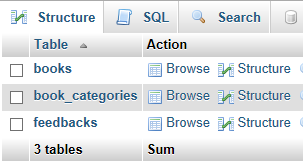
Worksheet 06

**Improve the Look**

# Setup

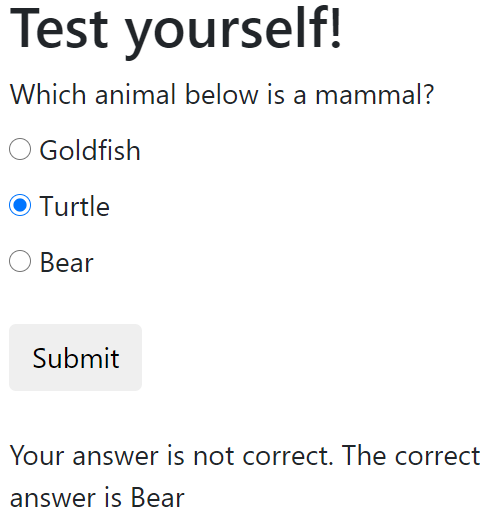
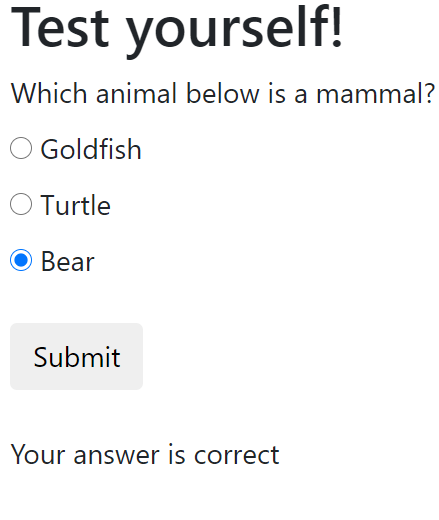
1. Download *C273\_L06ImproveTheLook.zip* from LEO. Save and extract it to C:\xampp\htdocs\C273\_L06ImproveTheLook\.
2. Start NetBeans. Create a PHP Project with Existing Source that points to above folder.
3. Start the XAMPP Control Panel, and **start both the Apache and MySQL component**.
4. Type <http://localhost/phpmyadmin> on your browser to open PHPMyAdmin.
5. Create database **c273\_p06**. Import the file **c273\_p06.sql** into that database. You should have the table **books, book\_categories** and **feedbacks** under the database.



# jQuery Recap

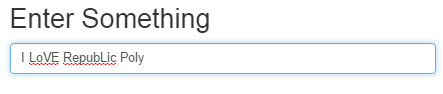
**Exercise 1**

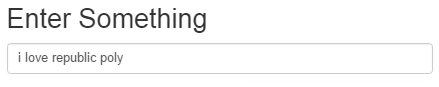
Open exercise1.html, it contains a multiple-choice quiz. It has 1 question with 3 options for the user to select one answer. Write the code to achieve the following. After the user selects an answer, alert him or her if he or she is wrong, and show the user the correct answer in a separate field.

**Exercise 2**

Open exercise2.html, it contains a textbox. After the user enters text, all the letters will be converted to lowercase as soon as he or she clicks anywhere else in the form. (Use the change event handler and toLowerCase() method)





# jQuery UI: Autocomplete, Date Picker, Slider and Star rating

Enables users to quickly find and select from a pre-populated list of values as they type, leveraging searching and filtering. See <https://jqueryui.com/autocomplete/>

Select a date from a popup or inline calendar. See <https://jqueryui.com/datepicker/>

1. **TODO**: Open exercise3.html. Link the following css and js files to this file.

|  |
| --- |
| <link rel="stylesheet" href="css/jquery-ui.min.css">  <script src="js/jquery-3.5.1.min.js" type="text/javascript"></script>  <script src="js/jquery-ui.min.js" type="text/javascript"></script> |

jquery-ui.min.js uses jquery-3.5.1.min.js, so it is important that the latter is above the earlier. This is called plugin dependency.

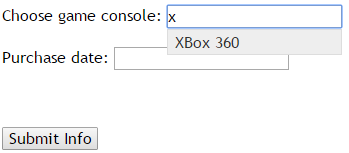
1. **TODO**: Type the following jQuery code within the <script> tags to make use of the autocomplete feature for the game console text field.

|  |  |
| --- | --- |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10**  **11**  **12**  **13** | $(document).ready(function() {  var availableConsoles = [  "PC",  "Playstation3",  "PSP",  "Nintendo Wii",  "XBox 360"  ];  $("#id\_consoles").autocomplete({  source: availableConsoles  });  }); |

Line 3 - 9 – use array **availableConsoles** to save all possible values for the autocomplete feature. **Autocomplete is a jQuery UI**

Line 10 - 12 – apply the autocomplete feature to the HTML elements with the id **id\_consoles.** The values come from the elements of the array **availableConsoles**.

