



Functions

Overview

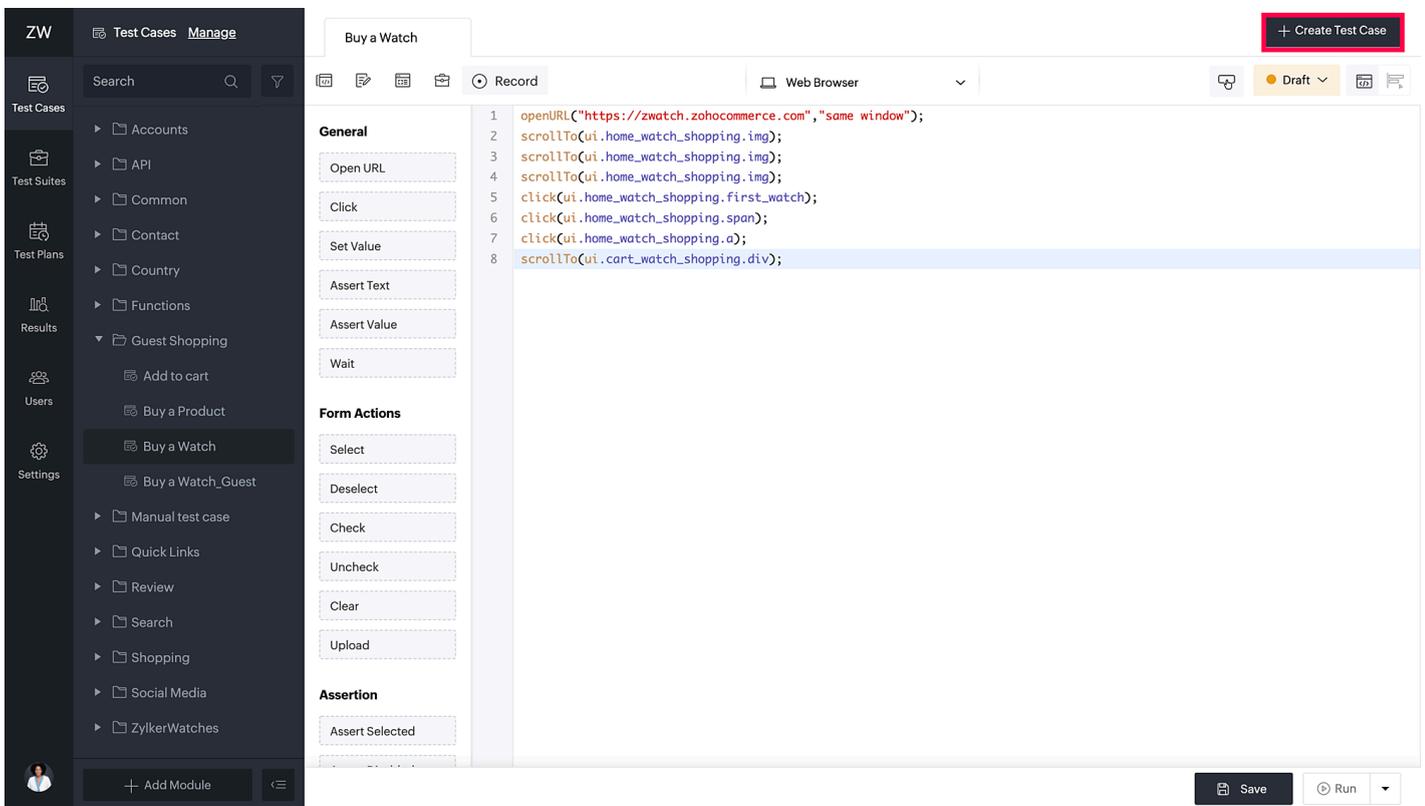
Functions are [test cases](#) that can be reused for a specific task and can be called multiple times throughout a test case with or without different arguments. For example, take the case of an ecommerce shop. Test cases like *Add to Cart* are common actions inside their web services and mobile applications. These test cases are repeated actions and can be listed under the functions category.

Functions can be referred to in test cases using its **Link Name** or using the task [call function](#).

