CENTRAL WASHINGTON UNIVERSITY

Department of Computer Science

Skip to main content
Accessibility features
Contact Information

| **Gellenbeck** | **CS 250** | **Calendar** |

Search ~250 Go

Lab 9: JavaScript Form Validation

Learning Objectives

After completion of this lab, **you should be able to**

- Work collaboratively as a **pair programming** team
- Assignment
- Create a simple form with validation before submit
- Attach an event handler in JavaScript code
- Use a JavaScript method to validate for **required** fields
- Use **regular expressions** to validate user input
- Validate form elements with **event handlers**
- **Register event handler** functions
- **Write event handler** functions
- **To Receive Credit**

Work collaboratively as a pair programming team

CS 250 in-class labs will be done using **pair programming**. Your partner for today's lab is listed in the table below:

Hebeler 204
Grader: John Wright II

Team 1 Abundiz, Sergio Chandler, Alan	**Team 2** Bajwa, Deepinder Dickerson, Andrew	**Team 3** Burley, Jonathan Juarez, Adrian	**Team 4** Byars, Frank Plitkins, Kristofer	**Team 5** Carpenter, Daniel Prescott, Brandon
Team 6 Crockett, Jordan Canada, Justin	**Team 7** Hansen, Christopher Ahmady, Temourshah	**Team 8** Kinkade, Kyle Belfiglio, Alexander	**Team 9** Porter Jr, Anthony Burton, Henry	**Team 10** Rozelle, William Strom, Brandt
Fill in: Taing, Pokuy				

You may wish to review basic pair programming guidelines before you begin.

- One team member (the **driver**) has control of the keyboard/mouse and actively implements the program
- The other team member (the **navigator**) continuously observes the work of the driver to identify tactical defects (such as syntactic and spelling errors, etc.) and also thinks strategically about the direction of the work

You should **change roles** about every ten minutes during lab.

Assignment

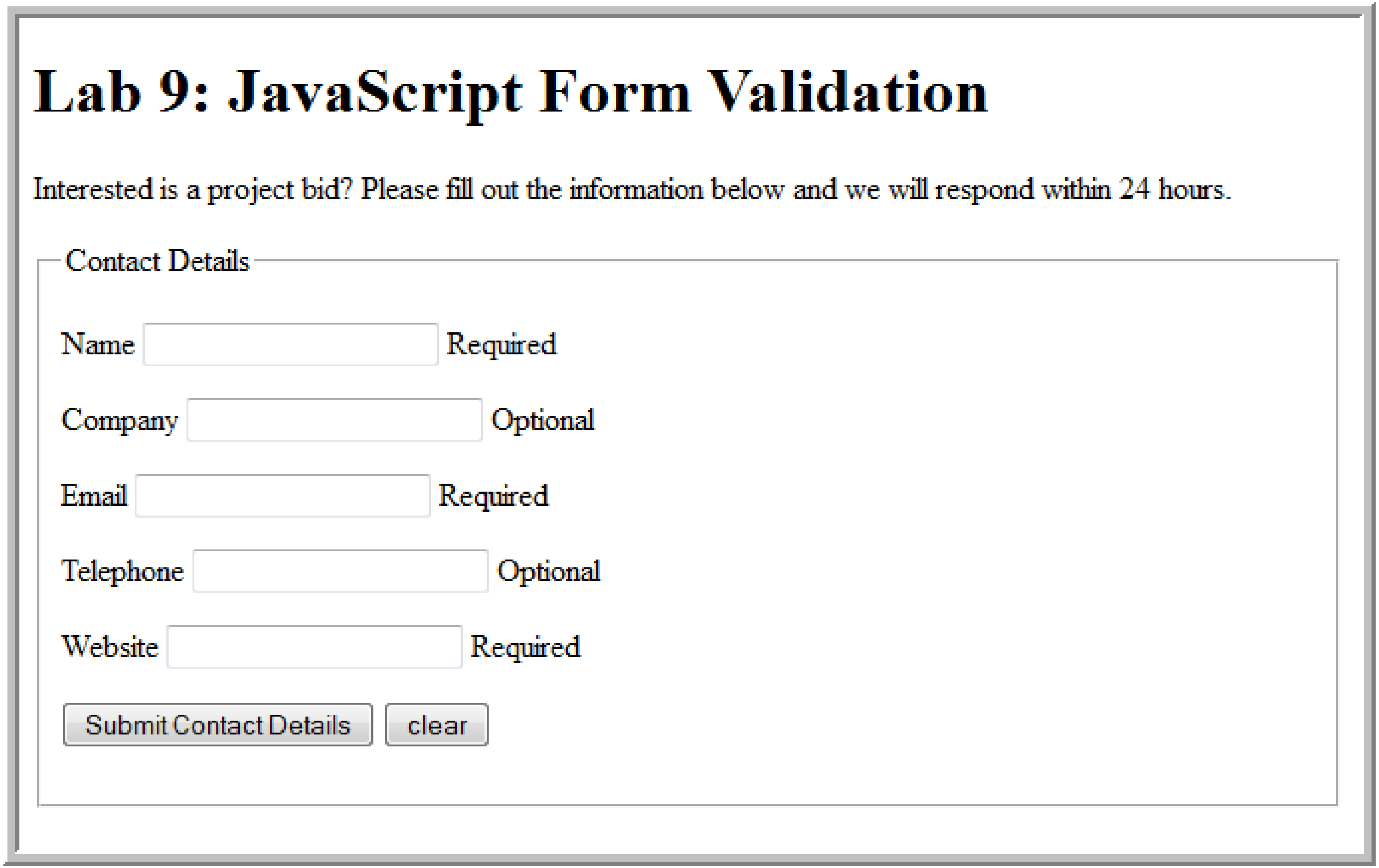
Today's lab will cover form validation using DOM2 Event Model. You will need to create two files inside `U:\htdocs\labs\lab9` folder

