



12/8/2020

APP Testing Guide

A-Z Learn LMS Application

Cohort 4.2
TECHCAREERS

Contents

Creating a course	2
Pass case	2
Fail cases	3
HTML5 Validation demo	3
JavaScript Validation demo	4
Backend Validation demo	5
Registering a user.....	6
Pass case- Registering a student	6
Pass case- Registering an Instructor	7
Fail cases	8
HTML5 Validation demo	8
JavaScript Validation demo	9
Backend Validation demo	11

Application Testing Instructions

We have implemented three layers of validation across our application. Those are,

1. HTML5 validation
2. JavaScript validation before data submission to backend
3. Backend validation before data submission to database

We have kept consistent validation pattern in the front-end views, and backend end points.

Following are some test cases to check the validation implementation across the three layers mentioned above. Note, same pattern can be followed to test any other front end view or backend end point.

Creating a course

Pass case

- Enter a course name (maximum 50 characters)
- Enter hours required for this course (0-999.99)
- Enter description for the course (maximum 250 characters)

Course

Course was successfully created

Course Name

Python ✓

Hours

40 ✓

Description

Fundamentals of Python, Data Analysis using Python ✓

Back

Create Course

Fail cases

HTML5 Validation demo

- Leave Course Name field empty: this field is required
- Enter 1234 for Hours field: maximum range for this field is 0-999.99
- Leave Description field empty: this field is required

The app will show error messages detailing the errors, see below-

Course

Error: Form was submitted with invalid data fields

Course Name

Please enter a course name.

Hours

1234

Please fill in the valid value for Hours.
Range: 0 to 999.99

Description

Please enter description for Course.

[Back](#)[Create Course](#)

JavaScript Validation demo

- Leave Course Name field empty: this field is required
- Enter 1234 for Hours field: maximum range for this field is 0-999.99
- Leave Description field empty: this field is required

If you try to create course with above data, error messages will be displayed due to HTML5 validation. See below to see validations in web inspector-

The screenshot shows a web form titled "Course" with three input fields: "Course Name", "Hours", and "Description". The "Course Name" field is empty, the "Hours" field contains "1234", and the "Description" field is empty. Below the form, there is a "Back" button and a "Create Course" button. A red error message at the top of the form states: "Error: Form was submitted with invalid data fields".

The web inspector on the right shows the HTML structure of the form. The "Course Name" field is an . The "Hours" field is an . The "Description" field is a with a

Please enter description for Course.

 below it.

Tamper HTML5 validations in web inspector as shown in image below-

- Remove "required" from input for Course Name in web inspector tool
- Change type from "number" to "text" for input for Hours in web inspector tool
- Remove "required" from input for Description in web inspector tool

Now as you have tampered with HTML5 validation, error messages will not be displayed from HTML5 validation. But the second layer of validation- JavaScript validation layer will not allow the form to be submitted. It will show error.

The screenshot shows the same web form as before, but now the "Course Name" field contains "abcd", the "Hours" field contains "1234", and the "Description" field is empty. The "Back" button and "Create Course" button are still present. The red error message at the top of the form still states: "Error: Form was submitted with invalid data fields".

The web inspector on the right shows the HTML structure of the form after tampering. The "Course Name" field is an . The "Hours" field is an . The "Description" field is a with a

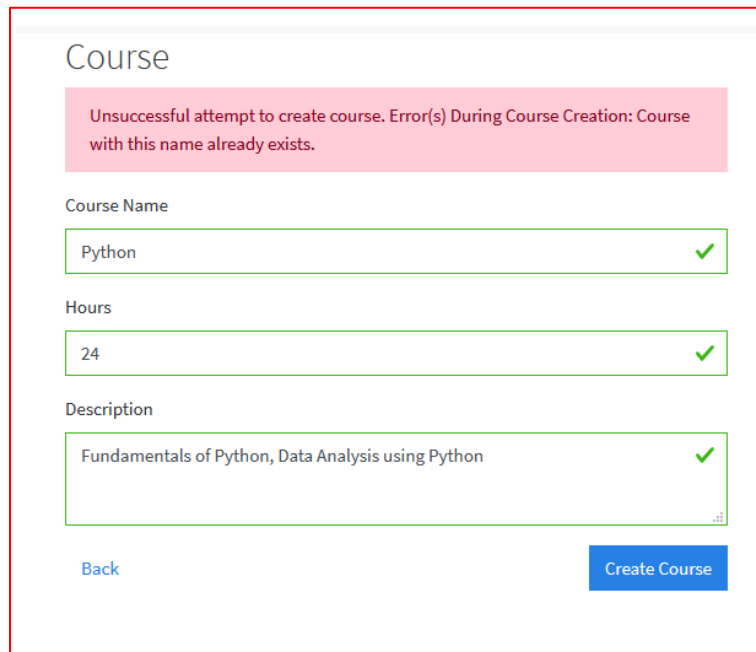
Please enter description for Course.

 below it.

