pending..
 - vii. pending..
 - viii. pending..
 - ix. pending..
 - x. pending..
 - c. When user clicks on a blog, it should open in a new page (title, blog, tags, responses and source url) (as it is displayed on the Medium Blog).
 - d. There should be an option of “next 10 blogs”, it should crawl next 10 blogs with the above tag.
 - e. When user click on a tag(inside a blog) then it should crawl 10 latest blogs of that tag.
 - f. Display how much time it took to crawl individual page.
4. If no blogs are found it should suggest other tags related to the tag that user has searched (Bonus if you think of algorithm or a way to display Suggested Tags/ Typos) **[10 points] + [10 Bonus Points]**
5. Display Search History. **[10 points]**
6. All crawled data, tags and responses should be stored in database with proper associations. **[10 Points]**
7. Show your creativity (FrontEnd/BackEnd),If you can think of any interesting feature then do implement it. **[30 Bonus points]**
8. ReactJs/AngularJS **[20 Bonus points]**
9. Bonus: Test Cases **[20 Bonus Points]**