Constructed Response & Drop-down Item Type Tech Spec

[I. Item Type Overview 1](#_Toc7053609)

[A. JSON Structure 1](#_Toc2119231104)

[B. Other Database Fields Used 4](#_Toc1526647723)

[C. Scoring Types used 4](#_Toc1658078740)

[D. Item Validation Rules 4](#_Toc1115720299)

[E. Third Party Libraries Used 5](#_Toc16354548)

[II. Item Content and Response View 5](#_Toc1838972338)

[III. Technical Updates 11](#_Toc869903336)

[A. Outstanding Defects 11](#_Toc98845174)

[B. Business Flow 12](#_Toc419974262)

[C. Implementation 13](#_Toc374722905)

[D. Software Updates Needed to Align with Project Standards 15](#_Toc2006325650)

[E. How to consume 15](#_Toc105858058)

[F. Click History 16](#_Toc861599324)

[IV. Test Scenarios 17](#_Toc178974331)

[V. Final Effort Estimate 21](#_Toc1778813008)

[VI. Questions 21](#_Toc1572491758)

# Item Type Overview

## JSON Structure

* + Constructed Response item JSON structure below.

{

"uuid": "0b870c6-d7d3-24be-bbc-7fc4bccbb143",

"stemContent": "<p id=\"cb-277863-5-71\"><span id=\"cb-884706-2-90\" style=\"color:#000000;\">Click inside the answer box. Type an answer. Watch the video to see how to answer.&nbsp;</span></p>\n\n<hr id=\"cb-519005-0-82\" />\n<p id=\"cb-266840-6-18\"><span id=\"cb-181068-9-45\" style=\"color:#000000;\">How many letters are in the word <u>apple</u>? Enter your answer.</span></p>\n\n<p id=\"cb-843875-10-6\"> <span class=\"constructedResponse-placeholder placeholder\" data=\"5\" id=\"cb-759079-9-91\" style=\"width: 1ch; text-align: center; color:black; background-color:greenyellow; border: 1px solid grey; border-radius: 5px; \">5</span><span id=\"cb-55837-6-6\" style=\"color:#000000;\">&nbsp;letters</span></p>\n\n<p id=\"cb-17222-8-38\"><span id=\"cb-138450-9-39\" style=\"color:#000000;\">Table with CR boxes</span></p>\n\n<table border=\"1\" cellpadding=\"5\" cellspacing=\"0\" id=\"cb-241849-5-6\">\n\t<tbody id=\"cb-632499-1-55\">\n\t\t<tr id=\"cb-537534-3-88\">\n\t\t\t<td id=\"cb-571745-7-78\" style=\"text-align: center;\"><strong>Word</strong></td>\n\t\t\t<td id=\"cb-646454-6-40\" style=\"text-align: center;\"><strong>Number of Letters</strong></td>\n\t\t</tr>\n\t\t<tr id=\"cb-722786-1-69\">\n\t\t\t<td id=\"cb-269426-0-43\">Dog</td>\n\t\t\t<td id=\"cb-318684-10-83\"> <span class=\"constructedResponse-placeholder placeholder\" data=\"3\" id=\"cb-25578-4-86\" style=\"width: 1ch; text-align: center; color:black; background-color:greenyellow; border: 1px solid grey; border-radius: 5px; \">3</span></td>\n\t\t</tr>\n\t\t<tr id=\"cb-100647-1-43\">\n\t\t\t<td id=\"cb-731453-7-82\">Plants</td>\n\t\t\t<td id=\"cb-828644-6-56\">  <span class=\"constructedResponse-placeholder placeholder\" data=\"Six\" id=\"cb-61930-10-61\" style=\"width: 1ch; text-align: center; color:black; background-color:greenyellow; border: 1px solid grey; border-radius: 5px; \">Six</span></td>\n\t\t</tr>\n\t</tbody>\n</table>\n\n<p id=\"cb-485345-3-44\">&nbsp;</p>\n",

"itemTypeCode": "CR",

"minItemWidth": 0,

"minItemHeight": 0,

"optionList": [

{

"id": "cke-ef5a3bd4ecb0896b39c1d5c332affcd15",

"index": 0,

"maxFieldSize": 15,

"responseName": "Response A",

"responseType": "AlphaNumeric",

"caseSensitive": true,

"maxCharacters": 15

},

{

"id": "cke-e30685ab10bf2ffa3d3b122fb0d5c0d5e",

"index": 1,

"maxFieldSize": 15,

"responseName": "Response B",

"responseType": "AlphaNumeric",

"caseSensitive": false,

"maxCharacters": 15

}

]

"correctResponse": [

{

"id": "cke-ef5a3bd4ecb0896b39c1d5c332affcd15",

"value": "Good",

"acceptableValues": [

{

"id":"av1",

"value":"good"

},

{

"id":"av2",

"value":"Nice"

}

]

},

{

"id": "cke-e30685ab10bf2ffa3d3b122fb0d5c0d5e",

"value": "Great",

"acceptableValues": [

{

"id":"av3",

"value":"great"

},

{

"id":"av4",

"value":"Grreatttt"

},

{

"id":"av5",

"value":"Awesome"

}

]

}

],

"rationale": {

"optionList": [

{

"id": "cb-176658-8-76",

"rationaleText": "rationale1"

},

{

"id": "cb-25578-4-86",

"rationaleText": "rationale2"

},

{

"id": " cb-759079-9-91",

"rationaleText": "rationale3"

}

]

}

}

* For Drop-down item, most of the fields are similar with constructed response except optionList, **correct response and item type code**

{

"optionList": [

{

"id": "cke-ead42409ce41a08d5f42e21336e2b7g82",

"index": 0,

"optionTags": [

"Select",

"want",

"choose",

"like",

""

],

"defaultValue": "Select",

"responseName": "Response a",

},

{

"id": "cke-ead42409ce41a08d5f42e21336e2b7d51",

"index": 1,

"optionTags": [

"Select",

"That",

"This",

"Those"

],

"defaultValue": "Select",

"responseName": "Response b",

}

]

"correctResponse": [

{

"id": "cke-ead42409ce41a08d5f42e21336e2b7g82",

"value": "choose",

},

{

"id": "cke-ead42409ce41a08d5f42e21336e2b7d51",

"value": "This"

}

],

    "itemTypeCode": "DD",

}

## Other Database Fields Used

* + We are not using any other specific database fields for this item type other than the standard field for an item.