1. Run exercise3.html and observe the results when you type on the game console field.



1. **TODO**: Type the following jQuery code within $(document).ready(function() { }); to make use of the datepicker feature for the Last Visit text field.

|  |
| --- |
| $("#id\_date").datepicker({minDate: 2, maxDate: "+1M +10D"}); |

Apply the datepicker feature to the HTML elements with id **id\_date. Minimum date** is 2 days from now. **Maximum date** is 1 month and 10 days from now.

**Resources**: <http://api.jqueryui.com/datepicker/#option-minDate>

<http://api.jqueryui.com/datepicker/#option-maxDate>

1. **TODO**: Type the following jQuery code within $(document).ready(function() { }); to make use of the slider feature for the Quantity field.

|  |  |
| --- | --- |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10** | $("#slider").slider({  value: 0,  min: 0,  max: 20,  step: 1,  slide: function (event, ui) {  $("#num").val(ui.value);  }  });  $("#num").val($("#slider").slider("value")); |

Line 1-5 – apply the slider feature to the HTML element with the id **slider.**

* **value:** The default value
* **min:** minimum value of slider
* **max:** maximum value of slider
* **step:** incremental value

Line 7 – selected value will be assign to the input field with id **num**.

Line 10 – on page load, the default slider value will be assign to the input field with id **num**.

1. Run exercise3.html and observe the results when you select a value on the slider.



**Resources**: <http://api.jqueryui.com/slider/>

1. **TODO**: Link the **jquery.raty.min.js** in the js/ folder to exercise3.html in order to make use of the star rating feature.

Type the following jQuery code within $(document).ready(function() { }); to make use of the star rating feature.

Resource: <https://github.com/wbotelhos/raty>

|  |  |
| --- | --- |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8** | $.fn.raty.defaults.path = 'js/img';  $('#target-text-demo').raty({  cancel: false,  scoreName: 'rate\_us',  number: 3, //change to 5  score: 2  }); |

Line 1 – set the default path for the star images used for the star rating feature.

Line 3 - 8 – apply star rating feature (**raty**) to HTML elements with id **target-text-demo**.

* **cancel**: the values set cannot be cancelled
* **scoreName**: the assigned value will be saved to an HTML input with the name "rate\_us"
* **number**: there are three possible values: 1, 2, or 3
* **score**: the default value is 2

1. **TODO:** Modify **example3.html:**

* Change the minimum date to 1 month before now**.**
* Make maximum rating to be 5 stars

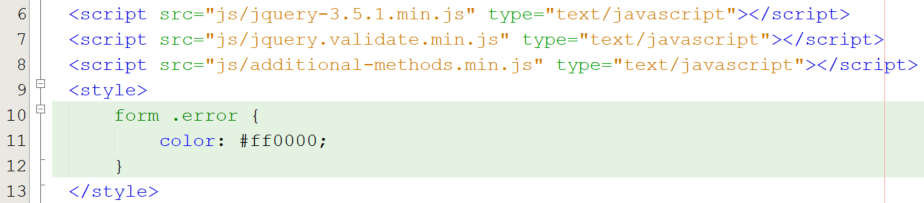
# Form validation

This jQuery plugin makes simple clientside form validation easy, whilst still offering plenty of customization options: <https://jqueryvalidation.org/>

Resource: <http://javascript-coder.com/form-validation/jquery-form-validation-guide.phtml>

1. Using NetBeans, open **exercise4.html**. Run the page and observe it.

The following is a portion of code from that page:



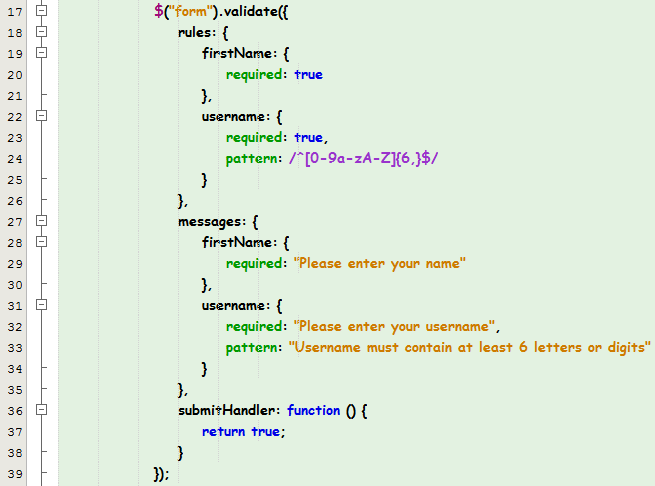
Line 7 - <script> tag to reference the jQuery validation script. It has a dependency on the jQuery library, therefore it is placed below it.