- `lab9.html` file to contain a XHTML form (to speed things up, the [code for lab9.html]() is available for download)
- `lab9.js` to contain JavaScript event register handlers and JavaScript validation functions

All elements will use the default styling, **no** CSS is required

Create a simple form with validation before submit

1. `lab9.html` contains the simple HTML form shown below consisting of a `fieldset`, `legend`, **five paragraphs**, each paragraph containing a prompt `label` and a text `input` field for user input, and ending with `submit` and `reset` buttons inside a paragraph element.
    - Note: to speed things up, the **[code for lab9.html]()** is available for download
    - You should **not** make any changes to the code for `lab9.html`; it is validated to XHTML 1.0 strict and has no accessibility errors
    - All you JavaScript code you will write should be in `lab9.js`, linked to in the head section of lab9.html.



Lab 9: JavaScript Form Validation

Interested is a project bid? Please fill out the information below and we will respond within 24 hours.

Contact Details

Name Required

Company Optional

Email Required

Telephone Optional

Website Required

Attach an event handler in JavaScript code

1. Create a new file, `lab9.js`, in folder `u:\htdocs\labs\lab9` to use to provide client-side validation for lab9.html
2. Attach an event handler to validate the form when the user clicks the submit button
   - JavaScript code given below

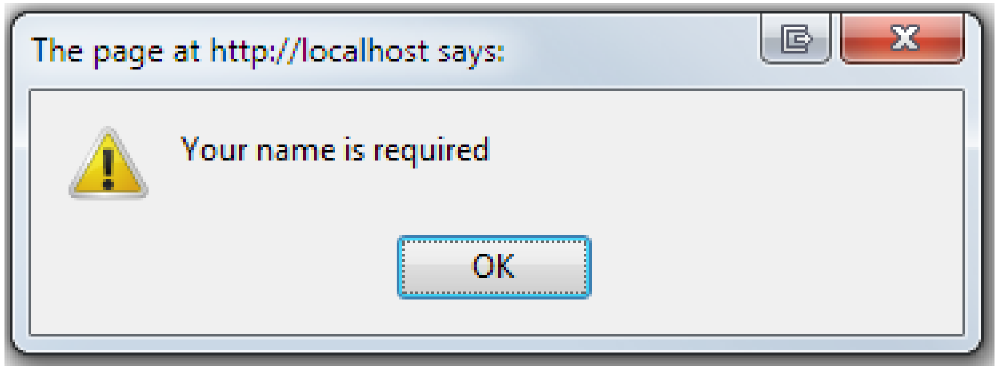
```
window.onload = pageLoad; // global code

// called when page loads; attaches the event handler
function pageLoad() {
  var contactForm = document.getElementById("contactForm");
  contactForm.onsubmit= validate;
}
```

Use a JavaScript method to validate for required fields

1. Write the JavaScript function **validate()** that generates a JavaScript alert message "Your name is required" and **returns** `false` if the Name text input field has not been filled in, otherwise it **returns** `true`.

