Pranay Rajesh Gheewala
B00826923

CSCI 3901

Date: 18th January 2019

Assignment -1

Content Management System (CMS)

Overview

There are many non-technical users who don't know Hyper Text Markup Language, but wants to post formatted information on the web. So, a Content Management System helps users to post formatted information on the web.

The CMSs asks writer to put simple notation around their text to denote formatting. For example, one might use underscrores before and after text to indicate italics, such as _italicized_. The CMS would then translate that notation to the appropriate HTML coding; version 1.0 of HTML would have encoded our example as <i>italicized</i>.

I have created a program that translates the contents of the text file into a HTML 1.0 webpage and prints that HTML to the screen same as CMS. My approach for the program is very simple to read text from the input file line by line and convert the file to HTML format. I have used if - else loops to check whether it is paragraph or list, while loops to check the bold italic and underline tags, for loops for splitting the line into words, switch cases to close the opened tags. It will translate the text and display the output as summarized below:

Text annotation	Meaning	Tag
First text line starts “title:	“ Title of the document	<title>
Blank line	New paragraph	<p>
some text	Italicize “some text”	<i>
more text	Bold “more text”	<b>
! even more text	Bold next word “even”	<b>
%last text%	Underline “last text”	<u>
- List info	List item “list info”	<li>

Working

In this, program asks user to enter the path of the file they want to convert to HTML using Scanner which is an inbuild function of Java.

There are some input validations set in place to check the following :

- It checks whether the file exists or whether it is empty or not.
- It checks if the file does not exist or is empty, the user will get a message “File you entered does not exists or is empty.”
- It checks if the user selects other file than .txtfile

If the file is not empty the program will read the text file line by line by using Buffered Reader method which is in build function of Java.

If the first non-blank line starts with “title: “ (no quotes) then the text that follows in that line will be reported in a <title> tag within the <head> tag of the HTML. After the tags metadata will be printed. The tags </title> </head> will be printed after the metadata.

A blank line in the text will represent a new paragraph. Paragraphs are captured with the <p> tag. The paragraph will be closed before a new blank line is encountered. To keep the track of paragraph tags and list tags a int variable is used which will tell when to use paragraph or list tags.

Text that starts with a minus sign (-) as the first non-space character will be a list item that will be enclosed in a list <li> tag. The list items will start with a <ul> tag.
Every item of the list will be enclosed within <li> and </li> tags. At the end </ul> tag will be printed.

Text between asterisks (*) will be bolded using the <b> tag. To keep the track of bold annotations a flag is used so that it will open and close bold tag at correct positions. When the first bold annotation is encountered the flag will be set to true and whenever another bold annotation is encountered it will be set to false.
If a user forgets to close a bold annotation in text file it will be closed before the paragraph is closed.
Text between underscores (_) will be italicized using the <i> tag. To keep the track of italic annotations a flag is used so that it will open and close italic tag at correct positions. When the first italic annotation is encountered the flag will be set to true and whenever another italic annotation is encountered it will be set to false.
If a user forgets to close a italic annotation in text file it will be closed before the paragraph is closed.

Text between percentage signs (%) will be underlined using the <u> tag. To keep the track of underline annotations a flag is used so that it will open and close underline tag at correct positions. When the first underline annotation is

encountered the flag will be set to true and whenever another underline annotation is encountered it will be set to false.
If a user forgets to close a underline annotation in text file it will be closed before the paragraph is closed.

The appearance of an ! (alone or at the start of a word) will translate that the next word will be bolded using the <b> tag. My program will remove any extra ! marks used by the user by mistake.

There are no data structures used in my program. I have used simple if-else loop, switch case and flags to count the number of occurrences of annotation in the text to keep the count of the opening and closing tags. As the program read the lines whenever an annotation is encountered it will replace it into appropriate HTML tags.

Files

The program will ask user to enter a text file name if it is in the same folder as the project or user need to enter the complete path of the file if it is in the different folder which is required to be converted into HTML. The text file will contain text along with the annotations and the program will convert annotations to the HTML tags according to table described above. The output will be displayed on the screen.

Assumptions

I have assumed that:

- The input text does not contain any HTML tags or any special symbols that need to be translated for HTML.
- None of the special symbols used to annotate the text formatting appear in the text.
- List items are a single line
- List will not be inside paragraphs: they will have a blank line separating them from the surrounding text, unless starting or ending the body of the file

• There will be no formatting annotations inside the title line

Limitations

There are few limitations of my program

- It does not convert the annotations in the list
- It does not use any data structure so the efficiency of the program is decreased
- It does not close nested tags in proper sequence