## Scoring Types used

* + - The Scoring types for Constructed Response are
* No Scoring needed
* Correct Only
* Partial Credit
  + - The Scoring types for Drop-down are
      * No Scoring needed
      * Correct Only
      * Subtractive Model
      * Collapsed Quota
      * Partial Quota
      * Partial Credit

## Item Validation Rules

These are the validations used in this item

* CORRECT\_ANSWER\_REQUIRED\_LEN- The correct Answer is required.
* SCORE\_METHOD\_REQUIRED- Scoring Method is required.
* MAX\_SCORE\_REQUIRED - Maximum Score is required and it cannot be zero.
* SCORE\_EXCEED - Total score cannot be greater than maximum score.
* SCORE\_DECEED - Total score cannot be less than maximum score.
* SCORE\_NEGATIVE - Score must be between zero and the maximum.
* CONTENT\_CODE\_SUGGESTED – At least one primary content code should be selected.
* KEY\_HAS\_SPACES\_OR\_SPL\_CHARS - Key has spaces and/or special characters
* CONFIRM\_ANSWER\_OPTION - Please confirm answer option selection
* RESPONSE\_TYPE\_REQUIRED - Response Type (alphanumeric or numeric) is required
* COLLAPSED\_QUOTA\_SCORING\_TABLE- Collapsed partial quota scoring table has been generated.
* SET\_SCORE\_0\_ON\_ALL\_CHOICES- Score Zero if All Response Options Selected.
* SUM\_TO\_MAX\_SCORE- For subtractive scoring all options must sum to the max score.
* MAX\_EXCEED- For collapsed quota, max must be less than number of responses.
* MAX\_LESSTHAN\_MIN - Maximum cannot be less than the minimum.
* MINMAX\_NEGATIVE - Minimum must be between 1 and max score.
* FIRST\_MIN- For collapsed quota, first minimum should be 1.

**Common Validations**

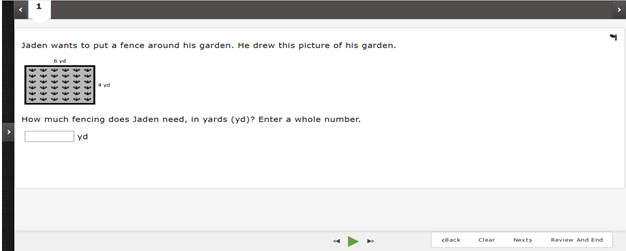
* Attributes and Metadata accordion Content Area, Grade, Framework, Taxonomy, Cognitive category, testing program, Usage and Estimate of difficulty validations are included for this item.

## Third Party Libraries Used

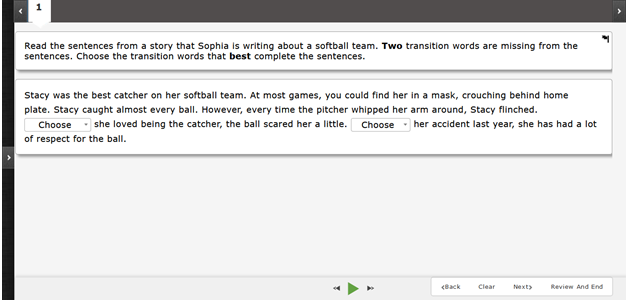
* Using CkEditor 5 custom plugin.
  + - * We have customized a plugin named as Item Response using Ckeditor 5.

# Item Content and Response View

* CB 1 SP preview (Constructed Response item)

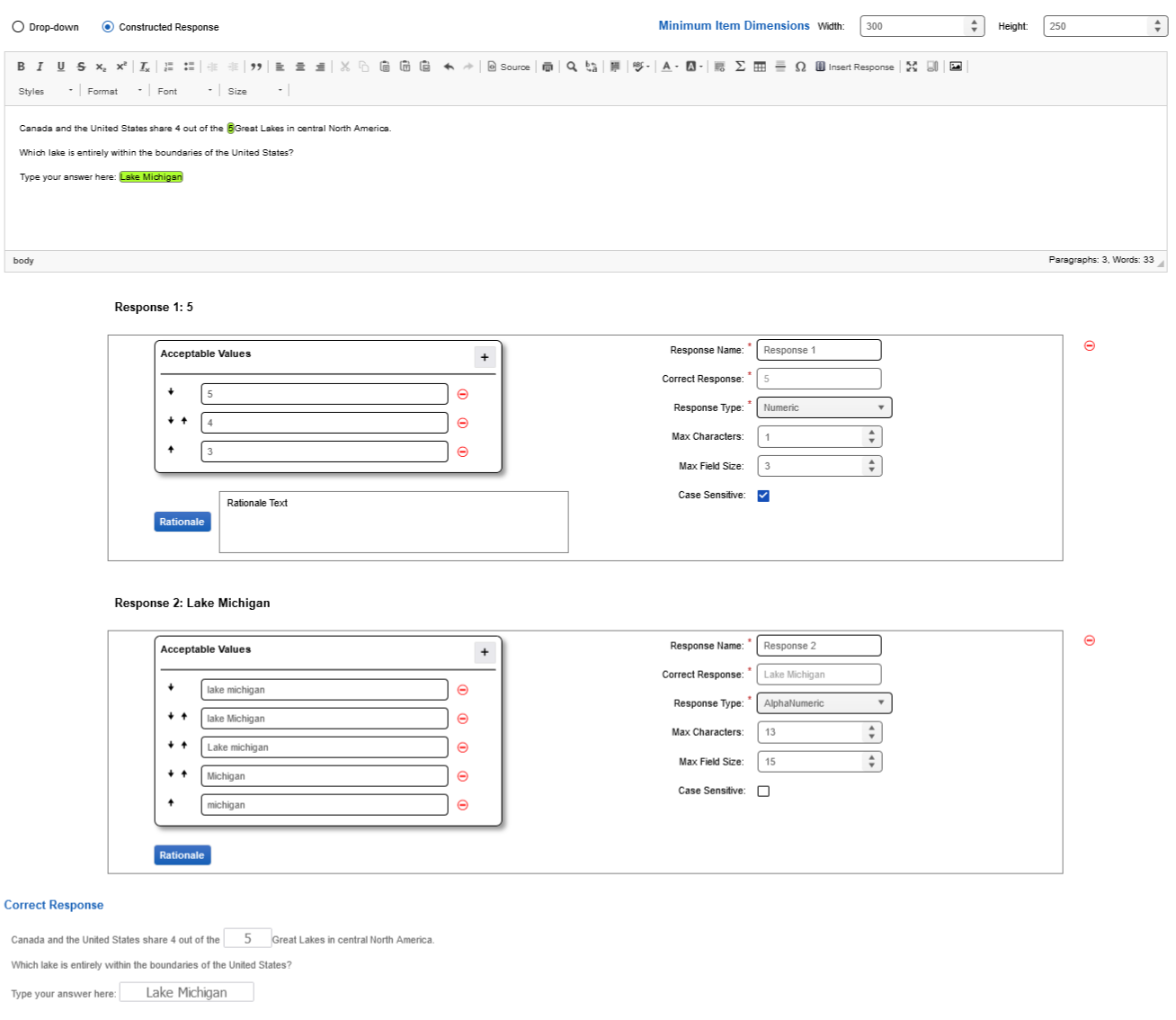


* CB 1 SP preview (Drop-down item)