Tip: document.getElementById may be used to extract the input element and its value property may be used to obtain its value (as a string)

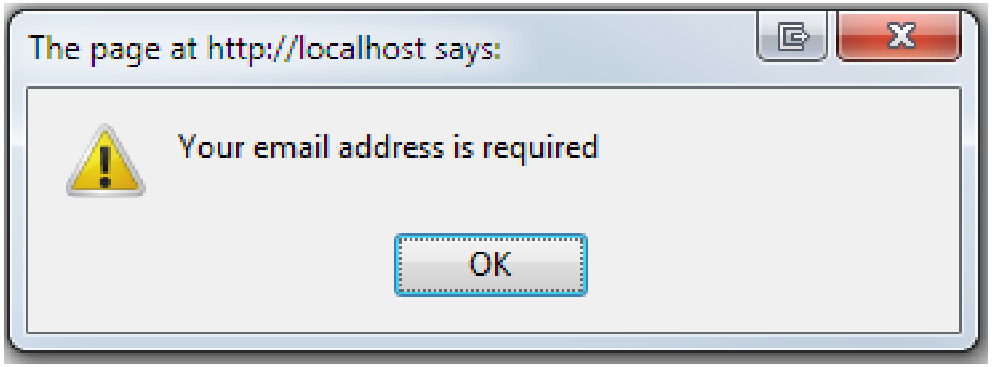
```
function validate() {
   var name = document.getElementById("custname").value;
   if (name == "") {
      alert("Your name is required");
      return false;
   }
   ...
}
```



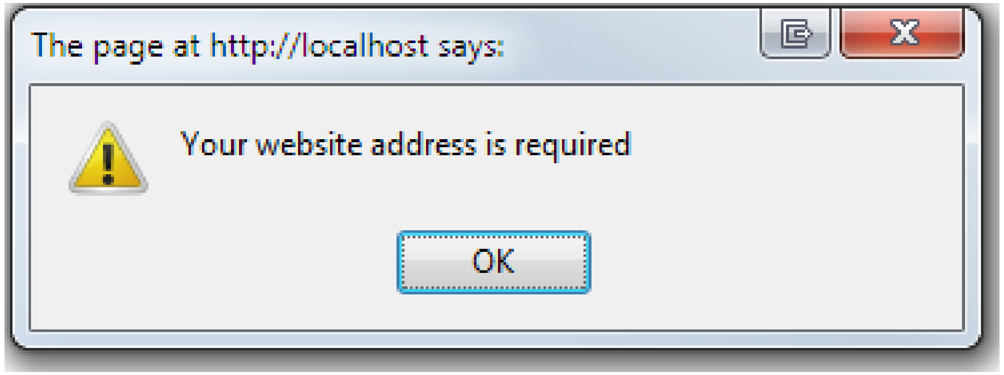
The page at http://localhost says:

Your name is required

2. Add the JavaScript code to make Email and Website **required fields** and let Company and Telephone be ***optional fields***.


The page at http://localhost says:

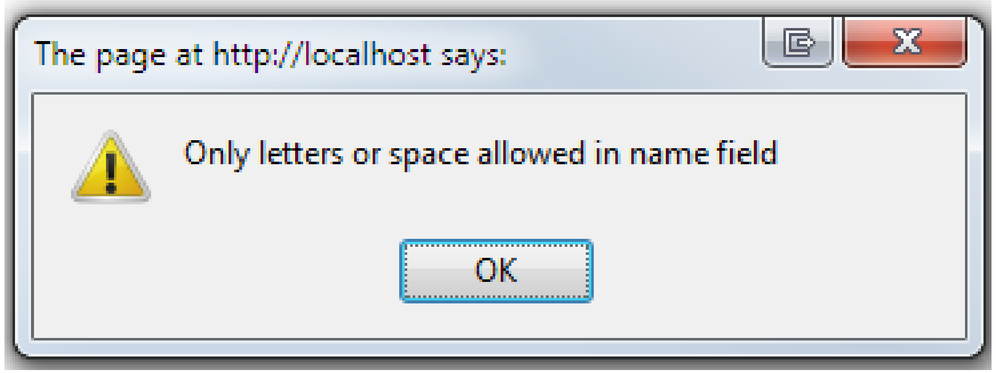
Your email address is required


The page at http://localhost says:

Your website address is required

Use regular expressions to validate user input

1. Write the JavaScript code that uses a **regular expression** to validate that the user name consists of only letters or the blank space character. Generate an alert message if the name is incorrect.



