Stage 2 Project Marks

|  |  |  |  |
| --- | --- | --- | --- |
| Banner ID 1 |  | Banner ID 2 |  |

|  |  |  |
| --- | --- | --- |
| Core Functionality ( / 10) | | Mark |
| Nothing works | 0 |  |
| An item can be added to the app (e.g. a Contact, Expense item or whatever) | 1 yes | x/10 |
| A new item shows up in the on-screen list | 1 yes |
| All item attributes are properly assigned | 1 yes |
| An item can be selected for edit and/or delete | 1 yes |
| Items can be deleted *(note – this can be a difficult feature to implement)* | 2 yes |
| Items can be edited *(as can this one)* | 0/2 no |
| The current list of items is stored persistently (e.g. using localStorage) | 1 yes |
| On start-up, the list of items is retrieved from persistent storage | 1 yes |
|  | | |
| User-Interface Processing ( / 10) | | Mark |
| There is no way to enter item data | 0 | x/10 |
| Item data can be entered | 1 yes |
| There is a form for entering/editing item data | 1 yes |
| Elements of item data have suitable input controls (e.g. checkboxes, selections etc.). Some care is needed here, since some data types have varying support in different browsers – e.g. Chrome and Opera provide better support for date input – give students the benefit of the doubt if the app is better supported in one or other browser | 1 yes |
| The user interface is styled to appear attractive | 1 yes |
| At least some item data is validated (either by regex expressions or Javascript code) | 1 yes |
| All item data is validated | 0/1 no |
| User input events trigger an update of the list of items (e.g. pressing an Add or Cancel button) | 1 yes |
| Items in the list display can be selected on-screen (e.g. by using a check-box or clicking on the item in a table row) | 1 yes |
| There is some mechanism for separating user-input from the item list view (e.g. separate HTML documents, hiding and showing <div> elements) | 1 no |
| The user is given appropriate cues (e.g. warning alerts, input control styling) for error conditions on input | 1 yes |
|  |  |
|  | | |
| **Project Features** ( / 10) | | **Mark** |
| There are features in this project that go beyond the base-criteria that are worthy of merit: for example, good separation of application components (e.g. MVC), good testing procedures, testing across various browsers, CSS styles and/or animations used well, worthwhile additional functionality or features. | Up to 10 marks (with some explanatory comments) | x / 10 |
| **Total Mark ( / 30)** | | x/30 |
| **Overall comments on Implementation stage** | | |
|  | | |

Stage 2 Project Marks

|  |  |  |  |
| --- | --- | --- | --- |
| Banner ID 1 | B00044649 | Banner ID 2 |  |

|  |  |  |
| --- | --- | --- |
| Core Functionality ( / 10) | | Mark |
| Nothing works | 0 |  |
| An item can be added to the app (e.g. a Contact, Expense item or whatever) | 1 yes | 8/10 |
| A new item shows up in the on-screen list | 1 yes |
| All item attributes are properly assigned | 1 yes |
| An item can be selected for edit and/or delete | 1 yes |
| Items can be deleted *(note – this can be a difficult feature to implement)* | 2 yes |
| Items can be edited *(as can this one)* | 0/2 no |
| The current list of items is stored persistently (e.g. using localStorage) | 1 yes |
| On start-up, the list of items is retrieved from persistent storage | 1 yes |
|  | | |
| User-Interface Processing ( / 10) | | Mark |
| There is no way to enter item data | 0 | 10/10 |
| Item data can be entered | 1 yes |
| There is a form for entering/editing item data | 1 yes |
| Elements of item data have suitable input controls (e.g. checkboxes, selections etc.). Some care is needed here, since some data types have varying support in different browsers – e.g. Chrome and Opera provide better support for date input – give students the benefit of the doubt if the app is better supported in one or other browser | 1 yes |
| The user interface is styled to appear attractive | 1 yes |
| At least some item data is validated (either by regex expressions or Javascript code) | 1 yes |
| All item data is validated | 1 yes |
| User input events trigger an update of the list of items (e.g. pressing an Add or Cancel button) | 1 yes |
| Items in the list display can be selected on-screen (e.g. by using a check-box or clicking on the item in a table row) | 1 yes |
| There is some mechanism for separating user-input from the item list view (e.g. separate HTML documents, hiding and showing <div> elements) | 0/1 no |
| The user is given appropriate cues (e.g. warning alerts, input control styling) for error conditions on input | 1 yes (subtle though) |
|  |  |
|  | | |
| **Project Features** ( / 10) | | **Mark** |
| There are features in this project that go beyond the base-criteria that are worthy of merit: for example, good separation of application components (e.g. MVC), good testing procedures, testing across various browsers, CSS styles and/or animations used well, worthwhile additional functionality or features. | Up to 10 marks (with some explanatory comments) | 7 / 10 |
| **Total Mark ( / 30)** | | 25/30 |
| **Overall comments on Implementation stage** | | |
| This works really well, does everything it should and has a clean and clear appearance, based on well-designed CSS styles. **Very minor** criticisms: the on-screen available balance is not visible in a default web-view on my PC because the box you put it in is not big enough (this was easily fixed by changing view settings, but it would have been better if the program defaulted to this); it is not obvious that an invalid value entered has been rejected – you code the validation well enough (the value entered disappears off-screen) but some overt warning – an alert box or a red field – would have been better. In every other respect, this does exactly what it needs to. Good work | | |