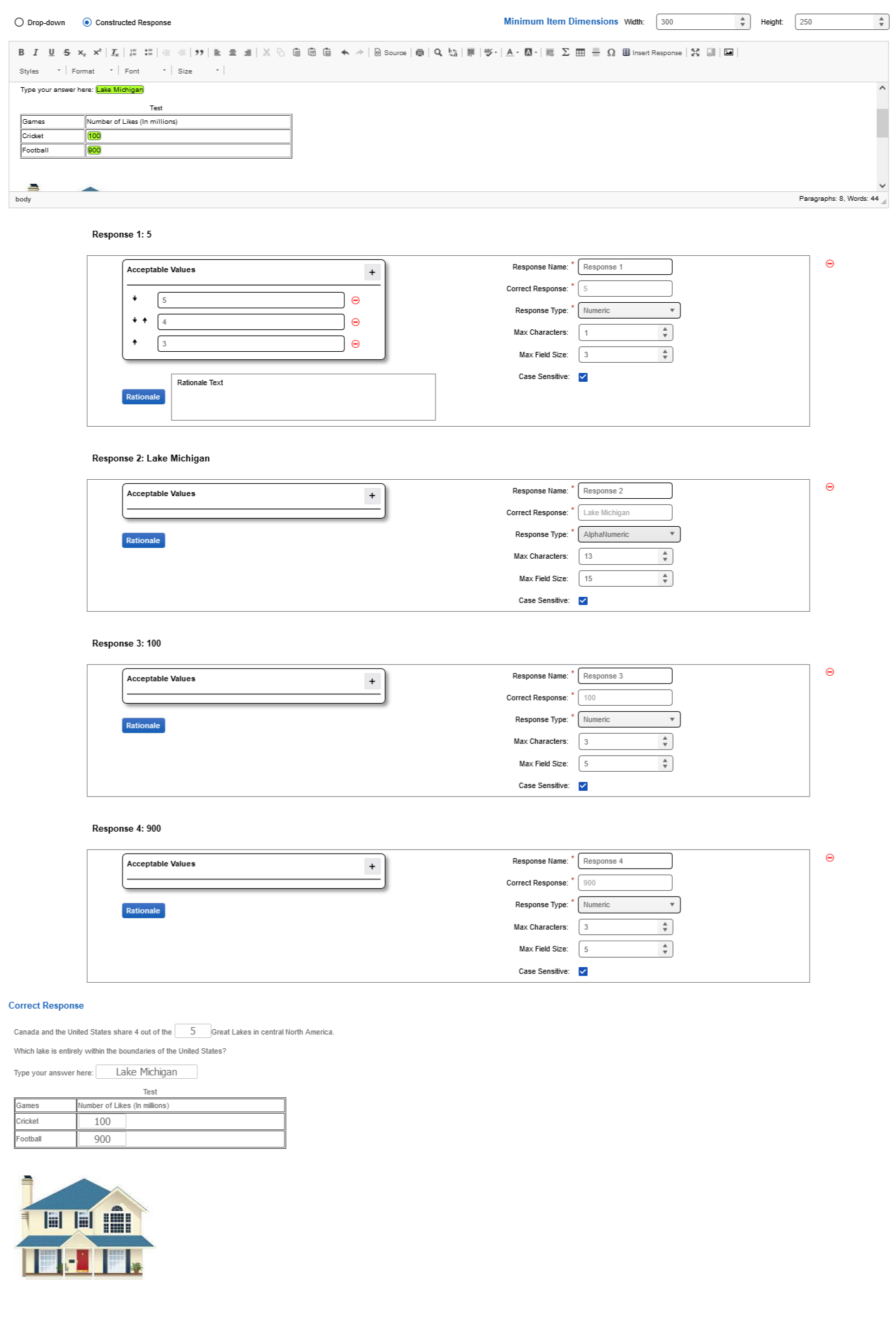
* Content Portal – Current Content Page (Constructed Response item)

Stem content with both numeric and Alphanumeric as Response Type, Acceptable Values for the generated responses.



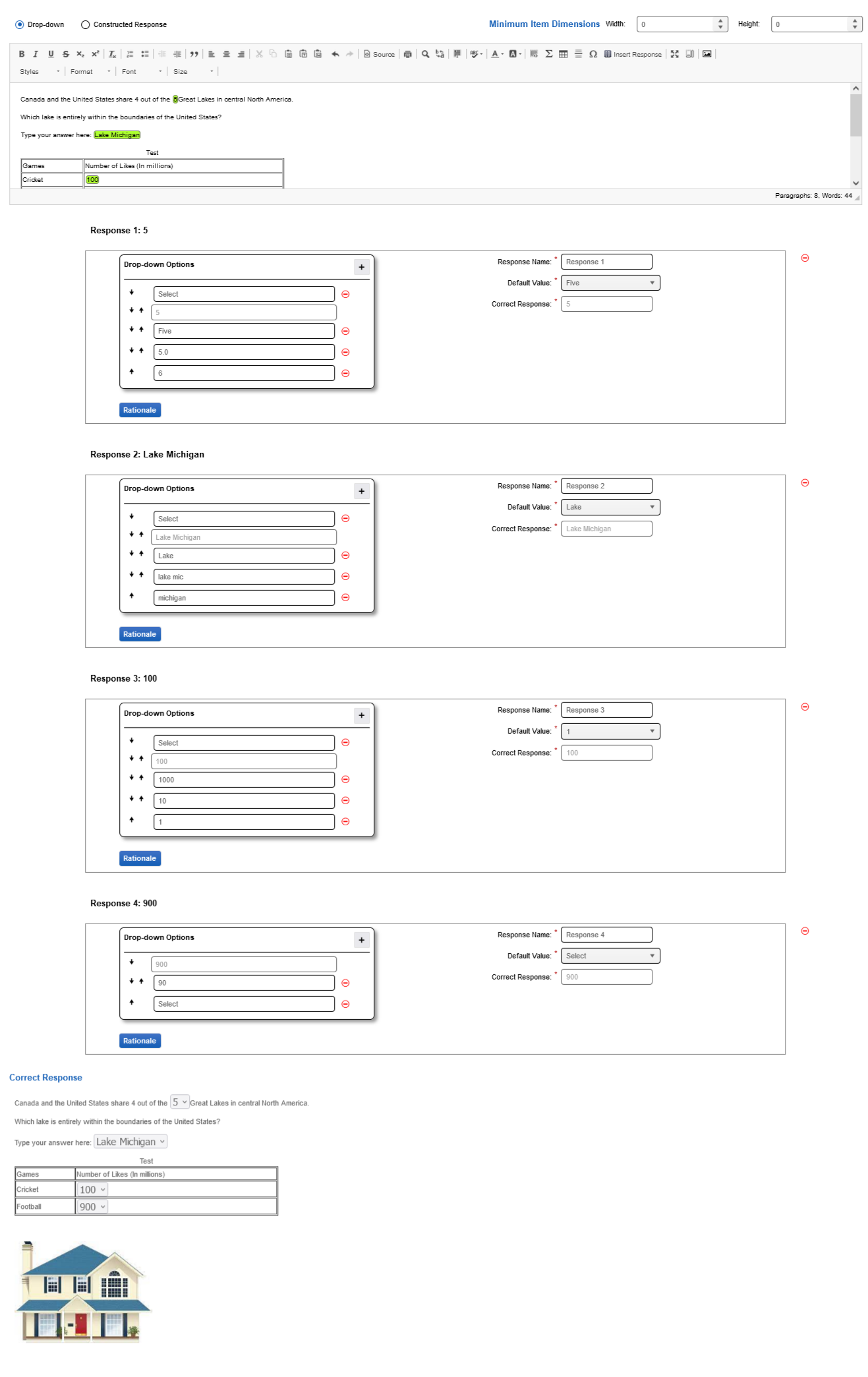
* Content Portal – Current Content Page (Constructed Response item)

Stem Content with numeric as response type and table and image.