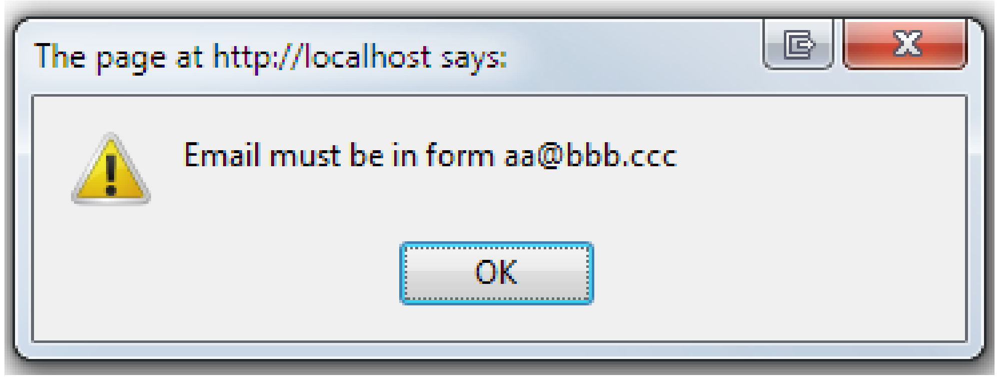
The page at http://localhost says:

Only letters or space allowed in name field

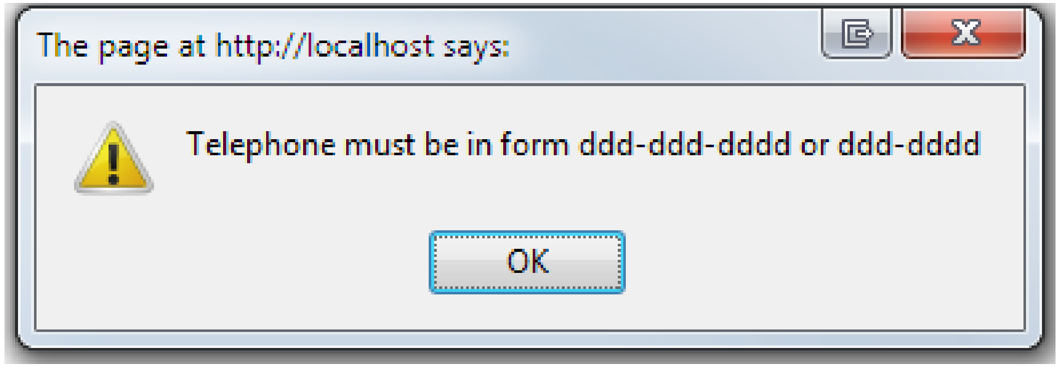
```
    if (!name.match(/^[A-Za-z ]+$/)) {
       alert("Only letters or space allowed in name field");
       return false;
    }
```

Tip: For help testing regular expressions, visit the Online Regular Expression Tester