Line 8 - <script> tag to reference the additional methods stored in additional-methods.min.js. <https://github.com/jquery-validation/jquery-validation/tree/master/src/additional>

Line 9-13 – style the class **color** to show the error messages in **red**.

In simple form, the format is like this:  
$('#formid').validate( { rules:{ }, messages:{ } } );

See example below:



Line 17 – use the **validate** function to apply validations on the HTML form with the selector **form.**

Line 19-21 – set rules for the input name **firstName**.

Line 22 – 25 – set rules for username: **required** and using pattern to specify the regular expression required (at least 6 letters or digits). Make sure that you use /^….$/

Lines 28-34 – customize the error messages for each error

**TODO:** Modify **exercise4.html:**

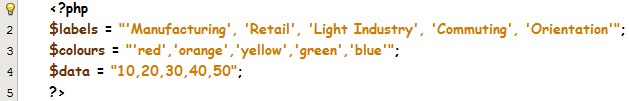
* Change validation for full name so that minimum length is 5

# Pie and Donut Charts

Chart.js is a simple, clean and engaging HTML5 based JavaScript charts. See <http://www.chartjs.org>

1. In NetBeans, open **exercise5.php**. Run the page. Three charts should be shown.

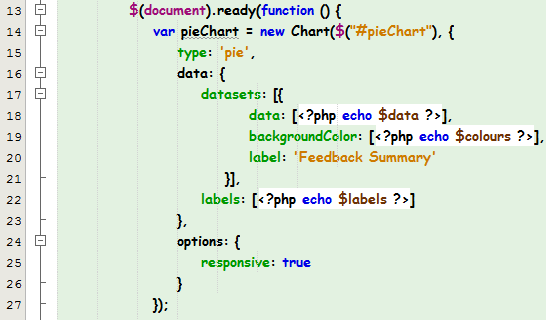
The one that resembles the solution is the first chart. Go to the code and find:



**Line 2-5**: the PHP variable **labels** which contains all labels for the pie chart, **colours** and **data** variables which contains the background colours and data values for each of the corresponding label respectively.

jQuery library and Chart.bundle.min.js is included in this php file

|  |
| --- |
| <script src="js/jquery-3.5.1.min.js" type="text/javascript"></script>  <script src="js/Chart.bundle.min.js" type="text/javascript"></script> |



**Line 14:** Instantiate a new chart which binds to the HTML element id **pieChart**

**Line 15:** There are many types line, bar, radar, pie, doughnut, etc.

**Line 18:** datasets containing the data points which in this case is stored in the PHP variable **data**

**Line 19:** The fill color of the arcs in the dataset. See<http://www.chartjs.org/docs/latest/general/colors.html#colors>

**Line 22:** The label for the dataset which appears in the legend and tooltip

The code to hold the chart:



Copy the sections of code as above to **summary.php**, you need to continue with that page:

* Run the page on the browser, and make sure the pie chart is shown
* Populate $labels, $colours and $data with the appropriate labels, fill colours and data points retrieved from $total\_ratings. See label mapping below

|  |  |
| --- | --- |
| Label | Rating number |
| Need Improvement | 1 |
| Okay | 2 |
| Good | 3 |
| Very Good | 4 |
| Excellent | 5 |

# Complete the Problem Statement

For the page **bookList.php, you need to:**

* Allow visitors to filter books by category.

**Hint**: Upon selecting a category from the dropdown list (use change event)

* If “all categories” (value: 0) is selected, show all rows (<tr>)
* Otherwise, hide all rows first, then show the table header and the row with that category

For the page **feedback.php, you need to:**

* Add an autocomplete feature to the area input to indicate areas that contain the letter entered by the user
* Add a datepicker / calendar feature to the last visit input
* Change the user rating input into a star rating format.
* Add form validation for the input values
  + visitor name: compulsory and minimum of 2 characters
  + area: compulsory
  + last visit: compulsory and follows format DD/MM/YYYY
  + comments: compulsory and maximum number of characters = 200 (**Hint**: use **maxlength** attribute)

For the page **summary.php, you need to:**

* display the rating distribution in a pie chart.

# Challenge Yourself

1. In the **feedback.php**, validate the name to have at least 2 characters long and consist of alphabetical, number, dot or underscore.
2. In the **feedback.php**, add one input box for user to enter their email address. The email address has to be validated. (you will need to modify the *visitor\_comments* database table as well to capture the email address).
3. In the **feedback.php**, use the bootstrap datetimepicker instead of the jQuery UI datepicker. Remember to change the validation to validate the new datetimepicker format.
4. Change the dropdown in the **bookList.php**, so that its values are retrieved dynamically from the database table **book\_categories.**