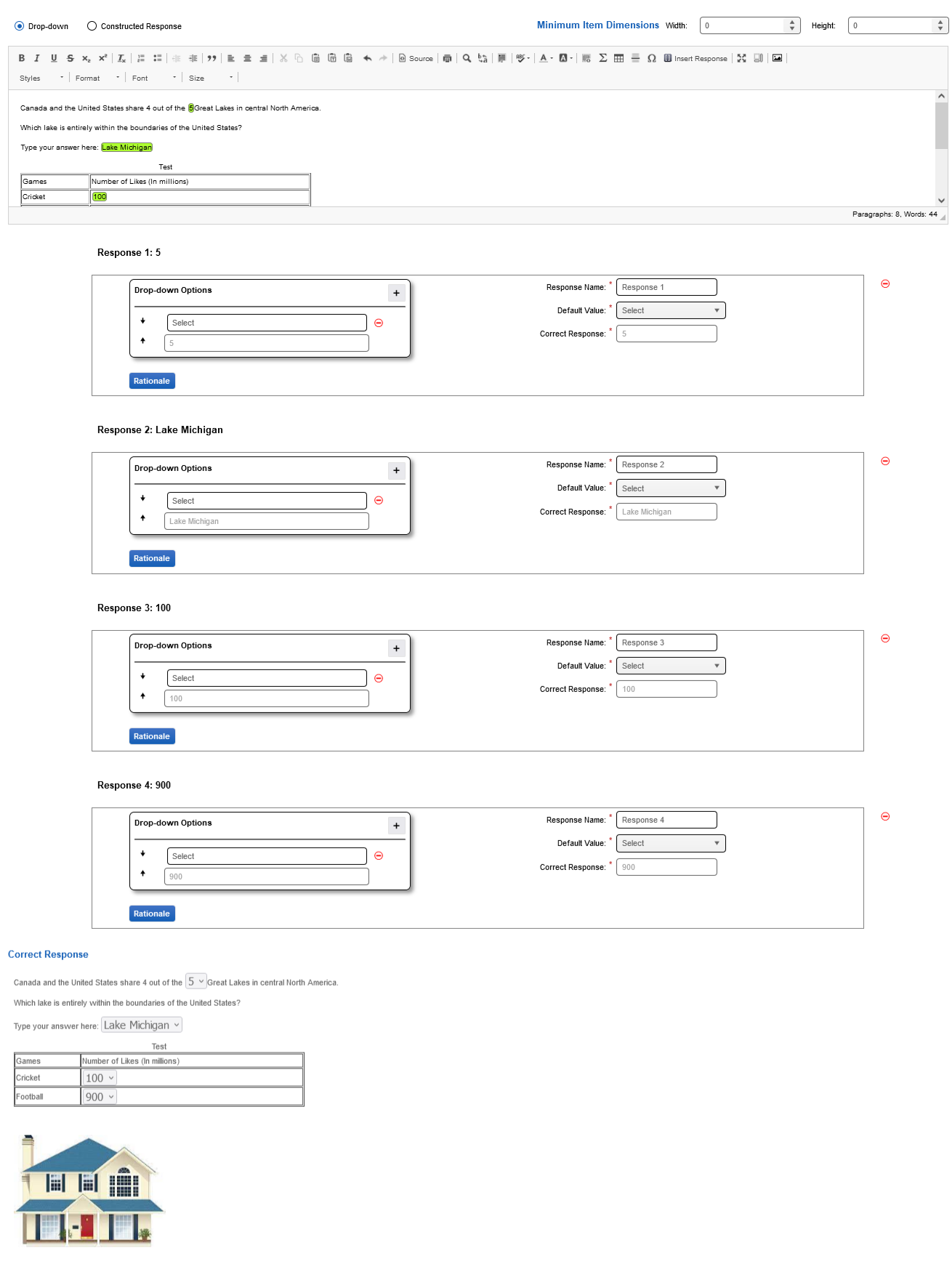
* Content Portal – Current Content Page (Drop-down item)

Stem Content with text and multiple Drop-down options.



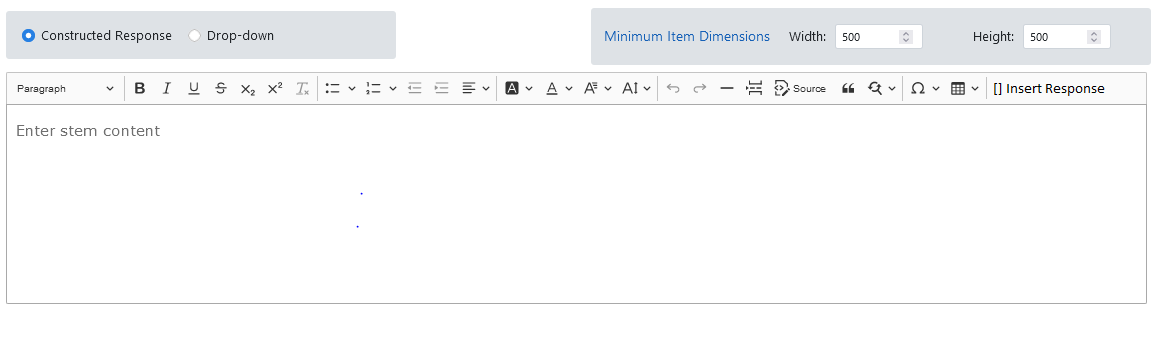
* Content Portal – Current Content Page (Drop-down item)

Stem Content with table data, default drop-down options and image.