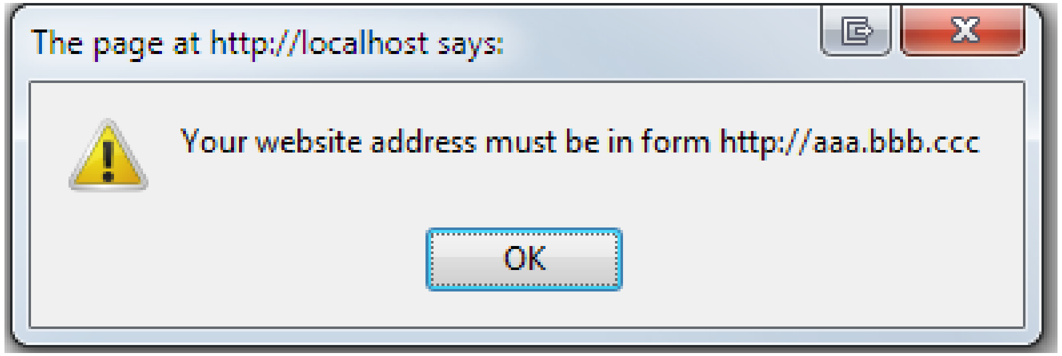
2. Write the JavaScript code that uses a **regular expression** to validate that the **Email** consists aa@bb.cc where *aa*, *bb*, and *cc* are one or more alphanumeric characters.
   - Note: Keep it simple for this lab assignment
3. Write the JavaScript code that uses a **regular expression** to validate that the optional **Telephone** field is in the pattern ddd-ddd-dddd or ddd-dddd where *d* is any digit
   - Note: Keep it simple for this lab assignment
4. Write the JavaScript code that uses a **regular expression** to validate that the **Website** field is in the pattern http://aaa.bbb.ccc where *aaa*, *bbb*, *ccc*, are one or more alphanumeric characters
   - Note: Keep it simple for this lab assignment


The page at http://localhost says:

Email must be in form aa@bbb.ccc


The page at http://localhost says:

Telephone must be in form ddd-ddd-dddd or ddd-dddd


The page at http://localhost says:

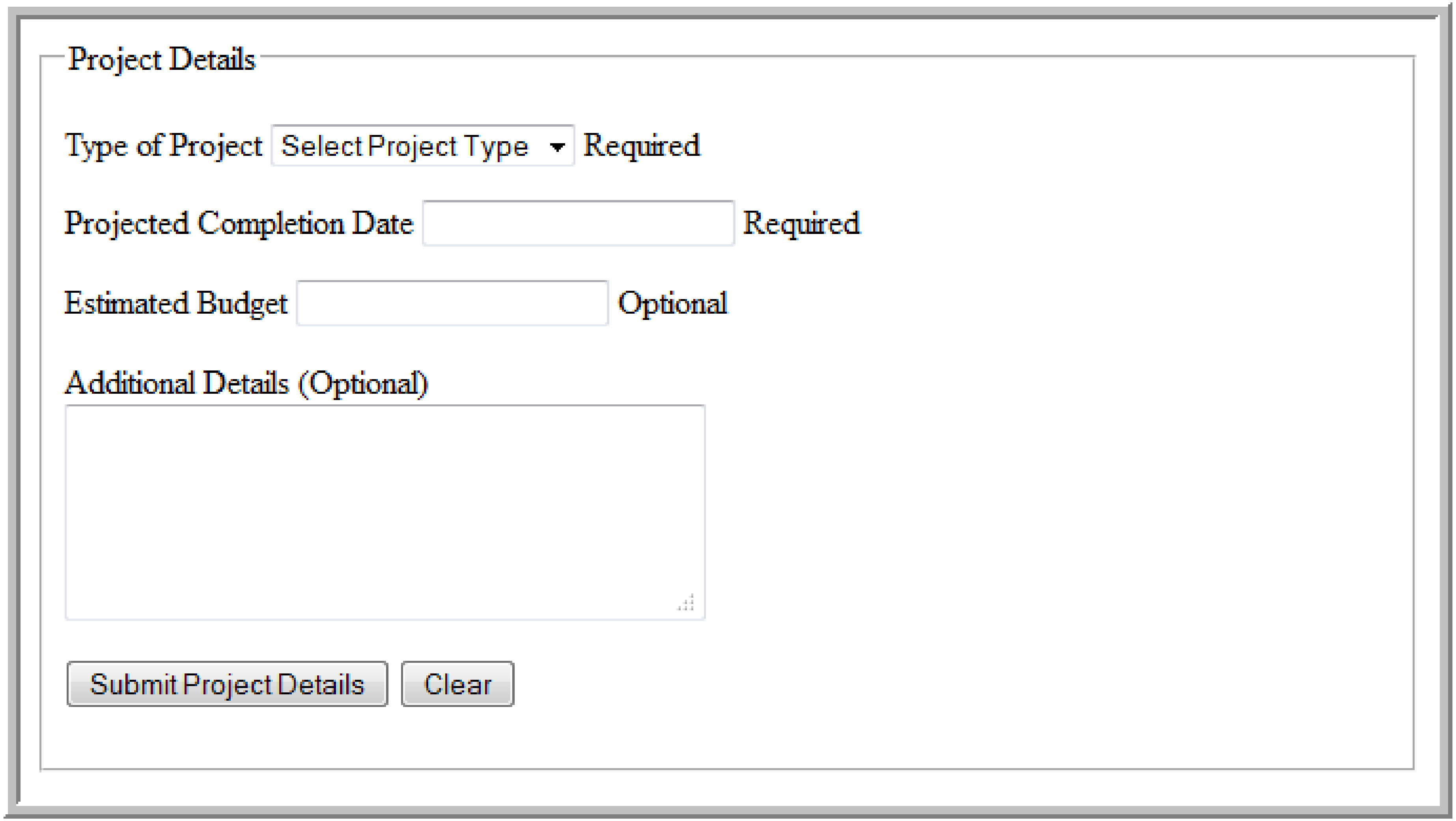
Your website address must be in form http://aaa.bbb.ccc

