# ICA15: Single Page Architecture – Part 2

In this activity we add error trapping and an improved Bootstrap interface to our “inspirational quotes” application.

Make note of the following URLs.

|  |  |
| --- | --- |
| **URL** | **Notes** |
| <https://misdemo.temple.edu/dailyquote>  | GET from this URL to get a JSON response containing a fun and/or inspirational quote. Each GET requests results in a new/different quote.  |
| <https://textbelt.com/text>  | POST to this URL to send a text message. You must pass “key” and “message” and “phone” as URL encoded data. This API feature returns confirmation data in a JSON array. |
| <https://misdemo.temple.edu/auth>  | GET from this URL and send a “username” and a “password” as URL Encoded data. Each GET request results in either a JSON array of data or an error message in a string of text.One of several good username / password combinations is: “joe” and “abc123” |

## Instructions

1. Start with ica15\_quotes.zip. Set up your project in VS code.
2. Explore the readme folder. This folder contains some usernames and password that work with the “auth” API in the above table.
3. In the <script> portion of the page, find this line:

|  |
| --- |
| // this command runs when the page is ready.... $("#div-sample").show(); |

This line of code controls which of the 8 span columns gets displayed first. Try changing it to div-quote, div-sms, and/or div-login. Reload the page after each save and observe the change.

You should also note that each 8-span column is hidden by default with the in-line CSS statement “display:none;” That means that the only way these will appear after page load is if we use JavaScript / jQuery to show them!

1. Find the click event for btn\_login. A few lines of code have already been provided there. These lines of code wipe out any content in the login\_message div, remove any CSS classes applied to the login\_message div, and create variables that will hold user input.

Add logic to the click event handler so that if username or password is blank, an error message gets put into login\_message, and login\_message gets the two CSS classes “alert” and “alert-danger”.

Otherwise, call the supporting function named “login\_controller”.

1. In the login controller, add logic so that if the data returned by the API call is of the datatype string, then the string is put into the login\_message div. Again, give the login\_message div the CSS classes “alert” and “alert-danger”.

Otherwise, advance the user to the next interface by hiding the tag with the id “div-login” and showing the div tag with the id “div-quote”.

To perform this operation, you will need to used JavaScript’s typeof operator. This is the first and last time you will need this operator in this course!

1. Now, in the click handler for btn\_send\_text … can you do the same thing?
	1. Check to make sure that the user has added text to the input tag “message”
	2. Check to make sure that the user has added text to the input tag “phone”
	3. If the user has not provided both these items, then put an error message into the confirmation div.
	4. If they have added text, call the supporting function named “send\_the\_text”.
2. The logout link does not work. Fix this. Add a click event handler that will show and hide div tags appropriately when the logout link is clicked. You need to hide \*both\* the div tags that might be visible at the time the user clicks “Logout” and then show the one tag you want to see.
3. Test your work.
4. Here’s a thought: is it necessary to show the input tag that contains the “message”? Change the input type of that tag to hidden and delete the corresponding label in the HTML.
5. Test again.
6. Upload your ica15\_quotes folder to misdemo, and make sure that you can determine the URL for your work. Use the folder, so you do not have to specify “index.html” as part of your URL.

For example: https://misdemo.temple.edu/tuz54321/ica15\_quotes

***You need to upload the whole folder, and all its contents. Not just the index.html file.***

1. On canvas, turn in the URL to ica15\_quotes.