

CMPS 3680 Lab 1

Spring 2025

Part 1 - Create HTML Form

Create a simple form consisting of the following elements (at least):

1. Three required input fields, including an email field
2. One set of required radio buttons
3. One required text area
4. Two optional fields of your choice

In the form tag you will set the following attributes:

1. Set the name to "contactForm"
2. Set the method to "get"
3. Set the action to an empty string

Part 2 - Javascript Validation and Page Manipulation

For this exercise, **DO NOT** use HTML5 form validation. Please do the following:

1. Define the function `validateForm(event)` and meet these specifications:
 - a. Obtain the data from the form using `document.forms`
 - b. Define a boolean `valid_form` and set its initial value to true
 - c. For each required field in the form, if the field is blank you will do two things:
 - i. Add the css class `error` to the field
 - ii. Update the `valid_form` boolean accordingly
 - d. To validate the email field verify that the string contains both an "@" symbol and a period
 - e. If `valid_form` is false, use `event.preventDefault()`; properly to prevent the form from submitting.
2. Add a "load" listener to the window that will do the following:
 - a. Inside this listener, select the submit button from your form and add a "submit" listener that will call the `validateForm` function you previously defined.
 - b. Obtain the url search parameters using `window.location.search` and `URLSearchParams`
 - c. If `URLSearchParams` contains data, create a new HTML element with the info and append it as a child to your main container below the form.
 - d. The information should only be displayed once the form has been successfully submitted and validated.

Part 3 - CSS

Spend some time using css to style your form to your liking.

Put in some time here and think about your user's experience. Be creative.

For your error class, set the border, outline, and box shadow to red so the invalid entries stand out.

A few things to consider:

1. If the user enters an invalid option initially, but then fixes it, the error class should no longer be present.
2. Feel free to use any libraries, tools, or frameworks you are comfortable with.
We will talk more about this during the lab.
3. This is a simple review of HTML, Javascript, and CSS. We are not making any remote calls, just validating the form and displaying the information on the page.

Once your project is complete, I would expect an invalid form attempt to look something like this:



A contact form titled "Contact Form" on a light blue background. The form contains several input fields with red borders indicating validation errors. The fields are: "First Name" (empty), "Last Name" (containing "Royer"), "Email Address" (empty), "Telephone Number" (empty), "Major" (radio buttons for CS, CE, EE, OTHER, with CS selected), and "Comments" (empty). A "RESUBMIT" button is located at the bottom.

Contact Form

* First Name

* Last Name

* Email Address

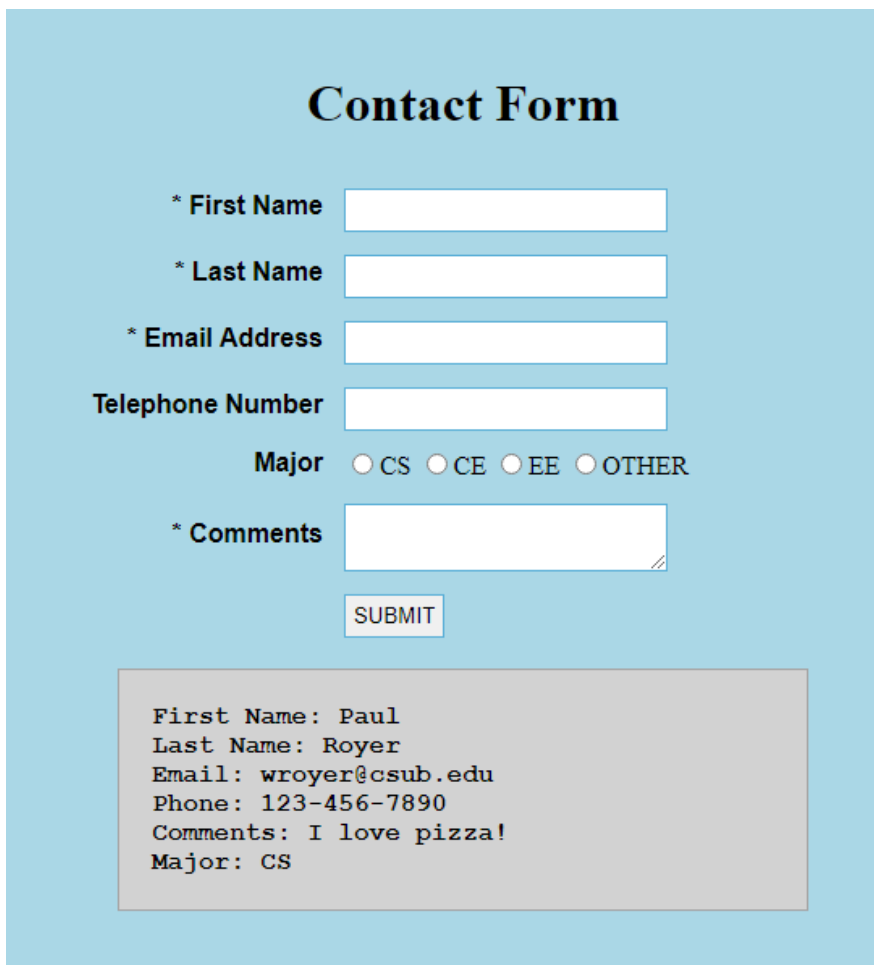
Telephone Number

Major ☒ CS ☐ CE ☐ EE ☐ OTHER

* Comments

RESUBMIT

After a successful form submission it should look something like this:



The same contact form titled "Contact Form" on a light blue background, but now it displays the submitted data in a grey box at the bottom. The input fields are empty. The "Major" section shows radio buttons for CS, CE, EE, and OTHER, all of which are unselected. A "SUBMIT" button is located above the data display box.

Contact Form

* First Name

* Last Name

* Email Address

Telephone Number

Major ☐ CS ☐ CE ☐ EE ☐ OTHER

* Comments

SUBMIT

First Name: Paul
Last Name: Royer
Email: wroyer@csu.edu
Phone: 123-456-7890
Comments: I love pizza!
Major: CS