

CSC309, Summer 2007, Assignment 3

In this assignment you will build a typical Web 2.0 application – an optimized browsing tool.

Notes:

- Your code should function correctly in the CDF environment. Part marks may be available for design aspects rather than just functional aspects of your code, and your code will not be debugged by the marker to make it work.
- It is not sufficient to just test your code; functionality alone does not guarantee maximum marks. Your solution should be professionally designed and support maintainability including future changes and extensions.
- Your solution files should never have public read, write, or execute permissions until after we stop accepting submissions.
- Provide a README file with clear instructions for all the steps to install and run your solution files in a CDF account.
- This assignment is marked out of 100, i.e., up to 10 bonus points are available.

Deliverables:

- Archive (tar and gzip) a folder containing the solutions you have created for Questions 1, 3b, 5, and 6.
- Archive (tar and gzip) the entire application folder deployed in the Tomcat server, containing all the files you have created for Questions 2, 3, 4, 5.
- Provide a README file as to how to install, configure and run your solution in your CDF account.

Question 1. [20 points]

Use Ajax to develop a "visual cache" navigation tool that places multiple browsing windows on a web page.

The visual cache tool displays a command area and several target areas on a page, in a horizontal split layout: command area at the top, and four target areas stacked up vertically over the rest of the page.

The command area has two text inputs ("address" and "target") and a button labelled "Go".

When the user presses the "Go" button, the tool downloads the document from the URL in the "address" window and inserts the downloaded content into the innerHTML of the target area corresponding to the number in the "target" input. If any of the "address" or "target" inputs are invalid when the user presses the "Go" button, the tool reports an error.

Question 2. [20 points]

Implement a Java class that downloads documents from the Web and delivers the documents in a compressed format. The class has a single method, called `getDocument`, which takes a single

parameter, a URL, and returns as result a zipped file containing the document downloaded from the given URL parameter.

Use `java.util.zip` for zipping documents. Ignore image files and zip only the main document present at the given URL.

Question 3. [20 points]

(a) Use `Java2WSDL` to generate a WSDL file from the Java class in Question 2.

(b) Use `WSDL2Java` to generate a client-side Java class for using a web service that complies to the WSDL file in Part (a).

Question 4. [20 points]

Use Apache Axis to deploy the Java class in Question 2 as a SOAP web service.

Question 5. [20 points]

Use the client-side Java class of Question 3 (b) to develop an Applet that displays the unzipped content of the file provided by the web service of Question 4. Use `java.util.zip` to perform the unzipping. The Applet takes the URL of the document to download from a user edited field within the Applet, and passes this URL as a SOAP message to the web service.

Question 6. [10 points]

Modify the Applet in Question 6 to take the URL parameter from the address field of Question 1 and to display the results within the selected target area of Question 1.

Good luck, and have fun!