Lab 4 Enhanced Search Engine

From Lab 1 to Lab 3, you have implemented a basic search engine, and deployed it on AWS. In this lab, you are asked to accomplish the following objectives:

- Search Engine Enhancement: Here you are given the complete freedom to design your own search engine.
- Code Tidy Up. Here you are expected to release your final project code externally, which requires you to clean up your code and scripts for understandability, maintainability, robustness and ease of deployment.
- Project report. Here you are required to write a project report to document your design, analysis and experiences.

Search Engine Enhancement

To enhance the search engine, you can either choose:

- **an algorithm route**: to implement alternative implementations (hopefully better) of a selected key algorithm in the search engine; or
- **a feature route**: to add more innovative features at the frontend or/and backends.

This part of the lab is open-ended, and you are not required to implement any of the suggestions provided. Regardless of your choice, you must describe your design decisions in the project report.

Whichever direction you take, focus on **depth rather than breadth**. A well-developed and thoroughly enhanced feature is better than a wide set of shallow improvements.

Reference List of Features

1. Frontend
    - Spell Correction
    - Autocompletion
    - Multi-word searching
    - Search suggestion
    - Query phrase interpretation, e.g. compute simple math equation if submitted as query.
2. Backend
    - Complex Ranking System

- Store screenshots or summary for some of the highest ranked web pages
- Optimize search engine data structure, i.e. less storage, faster access, and etc.

3. Aesthetic and User Friendliness
- Minimize number of clicks for each search, e.g. avoid clicking a button to go back the query page.
- Customize results table for mobile devices, e.g tablets, smartphones, and etc.
- Animated logo, e.g. Google Doodles

4. Deployment and Autoscaling
- Running multiple bottle frontends and load balancer in one AWS instance to maximize resource utilization.
- Dynamically launching new bottle frontend by monitoring resource utilization.

5. Performance Improvement
- Optimize the number of connections that can be handled on the server, i.e 2000+ concurrent connections.
- Minimize processing time for each query, i.e. < 1ms. Note that the processing should exclude the network latency.

One-Click Deployment Script and Termination Script

In this Lab, you are required to submit a deployment script for launching your search engine on AWS, and a termination script for shutting down an active running instance from AWS.

The deployment script does not only launch an AWS instance (like Lab 2), but also copy files to the server, and launches the frontend after it is started. By the time when this script is completed, it should return an IP address or public DNS with port number (if not port 80), and your search engine has to be accessible from the public network.

A typical use of the deployment script is following:

1. User specifies AWS credentials in a separate key file;
2. User invokes your deployment script;
3. Deployment script loads the AWS key file, launches AWS instance, copies application files to the new instance, installs packages on AWS instance, and launch the search engine on server.
4. When the server is stable, the deployment script returns the IP address or public DNS of the new AWS instance. Also, the instance ID of new machine should be returned.

5. User accesses the search engine service through the returned IP address or public DNS from browser.

Note that if your search engine requires other 3rd party tools or API configurations, it is **not** required to include it in the deployment script. For example, if you are using Google Login APIs,and you can only specify the redirect URL and origin on Google Console after the public DNS is return from the new instance. In this case, it is impossible to include such configurations into the deployment scripts.

The termination script is similar to the deployment script, except that instead of launching a new AWS instance, it shuts down an active instance. A typical use of the termination script is following:

1. User specifies AWS credentials in a separate key file;
2. User invokes termination script and pass the instance ID from command line;
3. Termination script shuts down AWS instance.
4. Upon completion, the termination script returns message indicating whether the termination process has been completed successfully.

Final Report

In the final report, you must clearly describe your search engine, and what features have been implemented in Lab 4. The final report should have the following sections:

1. Names and student numbers of all members.
2. Describe the design of your enhanced search engine in detail. If you enhanced an algorithm, describe the different candidate algorithms and how they are different from the baseline implementation in Lab 3, and describe the quantitative metric you use to judge the merits of the candidates and how you chose your final candidate.
3. Indicate the difference of your proposed design and completed design if there is any. If the search engine is completed differently than the proposed design, explain why.
4. Explain your testing strategy during the development. Describe how you identify the corner cases.
5. Lessons learned from this project.
6. Describe what you would do differently if you had to do it again. What would you do if you had more time. Did any parts take longer than you thought, and Why?
7. How the material from the course help you with the project.
8. How much time it takes for you to complete each lab outside the lab sections.
9. Which part of the project you think is useful and you believe the labs should spend more time on it.

10. Which part of the project you think is useless and you think it should be removed from the labs when this course is being offered in the future.
11. Other feedback or recommendations for the course.
12. Responsibilities of each member. If you believe that workload is distributed unequally in your group, you may describe the situation in this section.

Deliverables

- IP Address/public DNS of a server that runs your search engine actively for one week after the due date
- Deployment script, termination script and all source files, excluding data files generated for storing data generated by crawler. And a README file describing your code organization.
- Final report in PDF
  - final report should be named with your group number with following format: **group<group_number>-ece326-2025.pdf**

Hints

- Data files generated by the crawler should be excluded from the submission, however, you will need to store the data files on your AWS instance to provides results for your search engine.
- The deployment and termination script may use third party libraries, such as Boto and others. However, you cannot assume such libraries have been installed on the test machine. Therefore, your script are required to detect whether the required libraries have been installed; if libraries are not found from the system, your script should initiate the installation process of the libraries, or simply return an error message to notify the user which libraries are missing.
- You may implement the deployment scripts in Bash or Python based on your preference. You may use multiple files to implement the deployment script. However, the user should only be required to invoke one script.
- When copying files from your own machine to AWS machines through SCP, the strict host key will be checked, which is not preferred for the deployment script. To bypass the host key checking, use `scp -o StrictHostKeyChecking=no -i keyfile.pem local-file ubuntu@IP:~/remote-file`
- For the termination script, you may require the user to enter a target IP address, instance ID, and etc, to identify which instance to terminate.
- when installing packages on AWS instance through 'apt-get', you may be required to answer 'yes' to continue the installation. To avoid such prompt, always append '-y' option to "apt-get" to enforce yes by default.

Submission

Compress all your files, including the source codes and text files, and name it **lab4group\<group_number>.tar.gz** or **lab4group\<group_number>.zip**

Submit it on Quercus under Lab4