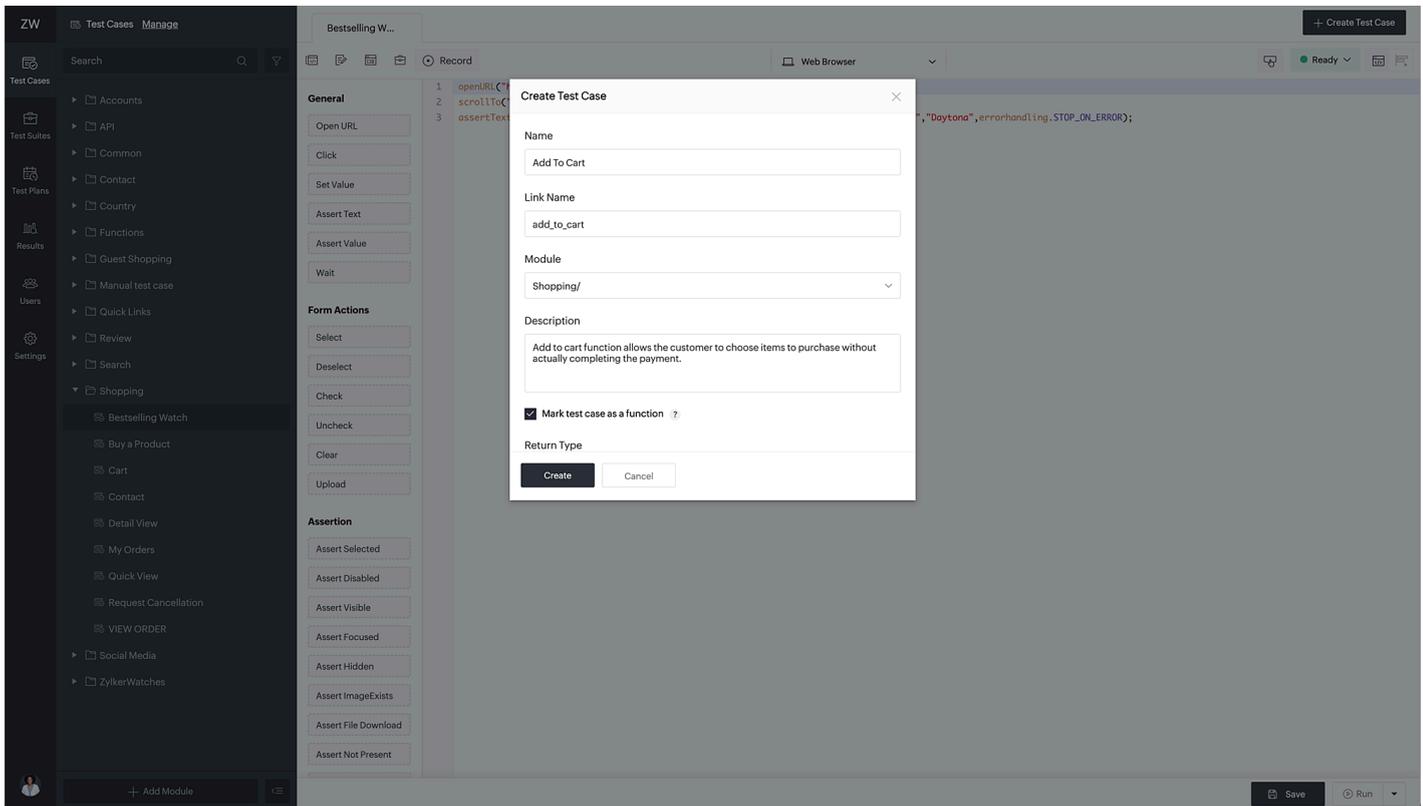
Create Function

Creating a function is similar to a creating a test case. To do so:

1. Click **Create Test Case**.



2. Enter the test case details, such as *Name*, *Link Name*, and *Module* to which it needs to be added. Additionally, providing a *Description* offers a better overview of the test case, but is not mandatory.



The link name will be unique for each function and will be auto-filled by default, similar to the test case name. They should be descriptive and accurately reflect the purpose of the function. For example, if your test case name is **Contact Us**, the link name would be `contact_us` by default. However, they can be edited as per the following conditions:

- The link name only supports lower case letters. For example, if the test case name is `Addtocart`, the link name would be `addtocart`.
- Special characters are not supported. They will be replaced with underscores (`_`). For example, if the test case name is `Replacement&Return`, the link name would be `replacement_return`.
- If there are two or more special characters consecutively, they will be replaced with a single underscore character. For example, if the test case name is `details & @contact`, the link name would be `details_contact`.