Stage 2 Project Marks

|  |  |  |  |
| --- | --- | --- | --- |
| Banner ID 1 | B00136307 | Banner ID 2 | B00241867 |

|  |  |  |
| --- | --- | --- |
| Core Functionality ( / 10) | | Mark |
| Nothing works | 0 |  |
| An item can be added to the app (e.g. a Contact, Expense item or whatever) | 1 yes | 6/10 |
| A new item shows up in the on-screen list | 1 yes |
| All item attributes are properly assigned | 1 yes |
| An item can be selected for edit and/or delete | 1 yes |
| Items can be deleted *(note – this can be a difficult feature to implement)* | 2 yes |
| Items can be edited *(as can this one)* | 0/2 no |
| The current list of items is stored persistently (e.g. using localStorage) | 0/1 no |
| On start-up, the list of items is retrieved from persistent storage | 0/1 no |
|  | | |
| User-Interface Processing ( / 10) | | Mark |
| There is no way to enter item data | 0 | 7/10 |
| Item data can be entered | 1 yes |
| There is a form for entering/editing item data | 1 yes |
| Elements of item data have suitable input controls (e.g. checkboxes, selections etc.). Some care is needed here, since some data types have varying support in different browsers – e.g. Chrome and Opera provide better support for date input – give students the benefit of the doubt if the app is better supported in one or other browser | 1 yes |
| The user interface is styled to appear attractive | 1 yes |
| At least some item data is validated (either by regex expressions or Javascript code) | 1 yes |
| All item data is validated | 0/1 no |
| User input events trigger an update of the list of items (e.g. pressing an Add or Cancel button) | 1 yes |
| Items in the list display can be selected on-screen (e.g. by using a check-box or clicking on the item in a table row) | 1 yes |
| There is some mechanism for separating user-input from the item list view (e.g. separate HTML documents, hiding and showing <div> elements) | 0/1 no |
| The user is given appropriate cues (e.g. warning alerts, input control styling) for error conditions on input | 0/1 no |
|  |  |
|  | | |
| **Project Features** ( / 10) | | **Mark** |
| There are features in this project that go beyond the base-criteria that are worthy of merit: for example, good separation of application components (e.g. MVC), good testing procedures, testing across various browsers, CSS styles and/or animations used well, worthwhile additional functionality or features. | Up to 10 marks (with some explanatory comments) | 6 / 10 |
| **Total Mark ( / 30)** | | 19/30 |
| **Overall comments on Implementation stage** | | |
| This looks nice and does much of what is needed, with two obvious omissions – data is not saved locally (there is no reference to localStorage in the code) and no data is validated (you can enter rubbish in the phone numbers and email fields). You’ve made good use of CSS to style the user interface and put a gloss on the final project (important for attracting users). Your documentation is short but appropriate. | | |