SWE 432 Tech Challenge Portion of Final Exam

**15 December, 2020**

In this portion of your final exam, you will build a small web application. I am providing specifications of minimal required elements (***MRE***), and specifications of several additional optional required elements (***ORE***). You must submit your solution within 24 hours of the beginning of the scheduled final exam, which is 430pm, Wednesday December 16.

Each element will be graded on completion and on quality of the work. Completion (it works) will count for 2/3 of each MRE and ¾ of each ORE, and quality will count for the remainder. The quality evaluation will include usability, design, and quality of the code.

The program will count for 20 points. That, plus 10 points for the written portion, will match the 30% of the final grade allocated to the final in the syllabus.

**RESOURCES**

Each student must work individually—no collaboration, no discussion during the exam period, and no help with design or debugging. I will be available on zoom for questions during the scheduled exam period (430-720 on Tuesday), and through email thereafter. You may access any course materials, including books, the course website, slides, examples, piazza, etc. You may also use the internet, including any libraries or packages, as long as they are free or open source.

**TECHNOLOGY**

You may use any technologies we discussed in the course.

**SUBMISSION**

1. You may deploy your running software on github-heroku, AWS, or anywhere else. The important part is that I must be able to run your program through the URL you provide.
2. Submit your final exam by putting all source files in a folder named “*MasonID*-swe432,” and zip the contents into one zip file. You should replace “*MasonID*” with your Mason ID, that is, your email address. Thus, my submission would be named “offutt-swe432.zip”. Include the following files:
   1. Page three of this document, with all information filled out.
   2. All source files, include.java, .js, .css, and anything else that you use.

Please **do not** zip your entire gihub-heroku folder, I don’t need all of those files and don’t have room to store them.

1. Submit your zip file into my dropbox file request. You will be able to place your file, but not see it or change it after you submit. You may submit updates up until the final deadline.

Submit your file through this URL: **XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX**

**ACCOMMODATION**

If you need accommodation, either as specified by the DRC at the beginning of the semester, or due to the pandemic and online nature of the final, contact me by email individually. Online accommodations could include internet access problems or illness. If your internet crashes during the exam period, contact me as SOON AS POSSIBLE through email.

**SPECIFICATIONS**

**Minimal required elements (MRE)**

(**12 pts**.) Sorting inputs.

1. (**6 pts.**) Your web application will accept a list of strings, and return them in sorted order.
2. (**3 pts.**) The user can choose to sort in either ascending or descending order.
3. (**3 pts.**) The number of strings or numbers is not limited.

Note that you are to design the input and screens, design how the strings are displayed, design how the orderings are chosen, and design and write code to perform the concatenation. I am intentionally **not** giving you requirements about how the UI will look or behave.

**Optional required elements (OREs)**

If you submit a program with MREs only, your grade on the final exam will be 60%, which will count as a C grade (passing) for the exam. Implementing OREs can increase your score to as much as 100% (20 points), but no higher.

Choose **no more than two** of the following OREs:

1. (**4 pts**.) **Numeric**: The user can choose to sort in numeric order.
2. (**4 pts**.) **Sanitize**: Dangerous strings such as “<*script*>,” “<*javascript*>,” and “*onLoad*” are removed before sorting.
3. (**4 pts**.) **Unique**: If the list has two identical strings, only one will be returned in the sorted result. For example, if the submitted list is [*shirt*, *pants*, *shirt*, *shoes*], the web app will return the list [*pants*, *shirt*, shoes].
4. (**4 pts**.) **Forward**: Your backend uses the *dispatcher.forward*() method to forward the request from one servlet to another. Tell me the class and method name where the *forward*() method is used to help me find it.
5. (**4 pts**.) **Alternate sorts**: The user can choose to sort based on an additional quality, such as string length or character position. Be sure to make it clear what the alternate sort position is.

The maximum score on the final tech challenge exam is 20 points. I am not allowing extra credit on the final, so I will only consider **two** OREs. In your submission, you must tell me which OREs you attempted. If you attempt more, I will only grade the first two.

**Fill out and submit this page with your final exam**

Your name as on the roster: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Your GMU email address: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The URL where I can run your program: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(**IMPORTANT**: Test access to your program through the URL before submitting!)

List the technologies that you used in your program:

Which of the following OREs are you submitting for grading (do not check more than 2):

[ ] 1. Numeric

[ ] 2. Sanitize

[ ] 3. Unique

[ ] 4. Forward

[ ] 5. Alternate sort

“*I swear upon George Mason University’s Honor Code that I have not discussed this exam or shared any part of my program with anyone but the instructor, and will not share or discuss this exam or any part of my program with anyone else. I will destroy all copies, both paper and electronic.*”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_