Marking Rubric for Mini-Project 2

**CMPUT 291/Fall 2025 – Project 2 Self Evaluation**

Repo being demoed:

On time or late:

Group member names along with CCID:

1. Rocky Chen CCID: minghe1

2.Guang Hua Liang CCID: guanghua

3.

For each section in the marking, test your project and mark it as 'pass', 'partial pass', or 'fail'.

**Testing and Demo Process:**

**Edit penalty rules:**

* 5-point deduction for 1-5 edits
* 10-point deduction for 6-10 edits
* 15-point deduction for 11-15 edits
* 20-point deduction for 16-20 edits.
* 20+ character edits → demo DISQUALIFIED (0/84 for demo marks)

Note: Edits are cumulative across files (adds+deletes+replacements). An edit can involve adding, removing, or replacing a character in your application code.

**At Demo Time, You May Choose:**

1. Run your code as is → graded strictly by our marking scheme.
2. Request minor edits (within limits) → penalty applied, **5 MINUTES MAX for changing code. Students should not spend more than 5 minutes debugging their code for fixes.**
3. Withdraw from demo → project visually assessed (0–15% mark range).

Once your demo begins, no further changes are allowed.

1. **Phase 1: Building a Document Store. [10 Marks]**

**Execution Time**

* Must complete within 3 minutes or score = 0 for this component

**Command Line Inputs**

* Accepts JSON filename as command-line input
* Accepts port number as command-line input

**Database and Collection Verification**

This is done via a mongosh session: mongosh --port <your-port-number>

* Database 291db successfully created (show dbs)
* Collection articles successfully created (show collections)
* The collection contains the correct data.

1. **Phase 2: Document Store Operations [74 Marks]**
   1. **Most Common Words by Media Type**

Media Type Selection:

* Blog returns top 5 words and matches expected output
* News returns top 5 words and matches expected output
* Case-insensitive input (e.g., NeWs) works for both news and blog
  + Note: Choice-based menus (e.g., 1 for news, 2 for blog) are acceptable.

Invalid Input Handling:

* The user may retry or return to the menu in case of wrong input
* An incorrect media type should not crash the program

Output Expectations:

* Exactly the top 5 common words (unless ties create more entries) with the correct word-count pair
* Sorted by frequency descending
  1. **Article Count: Difference Between News and Blogs**

**Proper handling of dates and no articles**

* Accepts formats such as “September 1, 2015” or clearly guides the user to the required format
* Shows “No articles were published on this day” for September 1, 2015

**Handle other input without error**

* The system does not exit/cause an error for inputs other than the required format

**Correctness of Output**

* Correct news count
* Correct blog count
* Correct difference computation between news and blogs
  1. **Top 5 News Sources (2015)**

**Feature Access**

* The user can access this section and see the News Sources
* The program does not terminate after accessing this section. System returns to the menu.

**Output**

* Displays up to 5 sources
* Correct source-count pairs (1 point each)
* Order is flexible
  1. **Five Most Recent Articles by Source**

**Handling of less than 5 articles**

* Shows only available articles (e.g., 1 result)

**Non-existent Source**

* Let the user know the source name is non-existent/no articles found
* The system continues running, no errors or termination.

**Correct Output**

* From a valid source. Displays at most 5 results.
* Includes article titles
* Includes publication date
  + If the date format differs from the required YYYY-MM-DD, partial credit applies.
* Sorted newest → oldest