Sure, I'll explain the code step by step in simple terms:

1. `(function() { ... })();`: This is an immediately-invoked function expression (IIFE). It's used to create a private scope for the code inside so that variables declared inside the function don't interfere with the global scope.

2. `const form = document.querySelector("#message-form");`: This line selects the HTML element with the ID "message-form" and assigns it to the variable `form`. The `document.querySelector()` method is used to find the first element that matches the provided CSS selector.

3. `form.addEventListener('submit', function(e) { ... });`: This line adds an event listener to the `form` element. It listens for the "submit" event, which occurs when the user submits the form (e.g., by pressing Enter or clicking a submit button). When the event is triggered, the function inside the second argument will be executed.

4. `e.preventDefault();`: This line prevents the default behavior of the form submission. It stops the form from being submitted in the traditional way (i.e., sending data to the server and reloading the page).

5. `const message = document.querySelector('#message');`: This line selects the HTML element with the ID "message" and assigns it to the variable `message`. This element is typically an input field where the user can enter their message.

6. `const feedback = document.querySelector('.feedback');`: This line selects the HTML element with the class "feedback" and assigns it to the variable `feedback`. This element is likely a container used to display feedback or error messages to the user.

7. `const messageContent = document.querySelector('.message-content');`: This line selects the HTML element with the class "message-content" and assigns it to the variable `messageContent`. This element is probably a container where the submitted message will be displayed.

8. `if (message.value === '') { ... }`: This is a conditional statement that checks if the `message` input field has no value (i.e., if it's empty).

9. `feedback.classList.add('show');`: If the `message` input is empty, this line adds the class "show" to the `feedback` element. This class is likely used to display the feedback or error message to the user.

10. `setTimeout(function() { feedback.classList.remove('show'); }, 4000);`: This line sets a timer using `setTimeout()` that removes the "show" class from the `feedback` element after 4 seconds (4000 milliseconds). This gives the user a brief visual indication of the error before it disappears.

11. `messageContent.textContent = message.value;`: If the `message` input is not empty, this line sets the content of the `messageContent` element to the value of the `message` input. It effectively displays the submitted message in the designated area.

12. `message.value = '';`: This line resets the `message` input field to an empty value after the message has been processed.

To summarize, the code is an event listener attached to a form. When the form is submitted, the code checks if the message input field is empty. If it's empty, it shows a feedback message briefly. If the message input has content, it displays the message in a designated area and clears the input field. The IIFE ensures that the code runs immediately when the script is loaded.