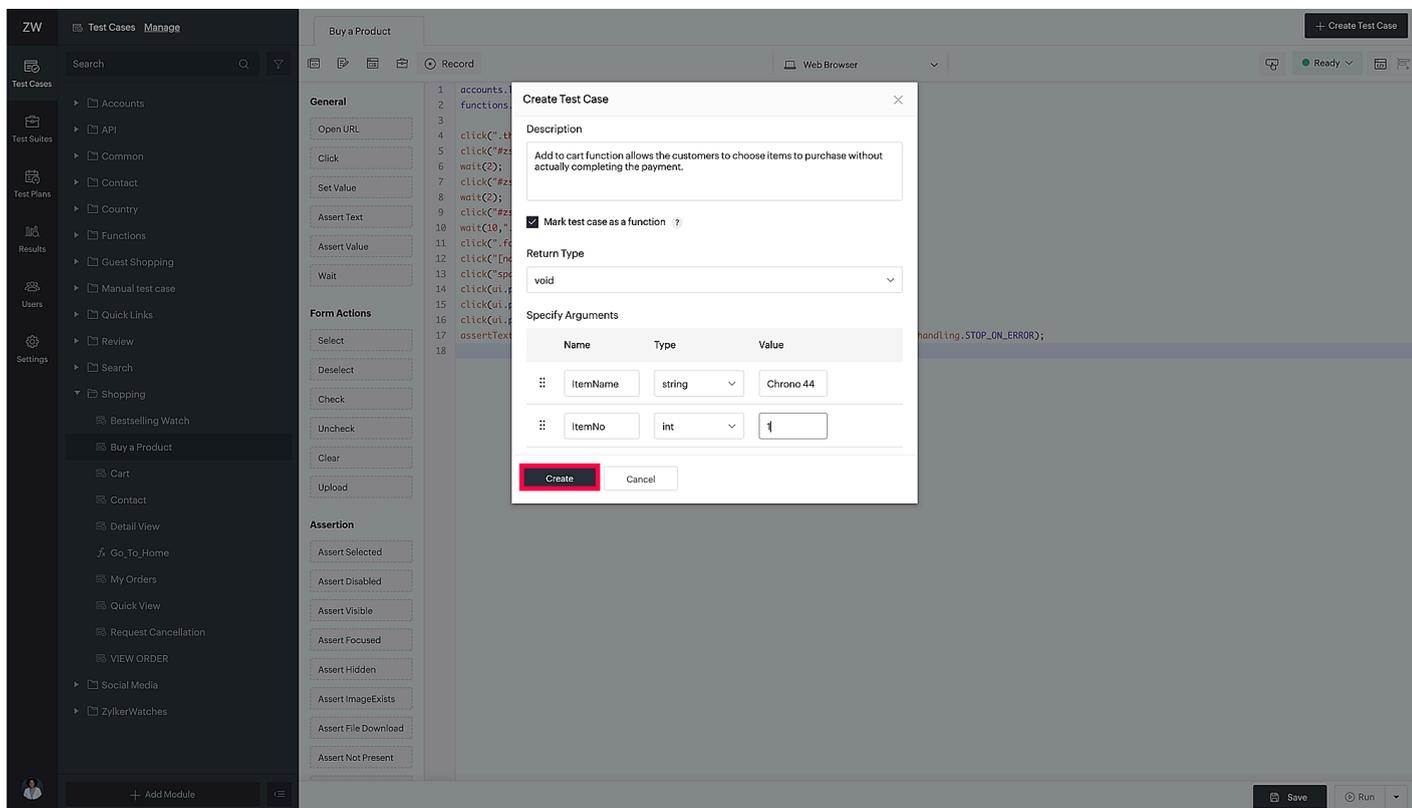
3. Select the **Mark test case as a function** checkbox below the description to convert the test case into a function.

 **Note:** If the test case isn't marked a function, then it's considered as a regular test case.

4. Mention the return type of the function. This specifies the data type of the value that the function will return after its execution. For example, if we create a function to greet every user inside an ecommerce site, then the function is a void type, as it doesn't have a return statement. It will simply print a greeting message for

every user. However, if the function was created to sum the total price of a product and the price value are integer data types, then the function is an integer type as it gives an integer sum in return.