Validate form elements with event handlers

1. Add a second form to `lab9.html` below your contact details form to allow the user to enter

project details

- Note: to speed up up, the HTML code for Project Details is available for download


Project Details

Type of Project Select Project Type Required

Projected Completion Date Required

Estimated Budget Optional

Additional Details (Optional)

The "Submit Project Details" button is a push `button` element, **not** an `input` element

```
<button id="submitButton" name="submitButton" type="button"
value="submitDetails">
```

```
    Submit Project Details
</button>
```

Project Type and Project Completion Date are **required**, Estimated Budget and Additional Details are **optional**.

Register event handler functions

1. Modify `lab9.js` to register four additional event handlers when the page loads to validate the new form elements and handle the button click

```
document.getElementById("projectType").onblur = checkProjectType;
document.getElementById("completionDate").onblur = checkDate;
document.getElementById("budget").onblur = checkBudget;
document.getElementById("submitButton").onclick = submitDetails;
```

Write event handler functions

Add **four functions** to `lab9.js` to handle the events

1. **function checkProjectType()**
    - if the select projectType's value is "0", show an alert message and return false; otherwise return true
2. **function checkDate()**
    - if the completion date input field's value is empty, show a "date is required" alert message and return false
    - if it does not match the regular expression for "mm/dd/yyyy", show a "date is invalid" alert message and return false
    - otherwise return true
3. **function checkBudget()**
    - if the budget input field's value is empty, return true (budget is *optional*)
    - if the budget is not a number, show a "budget is invalid" alert message and return false
    - otherwise return true
4. **function SubmitDetails()**
    - if the project type, completion date, and budget are valid, display "Project details submitted" in an alert box; otherwise display the appropriate error

To Receive Credit

Labs are graded by the student teaching assistant using the **lab's scoring rubric** [PDF].

Pair programming teams will receive the scoring rubric sheet at the start of lab. Write both names on the sheet to turn in the sheet in when you finish. Name 1 should be the student who saved the pair work in their CS 250 account.

- If you **finish during lab**, turn in to your grader the **Lab's scoring rubric sheet.**
    - If you have **not** yet satisfied all criteria to the level of 4, you need to continue working on the lab outside of class time
    - Your saved solution should be stored in the CS 250 account listed under Name 1
    - Your solution will be checked by the grader using the scoring rubric and both students will receive the same score for the lab assignment
- If you are **unable** to complete the solution during the lab period
    - Leave lab with both students having a copy of the partially completed solution
    - Agree to finish the lab together (establish a time) or independently
    - Turn in the **lab's scoring rubric** [PDF] at the **start of next Lab**
        - Use one rubric for teams finishing together or two rubrics for students finishing independently
        - Keep track of and include the **total completion time** (rounded to closest half-hour) it took to complete the lab assignment and include the time on the rubric
        - Write a score `0..4` in the rubric's **self assessment column** representing your completion status
    - Make sure your work is stored in your CS 250 account in folder `U:\htdocs\labs\lab9\`

Lab 9 is due at the start of lab of the next Lab. **No** late lab assignments will be accepted without prior approval. Your lowest lab score will be dropped.