CSC326H1F – Programming Languages – Lab 4 – Group 24 – Final Report

SECTION 1 - Names and Student Numbers

FIRST NAME	Sai Harshita
LAST NAME	Tupili
STUDENT #	1002938556
SECTION:	LEC 0101 PRA 0102

FIRST NAME	Marinette (Mei Jia)
LAST NAME	Chen
STUDENT #	1003042726
SECTION:	LEC 0101 PRA 0102

SECTION 2 - Design of Enhanced Search Engine

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.

The features that were implemented in lab 4 were autocorrection, multi-word searching, minimization of the number of clicks, and customization of the results table for mobile devices. An animated logo was also used to enhance the appearance, and the titles and descriptions of the results are displayed instead of their urls. Also, the number of pages are displayed and can be clicked to go to that certain page instead of just using "Previous" and "Next" buttons.

SECTION 2.1 - Functionality

The first priority in designing an enhanced search engine was to improve readability of the search results. In Lab 3, pageranked search results were displayed in the form of their url. In order to give the user more context into what the links contained, we proposed to replaced the url with the title of the page and include a short description of each webpage by extracting text from the page. By doing so, the user will have an idea of the content of the potential pages.

SECTION 2.2 - Appearance

We wanted to make the website as responsive as possible to possible window resolution changes so we opted to utilize Bootstrap's framework as its grid system allowed for the website to tailor to the dimensions of the window encompassing it. Notice as you decrease the size of the window, the search bar maintains its position at the center and the entire search bar shrinks to the width of the window.

To add some aesthetic touches to the webpage, the logo was animated with few actions so that it wouldn't be overly distracting to the user. Once every few seconds, the coffee bean logo will shrink in and make a face. Since the page was originally

made with a minimalistic design, the animated logo makes the webpage more interesting to look at.

SECTION 2.3 - Autocorrection

Every time a user searches something, the query is checked to see if the spelling is if correct. If any of the words are misspelled, those words are corrected, and the new list of words is used to search the database. This feature uses python's autocorrect library.

SECTION 2.4 - Multi-word Searching

The search engine can also handle multi-word searching. All words that are entered are used to search the database and get the combined list of pageranked urls.

SECTION 2.5 - Minimization Number of Clicks

The Roast search engine shows a search box on both the homepage and the results page. This make it easy for the user to make new searches.

SECTION 3 - Differences in Proposed and Completed Designs

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.

There were no differences between the proposed and completed design. We implemented all the features that we planned to implement when we started working on lab 4.

SECTION 4 - Testing Strategy

Explain your testing strategy during the development. Describe how you identify the corner cases.

The testing strategy used involved searching multiple combinations of words that exist and don't exist and seeing how the search engine responded to those queries. For the title/description of web pages feature, the program extracted from the <title> and <p> tags, and sometimes a webpage didn't have such tags on their page. We accounted for this by inserting the url as the title and no description in case such events occurred.

To test the auto correction and multi-word search features added in lab 4, the following queries (test cases) were used:

1. When "toront" is entered, the query is autocorrected to "toronto", and the results for "toronto" are displayed.
2. When "toronto" is entered, the results for "toronto" are displayed.
3. When "google" is entered, the results for "google" are displayed.
4. When "toront google" is entered, the query is autocorrected to "toronto google", and the results for both "toronto" and "google" are displayed.

SECTION 5 - Lessons Learned

Lessons learned from this project.

The main lessons learned from this project were that programming involves less of writing your own code and more of using code other people implemented. Especially in object-oriented programming, the theoretical knowledge learned in class is mainly applied to understand other people's code so that you can use them, build upon them and be able to detect bugs. Other lessons learned include frontend design, web server implementation, crawling and ranking pages, and deployment and management of web servers. These lessons are very valuable because they taught us both backend and frontend development which is very useful in the software industry.

SECTION 6 - Changes in the Future

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?

In terms of improving the readability of the search results, we would ideally like to find the surrounding text around a word that is being searched and display the local text surrounding that keyword. This will offer more relevant information in regards to what the user is looking for when searching for items. Also, we would improve the performance of the crawler by multithreading as well as implement auto completion to enhance user experience.

SECTION 7 - Course Material and Labs

How the material from the course help you with the project.

For most of the labs, the materials from the course did not affect how well the lab was done. Learning about concurrency for lab 3 when multithreading became more relevant helped to understand the underlying concepts, despite not actually having implemented the feature. Also, the exception and error handling lectures helped to understand how to handle exceptions that arose during AWS deployment.

SECTION 8 - Time Spent on Labs

How much time it takes for you to complete each lab outside the lab sections.

	Marinette (Mei Jia) Chen	Sai Harshita Tupili
Lab 1	15 hours	20 hours
Lab 2	15 hours	20 hours
Lab 3	3 hours	10 hours
Lab 4	7 hours	12 hours

SECTION 9 - Useful Parts of the Labs

Which part of the project you think is useful and you believe the labs should spend more time on it.
The Amazon Web Services portion is very useful, and it would be beneficial to spend more time learning how it works. Additionally, the database portion was

interesting, and it would have been nice to have had some more time to learn about SQL.

SECTION 10 - Future Labs

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.

All parts of this project were valuable experiences, and none of them should be removed from the labs as they increase students' knowledge of programming, web development, and databases.

SECTION 11 - Feedback and Recommendations

Other feedback or recommendations for the course.

A lot of time was spent trying to figure out the exact instructions for the labs as they tended to be vague, and sometimes bugs that already existed and were known in previous years were not addressed early on. For a better experience for next year students, these issues should be remedied.

SECTION 12 - Responsibilities of Each Member

Responsibilities of each member. If you believe that workload is distributed unequally in your group, you may describe the situation in this section.

The project workload was evenly distributed among the two group members. For lab 1, Marinette did the backend and Harshita did the frontend. For lab 2, Marinette did the frontend and Harshita did the backend and AWS. For lab 3, Marinette did the backend (pagerank and persistent storage) and Harshita did the frontend (error page and pagination. Harshita was responsible for the aws launch and termination scripts for lab 4 as well as the autocorrection (spell correct) feature and multi-word searching feature. Marinette got the titles and descriptions of the results and displayed them and customized the results table for mobile devices. Both of us worked on the report and the appearance of the search engine (HTML and CSS).