|  |  |
| --- | --- |
| Date |  |
| Developer (Person whose code is being reviewed) |  |
| URL for project repository (optional) |  |
| URL for the hosted web app |  |
| Reviewer |  |

Instructions

The reviewer will complete this form for the beta version of a project done by another student. After filling out the “Beta” column and adding comments, the reviewer will upload this document to the Beta Forum.

The developer will revise the beta version of their project and fill out the “Production” column to reflect any changes they have made. The developer will submit this completed form along with the production version of their project.

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Beta** | **Release** |
| Does it compile and run without errors? (Yes, or No and list any issues below) |  |  |
|  | | |
| Do all the links, buttons or other UI elements work correctly? |  |  |
|  | | |
|  | | |
| Does the style conform to C# coding conventions? |  |  |
|  | | |
|  | | |
| Do the design and implementation conform to best practices? |  |  |
|  | | |
|  | | |
| Does the solution meet each of the requirements below? | | |
| Users should be able to enter data. |  |  |
| Users should be able to do some kind of searching. |  |  |
| The domain model should have at last 7 properties (total for all model classes). |  |  |
| There should be at least 4 web pages. |  |  |
| There should be some kind of navigation on each page. |  |  |
| The site has been published to a server and is running online. |  |  |
| There are unit tests. |  |  |
| Business logic and UI have been separated. |  |  |
|  | | |
| Comments | | |
|  | | |
|  | | |

**Appendix**

Aspects of coding style to check

* Is proper indentation used?
* Are the HTML elements and variables named descriptively?
* Have any unnecessary lines of code or files been removed?
* Are there explanatory comments in the code?
* Do variable names use camelCase?
* Are properties, methods and classes named using PascalCase (aka TitleCase)?
* Are constant names written using ALL\_CAPS?

Best practices in Object Oriented Programming

* Is the code DRY (no duplicated blocks of code)?
* Are named constants used instead of repeated literal constants?
* Is code that does computation or logical operations separated into its own class instead of being added to the code-behind?
* Are all instance variables private?
* Are local variables used instead of instance variables wherever possible?
* Does each method do just one thing (no “Swiss Armey” methods)?
* Are classes “loosely coupled” and “highly coherent”?