Backend Validation demo

- Try to create a course with valid parameters but keep the course name same as the course which is already created and exists in the database

The HTML5 validation and JavaScript validation will allow the form to be submitted as the data is valid but since it is a duplicate course name backend validation will not allow to create the course with same name. See below-



The screenshot shows a web form titled "Course". At the top, a pink error message box states: "Unsuccessful attempt to create course. Error(s) During Course Creation: Course with this name already exists." Below this, there are three input fields, each with a green checkmark indicating successful client-side validation:

- Course Name:** The input field contains the text "Python".
- Hours:** The input field contains the number "24".
- Description:** The input field contains the text "Fundamentals of Python, Data Analysis using Python".

At the bottom of the form, there are two buttons: a blue "Back" button on the left and a blue "Create Course" button on the right.

Similarly, all other features have three layers of validation in our application. You can follow similar steps for creating cohort, creating homework, updating cohort, updating homework, updating course.

Registering a user

Pass case- Registering a student

- Enter a student name (maximum 50 characters)
- Enter a valid email (maximum 250 characters, format should be a valid email format)
- Enter a password (minimum 8 characters, maximum 250 characters, password should have at least: 1 upper case, 1 lower case, 1 digit and 1 special character)
- Select a cohort ID for student from the drop-down list
- Keep the checkbox for Instructor unchecked as we are registering a student

This will successfully register a student. See below-

Register

Successfully registered the user

Name

Kylie

Email Address

kylie@gmail.com

Password

.....

Cohort

Cohort 4.1

☐ Instructor

[Back](#)

Register

Pass case- Registering an Instructor

- Enter Instructor name (maximum 50 characters)
- Enter a valid email (maximum 250 characters, format should be a valid email format)
- Enter a password (minimum 8 characters, maximum 250 characters, password should have at least: 1 upper case, 1 lower case, 1 digit and 1 special character)
- Select a cohort ID for student from the drop-down list
- Check the checkbox for Instructor as we are registering an Instructor

This will successfully register an Instructor. See below-

Register

Successfully registered the user

Name

Kim

Email Address

kim@gmail.com

Password

••••••••

Cohort

Cohort 4.1

☒ Instructor

[Back](#)[Register](#)



Fail cases

HTML5 Validation demo

- Enter a student name (maximum 50 characters)
- Leave the email field empty: This field is required, maximum 250 characters, format should be a valid email format
- Leave the password field empty: This field is required, minimum 8 characters, maximum 250 characters, password should have at least: 1 upper case, 1 lower case, 1 digit and 1 special character
- Select a cohort ID for student from the drop-down list
- Keep the checkbox for Instructor unchecked as we are registering a student

The app will show error messages detailing the errors because some required fields are missing. See below-

Register

Error: Form was submitted with invalid data fields

Name

Kim

Email Address

Please enter a valid email.
e.g. youremailaddress@domain.com

Password

Please enter a valid password.
Password must have at least: one upper case letter, one lower case letter, one digit, one special character, minimum 8 characters in length.

Cohort

Cohort 4.1

☐ Instructor

[Back](#)

[Register](#)

JavaScript Validation demo

- Enter a student name (maximum 50 characters)
- Leave the email field empty: This field is required, maximum 250 characters, format should be a valid email format
- Leave the password field empty: This field is required, minimum 8 characters, maximum 250 characters, password should have at least: 1 upper case, 1 lower case, 1 digit and 1 special character
- Select a cohort ID for student from the drop-down list
- Keep the checkbox for Instructor unchecked as we are registering a student

If you try to register student with missing data as above, error messages will be displayed due to HTML5 validation. See below to see validations in web inspector-

Register

Error: Form was submitted with invalid data fields

Name

Kendall

Email Address

Please enter a valid email.
e.g. your@emailaddress@domain.com

Password

Please enter a valid password.
Password must have at least: one upper case letter, one lower case letter, one digit, one special character, minimum 8 characters in length.

