**Assignment 1: A web application for exploring Stack Overflow questions**

**Context:** suppose that I am interested in the most hot questions of a particular tag *T* on Stack Overflow (*https://stackoverflow.com*). I would like to browse Stack Overflow questions and their answers that are related to *T (https://stackoverflow.com/questions/tagged/javascript)*. I would like to see the titles of the **10 newest questions** and the **10 most voted** questions posted **in the past week** related to tag T, on **one page**. In this way I can easily keep track of the relevant questions. In addition, I would like to be able to read the full information of these questions in a convenient way.

**Task 1: implement a simple web application meets following requirements:**

* As a user, I can input a tag *T* in a textbox and click submit button, the web application shall extract from stackoverflow.com the 10 newest questions, as well as the 10 most voted related questions posted in the past week related to tag *T* as **one** merged list sorted based on their creation day in descending order.
* The web application shall display the title, creation date, and vote of each question as default.
* The web application shall display the full question thread after a user clicks on one of the titles using collapsible (<https://www.w3schools.com/howto/howto_js_collapsible.asp>). The full question threat includes the question body, its answers, and all associated comments for both question and answer if applicable. The creation date and vote for the question, answers, and comments should be displayed as well.
* The response time of the web application shall be less than 1.5 seconds given the network condition is good, and the response time should be shown on the bottom of the page.

We can use any language or framework you prefer for the web application. You can also choose whatever approach to extract the questions related to a particular tag, e.g., crawling their Stack Overflow website, using the API provided by Stack Overflow, etc.

**Task 2: wrap up your implementation as a docker image and upload to Docker Hub (**[**https://hub.docker.com/**](https://hub.docker.com/)**)**

Once your application is implemented, create a docker image for the implementation and upload it to Docker Hub, so that TA/Instructor can run the docker image and verify your implementation.

Quick introduction to docker: <https://docs.docker.com/get-started/>

Here is a youtube tuturial how to create a docker image (<https://www.youtube.com/watch?v=Rt5G5Gj7RP0>).

Requirements for docker image:

* All the configurations of the web application should be completed in the created docker image.
* The service for the website should be on by default when the docker image gets started to run.

In short, everything should be ready in the docker image. Once docker image is running in a container, TA/Instructor can go to the website and explore the questions without any further configurations.

**Submit a report to UMLearn includes:**

* + Description on how your application is implemented, including the logic, framework/python used.
  + The docker image link on dockerhub
  + Github repo for the source code of your application
  + Instructions on how to run the website and explore questions related to a tag, e.g., the URL of your web application, how to explore the questions.