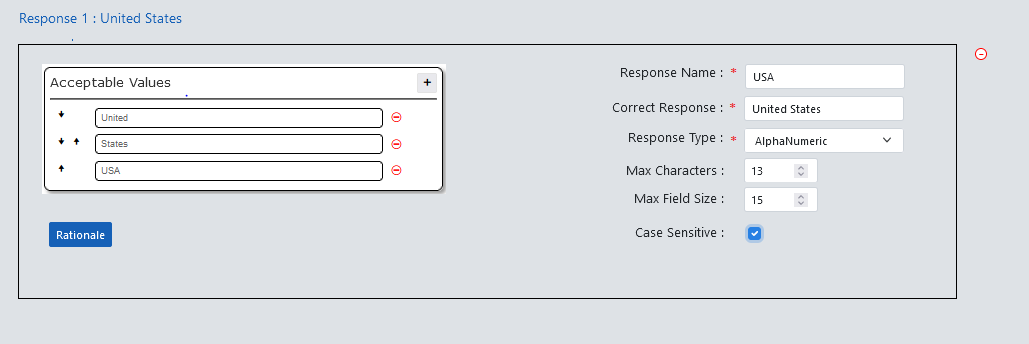
* Content Portal – New Content Page (Constructed Response item)

Item Selection (Selected as Constructed Response), Item Dimension and stem content with extra plugin (I.e., Insert Response)



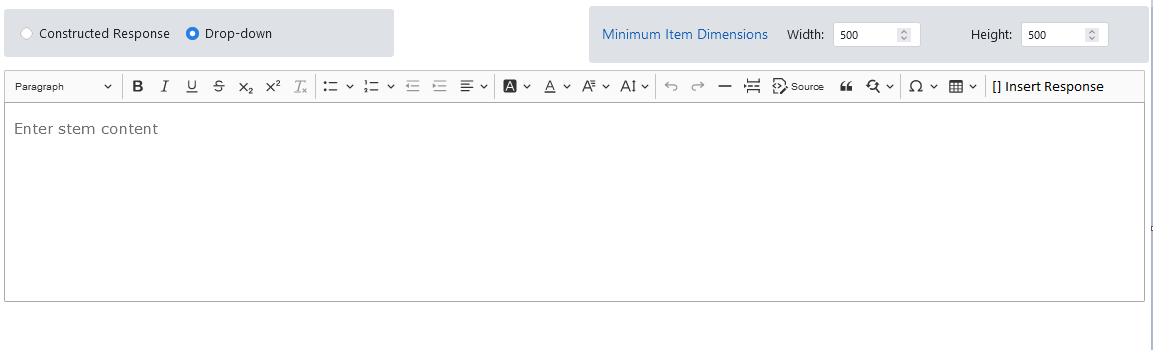
* Content Portal – New Content Page (Constructed Response item)

Response Grid



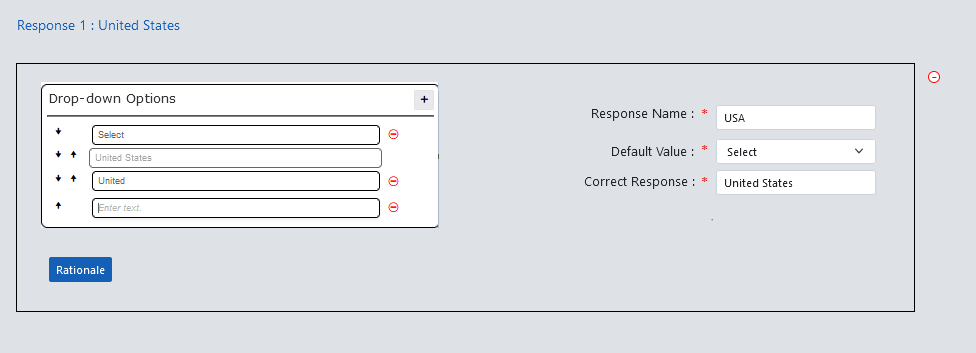
* Content Portal – New Content Page (Drop-down item)

Item Selection (Selected as Drop-down), Item Dimension and stem content with extra plugin (I.e., Insert Response)

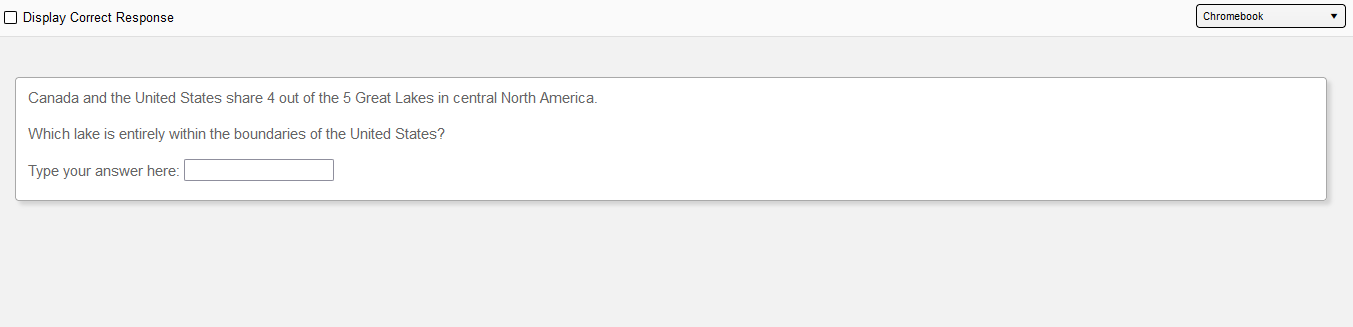


* Content Portal – New Content Page (Drop-down item)

Response Grid

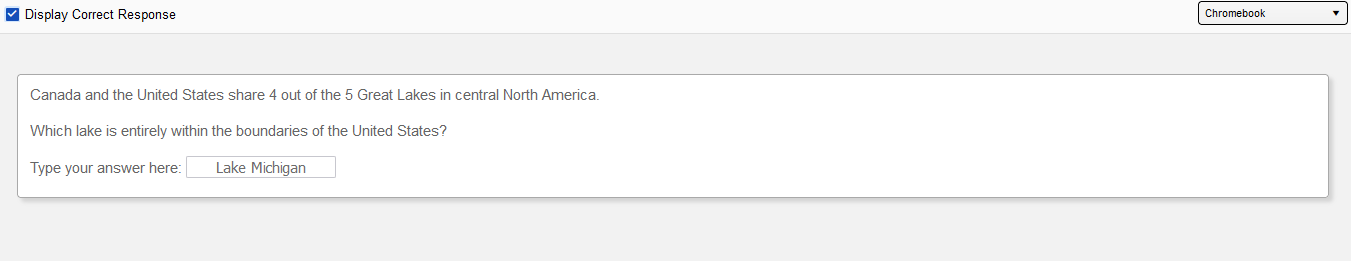


* Content Portal – Current Preview (Constructed Response item)

