**CMIS 375 Authentic Assessment**

**Final Programming Project**

Your final project will be a fully functional Perl/CGI program developed using a good, modular design providing an interesting and relevant service for Web users. You should work with your professor to make sure the service you have in mind will fulfill the requirements. You should design, create, and then execute the program. Be sure that your code executes correctly before submitting it. This project encourages you to use proper programming statements and reflect upon your overall design and approach.

**Project Objectives:**

* Create and process Internet forms through Common Gateway Interface programming
* Write and prepare written documentation for Perl programs

**Project requirements:**

* Your application context or topic must be approved by the professor prior to you starting your development. Be creative and have fun while you are determining your specific application.
* Performance: the application must run without errors and must do whatever is claimed in the documentation. The software should not be too slow, for example, it should not try to download huge files or graphics from the WWW while the user is waiting.
* User-friendliness: the application must be usable by a medium computer literate person. The web pages and links must contain enough information so that users know what the application does and what happens if they follow links or press buttons. The structure of the pages should be clear, consistent and contain no redundancies. The documentation should be clear and concise. The web pages and documentation should be spell and grammar-checked.
* Programming style: the source codes should be clear and should have appropriate comments.

**Projects submitted with evidence of plagiarism will be handled in accordance with UMUC Policy 150.25 – Academic Dishonesty and Plagiarism.**

Your project is due by midnight, EST, on the date posted in the class schedule. Your instructor’s policy on late projects applies to this project.

**Deliverables:**

* Perl/CGI application properly configured, deployed and running on the UMUC Nova server.
* URL to the application submitted to Assignment Folder in WebTycho.
* Well-written word document describing:

1. Your overall design
2. Your test plan, including test data and results
3. Your approach, lessons learned, design strengths, limitations and suggestions for future improvement and alternative approaches

**Format:**

**Documentation format and length:**

The word documentation describing and reflecting on your design and approach should be written using Microsoft Word and be no more than 5 pages in length and no less than 2 pages. The font size should be 12 point. The page margins should be 1 inch. The paragraphs should be with double line spacing. All figures, tables, equations and referenced should be properly labeled and formatted using APA style.

**Code format:**

* Header comment block,
* Appropriate comments within the code,
* Appropriate variable and function names, and
* Correct indentation.

**Grading:**

This activity is awarded 30% of the total grade in the course. In the grade book, the total number of points will be set to 100. The project elements are assessed as follows:

|  |  |
| --- | --- |
| **Attributes** | **Value** |
| Project Design | 20 points |
| Project Functionality | 40 points |
| Test Data | 20 points |
| Approach documentation | 15 points |
| Approach documentation grammar and spelling | 5 points |
| **Total** | **100 points** |

The following grading rubric will be used to assign points to your project.

**Grading Rubric:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Unacceptable** | **Acceptable** | **Distinguished** | **Points** |
| Program Design | (0-8) Points  - Use-case not documented or poorly documented  - Rarely employs proper use of parameters, variables and coding style  - Rarely employs correct & appropriate use of programming structures  - Poorly structured and inefficient algorithms | (9-15) Points  - Use-case partially documented  - Employs proper use of parameters, variables and coding style some of the time  - Employs correct & appropriate use of programming structures some of the time  - Efficient algorithms used some of the time | (15-20) Points  - Use-case completely documented  - Employs proper use of parameters, variables and coding style most of the time  - Employs correct & appropriate use of programming structures most of the time  - Efficient algorithms used most of the time |  |
| Functionality | (0-20) Points  - Application not configured correctly  - Application does not run as described  - Few requirements were fulfilled | (21-30) Points  - Application partially configured correctly  - Application partially runs as described  - Meets some requirements | (31-40) Points  - Application configured correctly  - Application runs as described  - Meets most requirements |  |
| Test Data | (0-10) Points - Little or no test data suite included | (11-17) Points  - Some test data and results included but some vital functionality not tested | (18-20) Points - Comprehensive test data suite included with results |  |
| Approach  documentation | (0-7) Points  - Offers unsupported or irrelevant opinions  - Does not consider alternatives  - Rarely identifies strengths and weaknesses of the experimental design and of the data | (8-12) Points  - Supports opinions with argument, evidence, or proof  - Considers alternatives but does not eliminate them  - Sometimes identifies strengths and weaknesses of the experimental design and of the data | (13-15) Points  - Conclusively argues/documents a point of view  - Considers alternatives and shows how and why they should/should not be accepted  - Identifies strengths and weaknesses of the experimental design and of the data |  |
| Approach  documentation  grammar/spelling | (0-2) Points  -Grammatical errors detract from the readability of the paper  -Sentence structure (run-ons, fragments) needs improvement and impedes readability  -Numerous spelling errors | (3-4) Points  -Few grammatical errors  -Sentence structure needs improvement  - Some spelling errors | (5) Points  - No grammatical errors  -Sentence structure and diction are effective and clear  - No obvious spelling errors |  |