5. Specify the arguments, if any. The arguments of a function are the input values passed when it is called inside a test case or another function, which the function can use to perform its operations. For example, an ecommerce function to show featured products doesn't take arguments but displays products, while a function calculating the total price takes quantity and price per unit arguments and returns the total cost.

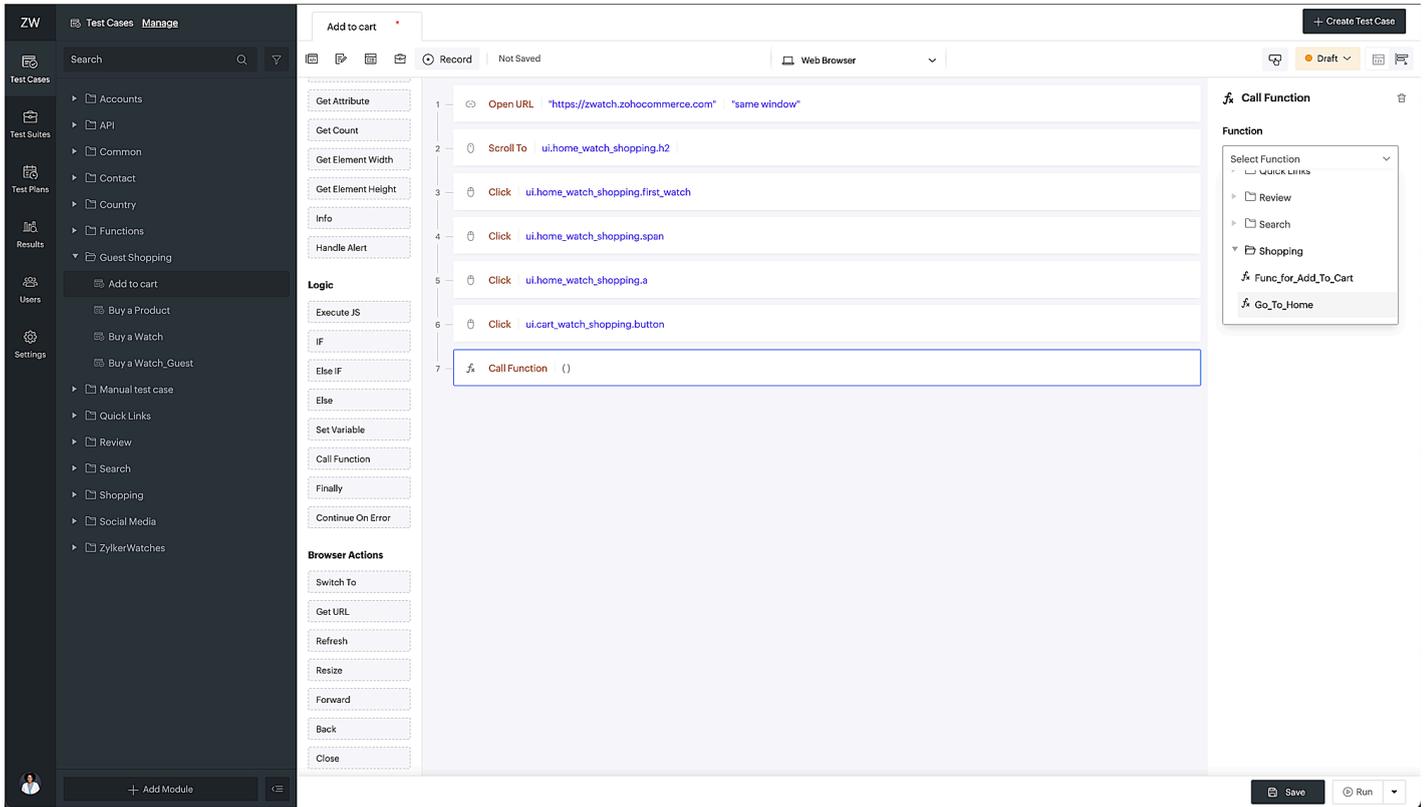


 **Note:** The fields return type and specify arguments will only be enabled once you check the box to make the test case a function.

