EH203 - Week 2 - Project

# Specification

You are to develop an online quiz web application. The application will load the questions from a file, display them on a page, check the user’s answers, and give them a score.

The web app will read the quiz questions from a file, which can use any suitable format, though you may find the format below useful:

T This is the title of the quiz!

Q This is the first question?

W This is an incorrect answer

W This is another incorrect answer

A This the correct answer because it has a ‘A’ beside it

W This is yet another incorrect answer

Q This is the second question?

W ...

You provided with a very basic example of such a web app, which is accessible at:

http://intare:8080/eh203-quizgen

# Additional Requirements

1. When the quiz is submitted it should remember which answers the user gave and redisplay these
2. Detect when a user hasn’t given an answer to a question and then
   * Display a warning
   * Don’t display the answers or the score
3. Provide a button or link to restart the quiz
4. Use CSS to create an attractive looking website
5. Avoid using scriptlets - everything should be done in EL

# Extra Credit

1. Use a servlet initialization parameter to configure the path to the data file, instead of hardcoding it in the servlet class
2. When the quiz is completed, show the correct answer for each question

# Useful Snippets

## Reading a file

To read a text file from a web app, you should place the file in the *WebContent* directory (or a subdirectory, e.g. *WebContent/data/quiz.txt*), and put the following code in the init() method of the servlet:

ServletContext srvCtx = config.getServletContext();

InputStream in = srvCtx.getResourceAsStream(“data/quiz.txt”);

BufferedReader reader = new BufferedReader(new InputStreamReader(in));

## Getting the c:forEach index

Sometimes you need the iteration count inside a <c:forEach> tag - for example if you want to print the question number with each question. This can be done by specifying a varStatus variable:

<c:forEach items="${questions}" var=*"question"* varStatus=*"qstat"*>

<b>qstat.count}. ${question.question}</b>  
</c:forEach>

## Dynamic EL

Supposing you need to access a Bean property or Map element whose name is the result of an expression - the following is not valid EL:

${param[‘checkbox’ + 1]}

However you can create a new attribute using <c:set> and then use that in the EL expression, e.g.

<c:set var=”paramname” value=”${‘checkbox’ + 1}” />  
 ${param[paramname]}