# MIS2402 Eliza SPA Demo

In this exercise, we will use an API endpoint found here: <https://misdemo.temple.edu/eliza/>

NOTE: The logic behind this API is based on a program named “Eliza” written in 1966. A goal of the original Eliza program was to demonstrate how language could be parsed to simulate a conversation between a patient and a psychotherapist.

If you are interested in the original Eliza code, you can read more about it here <https://en.wikipedia.org/wiki/ELIZA>.

Take a moment and click on that URL and see what happens. Also try the following resources:

<https://misdemo.temple.edu/eliza/greeting>

<https://misdemo.temple.edu/eliza/greetings>

<https://misdemo.temple.edu/eliza/remark>

<https://misdemo.temple.edu/eliza/history>

Here is the API documentation ***again***. It’s the same as what’s found here: <https://misdemo.temple.edu/eliza/>

|  |  |
| --- | --- |
| https://misdemo.temple.edu/eliza/greeting  | Issue a GET against ./greeting to get a random greeting. |
| https://misdemo.temple.edu/eliza/greetings  | Issue a GET against ./greetings to get all the greetings. |
| https://misdemo.temple.edu/eliza/remark | Issue a POST against ./remark with 'input' to get an Eliza response. You must also send the keys: studentname and userid. The response will be a JSON array containing 'success' and 'remark'  |
| https://misdemo.temple.edu/eliza/history  | Issue a GET against ./history and provide a studentname and userid. The result will be a JSON response representing the chat history that corresponds to that student and user. |

## Instructions

This Web Service can be used to construct a simple “Chat with Eliza” application. You can think of Eliza and the great-great-great-grandmother of ChatGPT.

This demo is ***NOT GRADED.*** It’s a good way to prepare for the exam though.

1. Start by downloading eliza\_spa\_demo.zip and setting up your work in Visual Studio Code.
2. There’s a modest amount of HTML work for you to do. You should be able to:
	1. Change the title tag from “MIS2402 Template” to “MIS2402 - Chat with Eliza”.
	2. Change the text near the logo from “MIS2402” to “MIS2402 - Eliza”.
	3. Change the name of the student in the footer from “Noname McHiggens” to your name.
	4. Change the value of the hidden tag with the name “studentname” from “Noname McHiggens” to your name.

In the above steps, the last step, step D, is especially important. The first three changes are cosmetic changes. The last change will alter the data sent to your web service.

1. Change the start up behavior of your code, so that div-login is shown on page load, instead of div-sample.
2. There are three discussion points/questions in this code. See:
	1. The discussion question in the login controller. Why would it be important to assign a value to a hidden input tag here?
	2. The discussion question in the history button’s click-event handler. Other click event handlers have error trapping. Why not this one?
	3. The discussion question in the “back” button’s click-event handler. Other click event handlers make calls to supporting functions. Why not this one?
3. Experiment with the application. There’s enough functionality in the code that you can interact with the Eliza web service.
	1. Explore the code … can you trace the sequence of events from clicking the send button to getting a response?
	2. Did you notice that the HTML tag with the id your\_text has both a **name** and also an **id** attribute? Those attributes have different values. Why is that important?
	3. Why is the web service’s remark feature a POST and not a GET?
4. Now it’s your turn. Complete the history controller. It should call the history feature of the web service and append rows into the table with the id of table-history.
5. No upload is required here. But wouldn’t that be a good thing for you to practice?