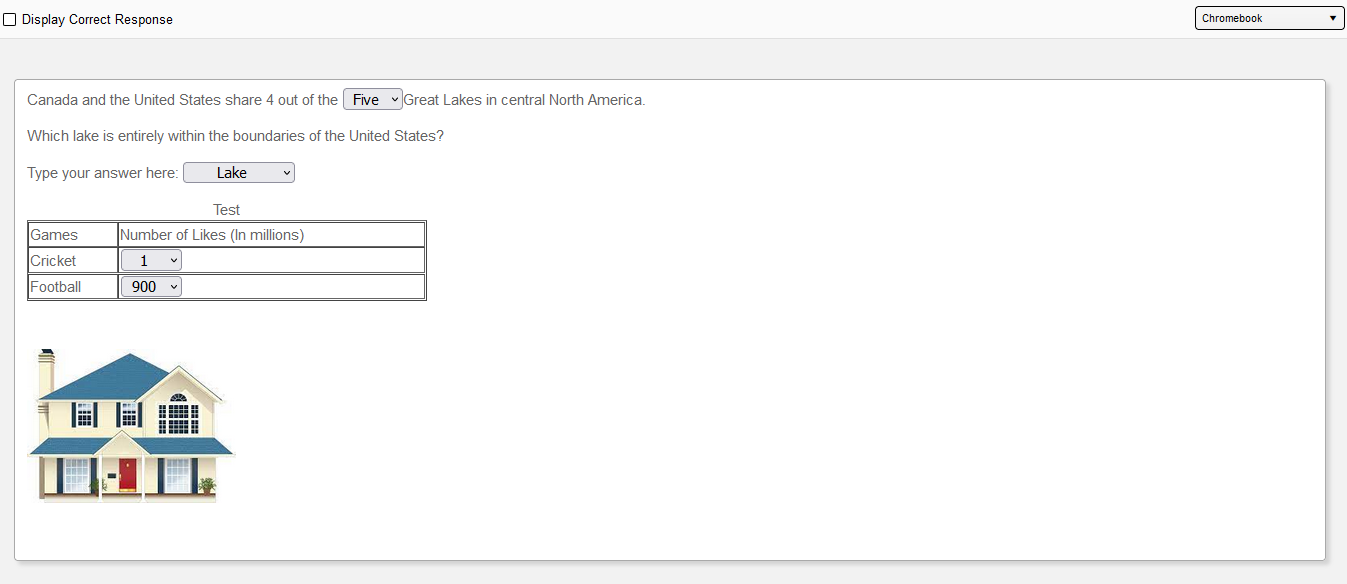


* Content Portal – Current Preview (Constructed Response item)

Enabling Display Correct Response.

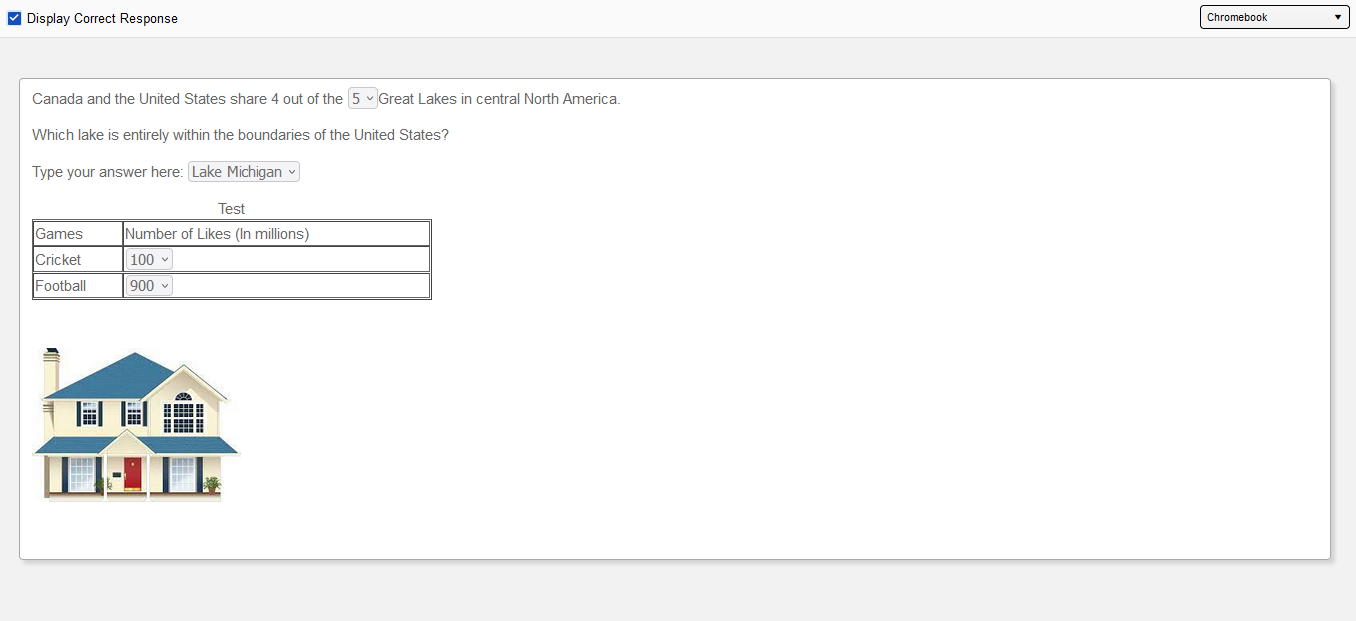


* Content Portal – Current CP Preview (Drop-down item)



* Content Portal – Current CP Preview (Drop-down item)

Enabling Display Correct Response



# Technical Updates

## Outstanding Defects

**Backlog defects**:

* US19790 - Constructed Response - add a "numeric +" option that will allow for a few additional characters beyond just numbers needed for math? ex. allow a comma in the number 2,000.
* 4017 - There is an additional popup when you try to save the item with a "final" status. It is asking to confirm correct and acceptable responses. This doesn't appear on any other item type.
* You are unable to create a drop-down menu for the text entered in the math tool, Subscript/superscript. This is the same functionality as current CB, but is not as obvious that you cannot do these things currently. This will be added as a tip until this functionality has been added. The functionality need to be added so we can create a drop down menu for text entered in the math tool.
* In Drop-down scoring – when user chooses subtractive scoring, it should display all options with the max score divided between the correct responses.
* There are some nice warnings when trying to set the response type to a value that does not make sense with the correct value. Could something similar be added when trying to change the "Max Characters" below the number of characters in the correct response? It auto populates the appropriate value when creating the response, but it can be manually adjusted lower and not reflect as an issue.
* DE22241 - Absolute value appears in the item text box but does not appear in the item Preview.
* US19778 - For reordering drop-down options, would it be possible to align all of the down and up arrows in separate columns? The last option having the up arrow aligned with the down arrows is confusing.
* 1215 - Unable to change font in drop-down options (wanting to italicize a variable, or include superscript for exponents, use MathLive).
* Defect #1128 - For Drop-down you are unable to create math tool content as a response option. Based on the CP/LM Demo and the "Math Edit - CR and Drop-down" email, we should have this functionality for drop-down items.

