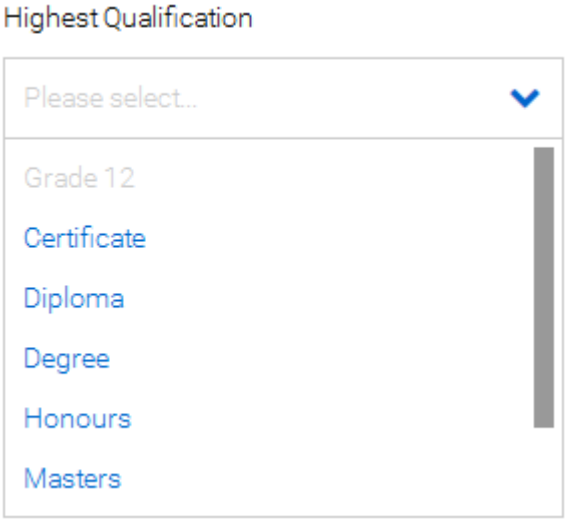
Technical Test

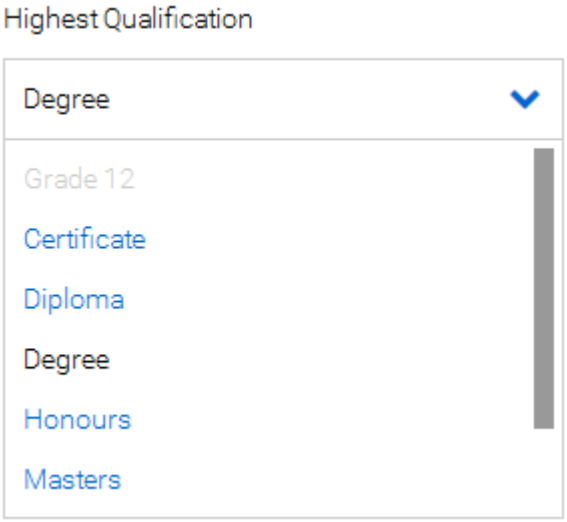
This widget is a custom single select dropdown
The items to appear in the dropdown must be configurable via external resource file
The selected value must be submitted to a restful endpoint that returns an associated value based on the input.
The returned response must be printed below the dropdown

Widget Behavior

- Options are hidden until the user clicks on the Single Select Field.
- When nothing is selected, placeholder text is displayed.



- After clicking on the Single Select Field, a dropdown panel showing a list of options becomes visible.
- Disabled options are grey and not clickable.
- Available options are blue.
- A custom scrollbar is displayed if there are more than 6 options.



- When the user selects an option, it changes to black to show that it is selected. The placeholder text is replaced with the selected text and the dropdown panel closes.
- The user may only choose one option.
- If an option was selected already and the user clicks on a new option, the previous selected option changes back to blue and the text in the field updates to the new option's text.

- When an option is selected, it is displayed in black text within the field and the value is stored in a hidden select element.

Widget Dimensions

Any width is acceptable

Widget Styling

Field

- Background Colour: #fff
- Border Colour: #ccc
- Border Width: 1px
- Height: 40px
- Width: 280px
- Label Colour: #000
- Placeholder Text Colour: #ccc
- Icon Colour: #06c
- Font: Helvetica, Arial, Serif

Dropdown Panel

- Background Colour: #fff
- Width: 280px
- Height: 190px
- Border Colour: #ccc
- Border Width: 1px

Dropdown Panel Scroller

- Colour: #999
- Width: 10px, centered within 20px clickable area
- Height: proportional

Dropdown Options

- Text Size: 14px in 20px line height
- Selectable Option Text Colour: #06c
- Disabled Option Text Colour: #ccc
- Selected Option Text Colour: #000

Instructions

1. Populate the dropdown list items using an external resource file (see “Restful Service Instructions” section below)
2. Code your own CSS3, HTML5 and JavaScript and PHP.
3. NO Bootstrap or similar frameworks allowed!
4. You may only use pure JavaScript, jQuery and PHP. No other frameworks allowed!
5. You may use a jQuery plugin for a custom scrollbar if you like.
6. Your example should at least work on the newest versions of Chrome, Firefox and Android (so the scrollbar etc. should also work on touch).
7. Upon selection and subsequent form submission, the widget needs to invoke a REST service with the selected value and the original list of items and subsequently display the calculated result on screen. (see “Restful Service Instructions” section below)

Restful Service Instructions

The initial options in the dropdown to be added to the resource file are

1. Some Schooling (disabled)
2. Grade 9
3. Grade 10
4. Grade 11
5. Grade 12 / Matric (disabled)
6. Certificate
7. Diploma
8. Degree
9. Honours
10. Professional Qualification
11. Masters
12. Doctorate

(note, the order of the list is significant and carries a weighting based on position [first in the last carries lowest weight and last in the list carries highest weight]:

1. Add a submit button to the widget that will submit the chosen option as well as the full list of items in the dropdown
2. Create an API (exposed via REST) in PHP that accepts a JSON input string containing the values as per point 1 above
3. Once submitted the API function needs to determine the response based on the chosen criteria:
    - The number of original items in the list
    - The position (weight) of the chosen item
4. The return value should be determined as follows:
        i. If the chosen position (weighting) is within the top 25% of submitted qualifications return “Your selected qualification of “*selected option* “ is inadequate for this position”

ii. If the chosen position (weighting) is within the top 26%-60% of available positions return “Your selected qualification of “*selected option* “ is dependent on additional information”
iii. If the chosen position (weighting) is within the top 61%-100% of available positions return “Your selected qualification of “*selected option* “ is highly desirable”

5. Furthermore, an associative array of hard coded values must be stored in the API implementation class and concatenated to the final response.
    - Associated array values to be concatenated:
      Some Schooling (disabled) -> “Next step: Inadequate Experience”
      Grade 9 -> “Next step: Inadequate Experience”
      Grade 10 -> “Next step: Inadequate Experience”
      Grade 11 -> “Next step: Dependent on Results”
      Grade 12 / Matric (disabled) -> “Dependent on Results”
      Certificate -> “Next step: Establish Level”
      Diploma -> “Next step: Longlist”
      Degree -> “Next step: Shortlist Candidate”
      Honours -> “Next step: Shortlist Candidate”
      Professional Qualification -> “Next step: Shortlist Candidate”
      Masters -> “Next step: Shortlist Candidate”
      Doctorate -> “Next step: Shortlist Candidate”

      Ex. If the user chose option 7 from the original list, the result should display:
      *Your selected qualification of **Diploma** is dependent on additional information.*
      Next Step: Longlist

      In the instance where the resource file is changed and alternate qualifications are passed to the API the Next step section should be omitted from the result

6. All code must be object oriented
7. Provide test classes if and where applicable
8. Provide all the relevant files along with a readme file providing instructions on how to deploy and run the widget on a web server of your choice.