Cohort

Cohort 4.1

☐ Instructor

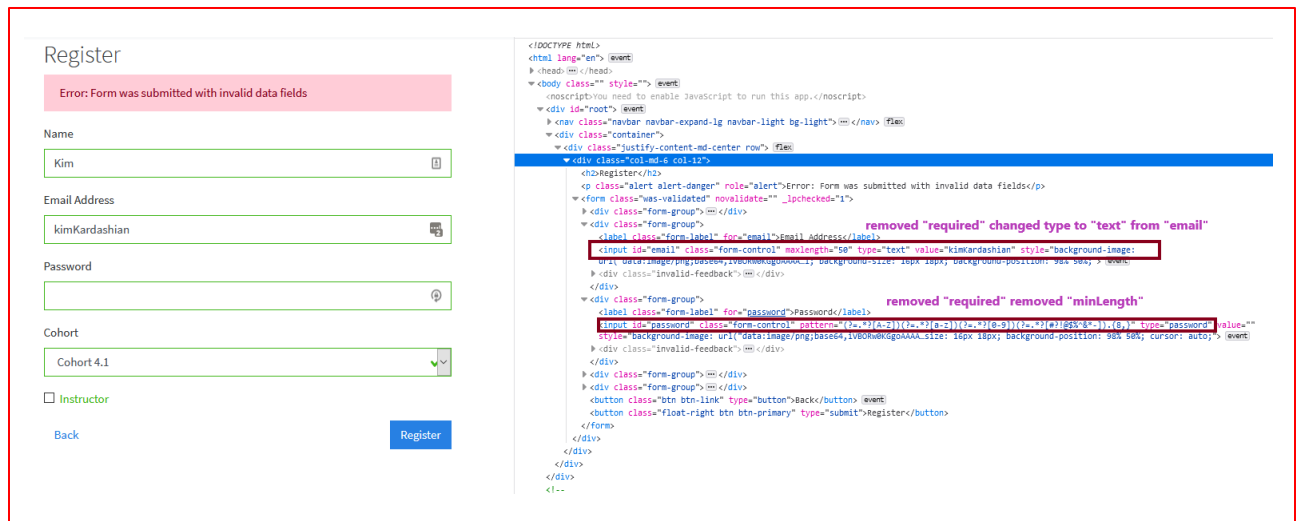
[Back](#)[Register](#)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Register</title>
    <script>
      <!-- You need to enable JavaScript to run this app. -->
    </script>
  </head>
  <body class="bg-light">
    <div id="root">
      <div class="nav navbar navbar-expand-lg navbar-light bg-light">
        <div class="container">
          <div class="justify-content-md-center row">
            <div class="col-md-6 col-12">
              <h3>Register</h3>
              <div class="alert alert-danger" role="alert">
                Error: Form was submitted with invalid data fields
              </div>
              <div class="form-group">
                <div class="form-group">
                  <label class="form-label" for="email">Email Address</label>
                  <input id="email" class="form-control" type="email" value="" />
                  <div class="invalid-feedback">
                    Please enter a valid email.  
e.g. your@emailaddress@domain.com
                  </div>
                </div>
                <div class="form-group">
                  <label class="form-label" for="password">Password</label>
                  <input id="password" class="form-control" type="password" value="" />
                  <div class="invalid-feedback">
                    Please enter a valid password.  
Password must have at least: one upper case letter, one lower case letter, one digit, one special character, minimum 8 characters in length.
                  </div>
                </div>
                <div class="form-group">
                  <label class="form-label" for="cohort">Cohort</label>
                  <select id="cohort" class="form-control">
                    <option value="Cohort 4.1">Cohort 4.1
                  </select>
                </div>
                <div class="form-group">
                  <input type="checkbox" /> Instructor
                </div>
                <div class="d-flex justify-content-between">
                  <a href="#">Back</a>
                  <button type="button" class="btn btn-primary">Register</button>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

Tamper HTML5 validations in web inspector as shown in image below-

- Keep the inputs as in image above
- Remove “required” from input for Email Address in web inspector tool
- Change type from “email” to “text” for input for Email Address in web inspector tool
- Remove “required” from input for Password and remove “minLength” in web inspector tool

Now as you have tampered with HTML5 validation, error messages will not be displayed from HTML5 validation. But the second layer of validation- JavaScript validation layer will not allow the form to be submitted. It will show error. See below-



Backend Validation demo

- Try to register a user with valid parameters but keep the Name same as the username which is already created and exists in the database

The HTML5 validation and JavaScript validation will allow the form to be submitted as the data is valid but since it is a duplicate username, backend validation will not allow to register the user with same name. See below-

Register

Unsuccessful attempt to register user. Error(s) During Registering User: User with this name already exists., Email already exists,please try to use a different email address.

Name

Kim

Email Address

kim@gmail.com

Password

.....

Cohort

Cohort 4.1

☐ Instructor

[Back](#)

[Register](#)

End of document