## Business Flow

* User can change the item type as Constructed Response / Drop-down using Radio button.
* The Min item dimensions functionality will be same as every item.
* Stem content will have the questions.
* User needs to select the text that needs to have correct response.
* After selection click on the Insert Response option from Ckeditor Toolbar, Correct response and response box will be generated with response name.
* For Constructed Response, User can see Acceptable Values grid and user can add values using Add icon.
* User can remove the added acceptable values using remove icon
* User can swap the values using swap icons.
* User can edit the default response name (i.e., Response 1).
* User can see the correct response, selected in ckeditor and that cannot be modified.
* User can choose response type as AlphaNumeric / Numeric, by default it will be based on the user selection in ckeditor.
* User can modify the Max Characters and Max field size values, default values will be based on the word user selected from ckeditor.
* User can enable / disable case sensitive for the correct response, by default it will be enabled.
* User can add rationale text for the Response by clicking on rationale button.
* User can be able to remove the response using remove icon.
* For Drop-down item, User can response box and correct response will be generated for the selected text.
* User can edit the drop-down options in the Drop-down option grid.
* User can add drop-down options using Add icon, can remove using Remove icon.
* User cannot remove the default value / correct response value.
* User can swap the options using swap icons.
* User can edit the default response name (i.e., Response 1).
* User can see the correct response, selected in ckeditor and that cannot be modified.
* User can choose the default values that needs to be shown in SP / CP Preview.
* User can add rationale text for the Response by clicking on rationale button.
* If display correct response is enabled, then correct response will be set in the Preview and user will not be able do any action on it.

## Implementation

***Create:***

* + - As part of Constructed Response / Drop-down item creation, will create the CRDD.jsx file inside components/create/cr\_dd folder.
    - This CRDD.jsx is completely responsible for rendering the item content creation except for the correct response render.
    - Subcomponent for Item Selection will be created and commonly used for Item Selection / Swapping, Drop-down, and Constructed Response items.

import TypeSelection from './TypeSelection';

<div data-testid={'id-container'}>

<fieldset className={'bg-light p-3 rounded m-1'}>

//TypeSelection will be called by passing all the required parameters

<TypeSelection

selectionType={item?.item\_json?.itemTypeCode}

onChange={handleItemTypeChange}

/>

</fieldset>

</div>

* + - Shared component ItemDimension will be used for this item.

import ItemDimensions from '../shared/ItemDimensions';

//ItemDimension will be called by passing all the required parameters

<ItemDimensions

 minWidth={item?.item\_json?.minItemWidth || 0}

 minHeight={item?.item\_json?.minItemHeight || 0}

 onChange={(dimension) => {

     if (dimension?.minWidth !== item?.item\_json?.minItemWidth) {

     updateItemJson('minItemWidth', dimension.minWidth);}

 if (dimension?.minHeight !== item.item\_json.minItemHeight){

   updateItemJson('minItemHeight', dimension.minHeight);}

}}

/>

* + - Common CKEditorBase will be used for the stem content as we need to use the extra plugin for Constructed Response and Drop-down items.

import CKEditorBase from '../../common/editors/ckeditor/CKEditorBase';

 //CKEditorBase will be called by passing all the required parameters

<CKEditorBase

    data={item.item\_json?.stemContent}

onUpdate={updateItemJson}

fieldName={'stemContent'}

className='content\_style'

    placeholder={stemContentPlaceholder}

    dataTestId='stem-content-editor'

config={{ extraPlugins: "insertResponse" }}

{...props}

/>

* + - CKEditor plugin will be created and used for ‘Insert Response’ when manual selection – Correct response is selected from the stem content.
    - CKEditor plugging - insertResponse(It should work like the ‘Bold’ plugin, when it is enabled all the selection should be highlighted with background color and box with border will be created for the correct response).
    - Subcomponent for TagInput grid will be created and commonly used for rendering the Drop-down options in Drop-down item and Acceptable values in Constructed Response item.

import TagInput from './ TagInput;

//TagInput will be called by passing all the required parameters

<TagInput

heading= {}

correctResponse={ item?.item\_json?.correctResponse}

defaultValue={ item?.item\_json?.defaultValue}

defaultDropDown={defaultDropDown}

onChange={(newTags) => createTags(newTags, item?.item\_json?.correctResponse.id) }

setDefaultResponse={setDefaultResponse}

titlePlaceholder={ 'enter text.'  }

itemType={ item?.item\_json?.itemType }

/>

* + - Subcomponent for Drop-down will be created and commonly used for all the fields Response Type in Constructed Response item and Default Value in Drop-down item.

import DropDown from './DropDown';

//DropDown will be called by passing all the required parameters

<DropDown

      data={getResponseLabelsData('responseType')}

      labelCode='response\_type'

      onUpdate={updateItemJson}

      updateKey='responseType'

      isRequired={true}

      showSelect={true}

      value={item.item\_json?.responseType}

      dataTestId={'cr-response-type'}

 />

* + - Common bootstrap/Input component will be used for other options Response Name, Correct Response, Max Characters, Max Field Size, Case sensitive in Constructed Response Item and Response Name, Correct Response, in Drop-down item.

import Input from '../../common/bootstrap/Input';

//Input will be called by passing all the required parameters

<Input

     id={'cr\_response\_name'}

     type={'text'}

     name={'cr\_response\_name'}

     label={{code: 'cr\_response\_name',

             addColon: true,

             className: 'text-right',

             style: {marginTop: '5px'}

           }}

inline={true}

 />

* + - Required constants, labels and messages will be added in the appropriate files.
    - Test files will be created for all the jsx files and located in the same cr\_dd folder itself.
    - New component CRDD will be imported and mapped with the unique item code in the CreateInterface.jsx.

***Response:***

* + - One new folder ‘cr\_dd’ will be created in components/display/item/ for creating the main response component for Constructed Response and Drop-down items.
    - CRDDResponses.jsx is the main component of responses that will receive the inputs and pass that to CRDD response sub-components.

Import ConstructedResponse from './ConstructedResponse;

Import Dropdown from './Dropdown;

//ConstructedResponse will be called by passing all the required parameters, if the itemtype is Constructed Response (I.e., itemTypeCode : 'CR').

<ConstructedResponse

item={item}

/>

//Drop-down will be called by passing all the required parameters, if the itemtype is Drop-down (I.e., itemTypeCode : 'DD').

<Dropdown

item={item}

/>

* + - Dropdown.jsx is the subcomponent that will handle response of Drop-down item.
    - ConstructedResponse.jsx is the subcomponent that will handle the response of Constructed Response item.
    - This same CRDDResponses component will be called from both the CRDD (Create) and CRDDPreview (Preview) component.

***Preview:***

* + - New folder ‘cr\_dd’ will be created in components/display/item for the Constructed Response and Drop-down item preview.
    - CRDDPreview.jsx is the component responsible for rendering the item preview.
    - CRDDResponses component will be called from the CRDDPreview.

import CRDDResponse from '../../response/cr\_dd/CRDDResponse;

// CRDDResponse will be called by passing with all the required parameters.

<CRDDResponse

item={item}

config={config}

onUpdate={handleResponse}

showCorrectResponse={showCorrectResponse}

selected={selected}

/>

* + - While calling the CRDDResponses will be passing the stem content with help of StemFormatter.

***Stories:***

* + - For Constructed Response and Drop-down components Create, Response and Preview will have stories to render and check the UI.
    - Will have a folder called ‘CR’ inside the stories/assets which contain json files with all the possible combinations as mentioned below.
      * cr\_numeric\_response
      * cr\_alphanumeric\_response
      * cr\_formated\_stem\_content
    - Will have a folder called ‘DD’ inside the stories/assets which contain json files with all the possible combinations as mentioned below.
      * dd\_numeric\_response
      * dd\_alphanumeric\_response
      * dd\_formated\_stem\_content
    - CRDDNew.stories.jsx will be created to call the create CRDD component with default UI.
    - CRDD.stories.jsx will be created under the stories/create/alternate to check the create CRDD component with alternate UI.
    - Same as for other styles, separate story files will also be created and used.
    - CRDDDisplay.stories.jsx will be created under the stories/display for Constructed Response and Drop-down preview check with default UI.
    - Inside the CRDDDisplay.stories will be using all the json that we have created under the asset which help us to check the preview in all the combinations.
    - Display stories will also have a separate story for all the available styles.5
    - For the CRDD response alone will have a separate story CRDDResponses.stories.jsx under stories/response.

## Software Updates Needed to Align with Project Standards

## N/A

## How to consume

* + - * This kite-react-items component will be used in the cp-protype application, and CreateInterface and PreviewInterface will be called.

<CreateInterface

item={item}

onUpdate={updateItem}

config={config}

/>

<PreviewInterface

item={itemObj}

onUpdate={onUpdate}

config={config}

/>

* + - * In the kite-react-items CRDD response component is common for both Create and Preview components.
      * In the Create component it will be called without stemComponent like below.

<CRDDResponses

item={item}

config={config}

showCorrectResponse={true}

/>

* + - * In CRDDResponses component, a subcomponent will be called based on the item type.
      * If the item type is Constructed Response

<ConstructedResponse

item={item}

config={config}

/>

* + - * If the item type is Drop-down

<Dropdown

item={item}

config={config}

/>

* + - * In the Display component it will be called with the stemComponent like below.

<CRDDResponses

item={item}

config={config}

onUpdate={handleResponse}

showCorrectResponse={showCorrectResponse}

selected={selected}

/>

## Click History

* + - The click history structure for Constructed Response item is below

      "responses": [

        {

          "responseoption": "6~",

          "ts": "Thu Apr 29, 2021, 08:56:42 GMT-0500 (Central Daylight Time)",

          "score": 0,

          "isCorrectResponse": false

        },

      ]

* + - The click history structure for Drop-down item is below

      "responses": [

        {

          "responseoption": "6~",

          "ts": "Thu Apr 29, 2021, 08:56:42 GMT-0500 (Central Daylight Time)",

          "score": 0,

"referencescore": null,

          "isCorrectResponse": false

        },

      ]

# Test Scenarios

* Below is the Constructed Response and Drop-down Test Scenarios.

**Content**

1. The User will be able to select Constructed Response / Drop-down item and open the content accordion.
2. Users will be able to save the Item by giving required fields. User will be able to Edit the Item after saving in Draft/Final/Approved/Complete.
3. Users will be able to save the item in Draft status without completing all required fields. Once saved, those fields can be edited and saved again.
4. The User will be able to switch the Constructed Response / Drop-down item by changing radio buttons. The stem content will be retained, and other information/values will reset to defaults
5. The User will be able to set Item dimensions for the item in item dimensions field.
6. The User will be able enter content in stem.
7. Users can select the content in the stem and use the Insert Response plugin from CKEditor toolbar to generate a response.
8. For Constructed response users can add the Acceptable values by clicking the Add icon.
9. Users can swap the Acceptable values using up and down arrows.
10. User can remove the Acceptable values using remove button.
11. User can edit the Response Name for both Constructed Response and Drop-down.
12. For both Constructed Response and Drop-down, User can check correct response is appearing correctly on the response.
13. For Constructed Response, User can select the response type [Alphanumeric / Numeric] from the drop down.
14. For Constructed Response, User can adjust Max Characters / Max field size using Numerical stepper box.
15. For Constructed Response, User can check the case sensitive to make the response as case sensitive.
16. For Drop-down, User can add the drop-down options by clicking the add button.
17. For Drop-down, User can swap the drop-down options using swap button.
18. For Drop-down, User can remove the drop-down options using remove button.
19. For Drop-down, User can choose the default values for the response from the drop down.
20. For both Constructed Response and Drop-down, User can be able to add rationale to the response.
21. For both Constructed Response and Drop-down, User can remove the response using remove icon.

**Preview**

1. The User will be able to open the preview, by clicking on the preview button in common header.
2. The Stem content that we created should be shown in CP (Content Portal) Preview.
3. For Constructed Response, User selected content in stem content area and click on “Insert Response” option in content accordion, input box will be generated in CP Preview according to the response type.
4. For Constructed Response, If User select case sensitive checkbox in content accordion, CP Preview input box will allow only exact match with case sensitivity.
5. For Constructed Response item, In CP Preview user can be able to enter the response content in response box/input box.
6. For Constructed Response, input box can allow maximum characters up to max characters value given in Response.
7. For Drop-down If User selected content in stem content area and click on “Insert Response” option in content accordion, drop down box will be generated in CP Preview.
8. For Drop-down Default value for drop down box in CP Preview is taken from Response.
9. In Drop-down item User can show multiple drop-down values by adding dropdown options in Response.
10. For Drop-down item, In CP Preview user can be able to select the response content in response box/drop down box.
11. For both Constructed Response and Drop-down, In preview when user click on the display correct response checkbox the correct response will be displayed.

**Scoring**

1. User can set the scores in scoring accordion.
2. The scoring methods for Constructed Response items are Partial Credit, Correct Only and No Scoring Needed.
3. The scoring methods for Drop-down item are Partial credit, No scoring needed and Correct Only.
4. User will be able to choose the scoring method from the drop-down menu.
5. When a user selected no scoring needed as scoring methods, no additional fields will be displayed.
6. When the user selects correct only as scoring methods, the correct response table will be displayed
7. When user selected Partial credit as scoring methods, the correct response and partial credit table will be displayed, and the max score is divided between the correct responses but is editable.
8. When user selected Partial quota as scoring methods, the correct response and partial quota model table will be displayed, and the max score is decreased by 1 between the correct responses.
9. When a user selected collapsed quota as scoring methods, the correct response, partial credit and collapsed quota tables will be displayed.
10. For collapsed quota current min will be 1 and current max will be number of correct responses and score will be max score.
11. When user selected Subtractive model as scoring methods, the correct response and Subtraction partial credit table will be displayed.
12. For the Subtractive model number of correct responses will be displayed in Subtraction partial credit table.

**Attributes and Metadata**

1. Users will be able to open the attribute and metadata accordion and will be able to select Item metadata, content codes, cognitive settings, tags, and item details.
2. These are mandatory fields and should be filled before saving the Item in a Final /Approved state.

# Final Effort Estimate

|  |  |
| --- | --- |
| **Tasks** | **Effort in Hours** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Questions

* Do we need Math edit functionality in Acceptable values and Drop-down Options?