6. Once the required details are filled, click **Create**.

Using functions

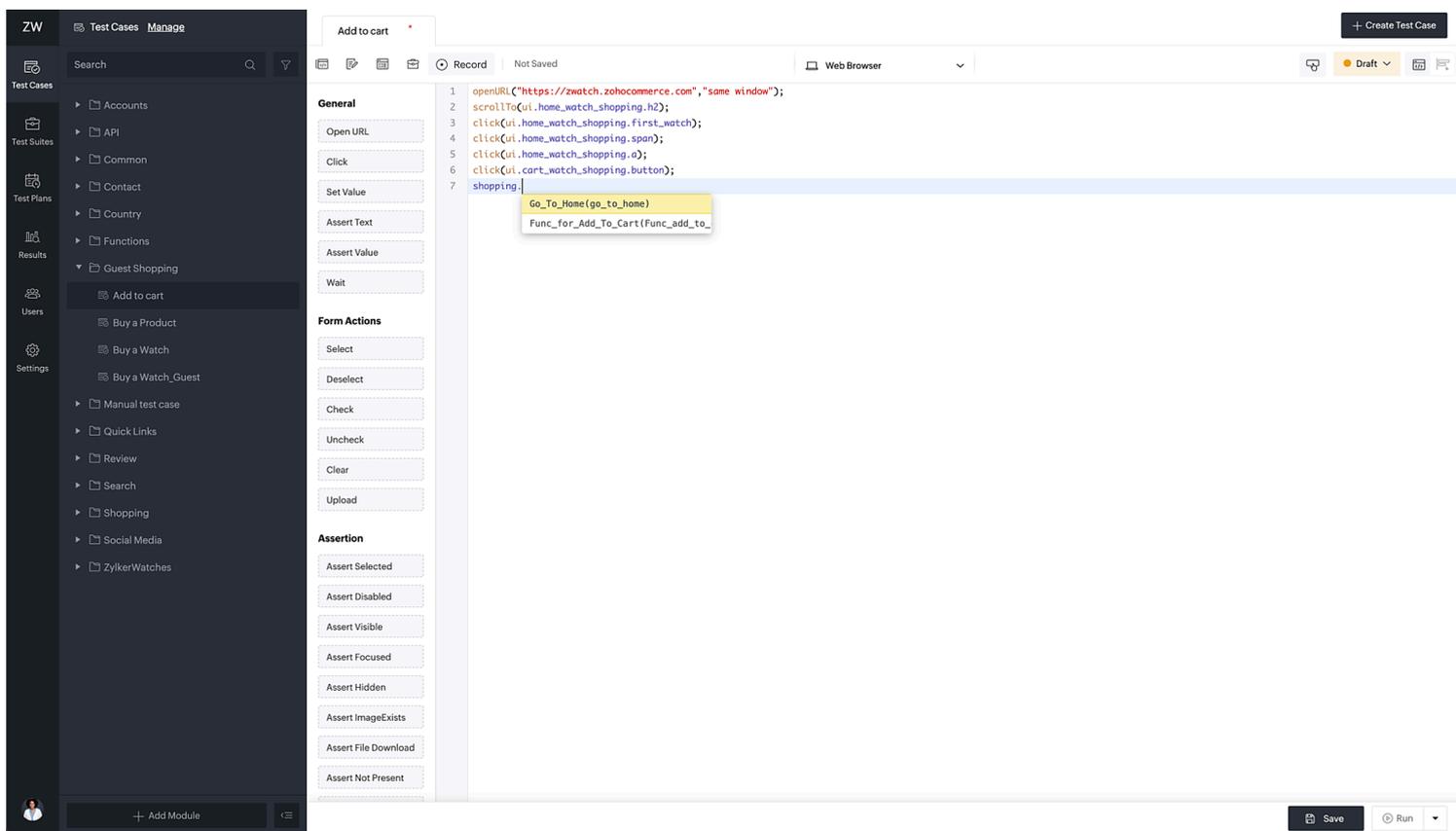
- To use functions in a test case or an existing function inside builder mode, simply call the function using the [Call Function](#) task. Drag-and-drop the **call function** task from the task panel to the editor. The right panel for the arguments will have a drop-down menu. Choose the module from the drop-down menu to select the desired function.



- For editor mode, simply refer to it inside using the format: **<module_link name>.<function_link name> (arguments);**

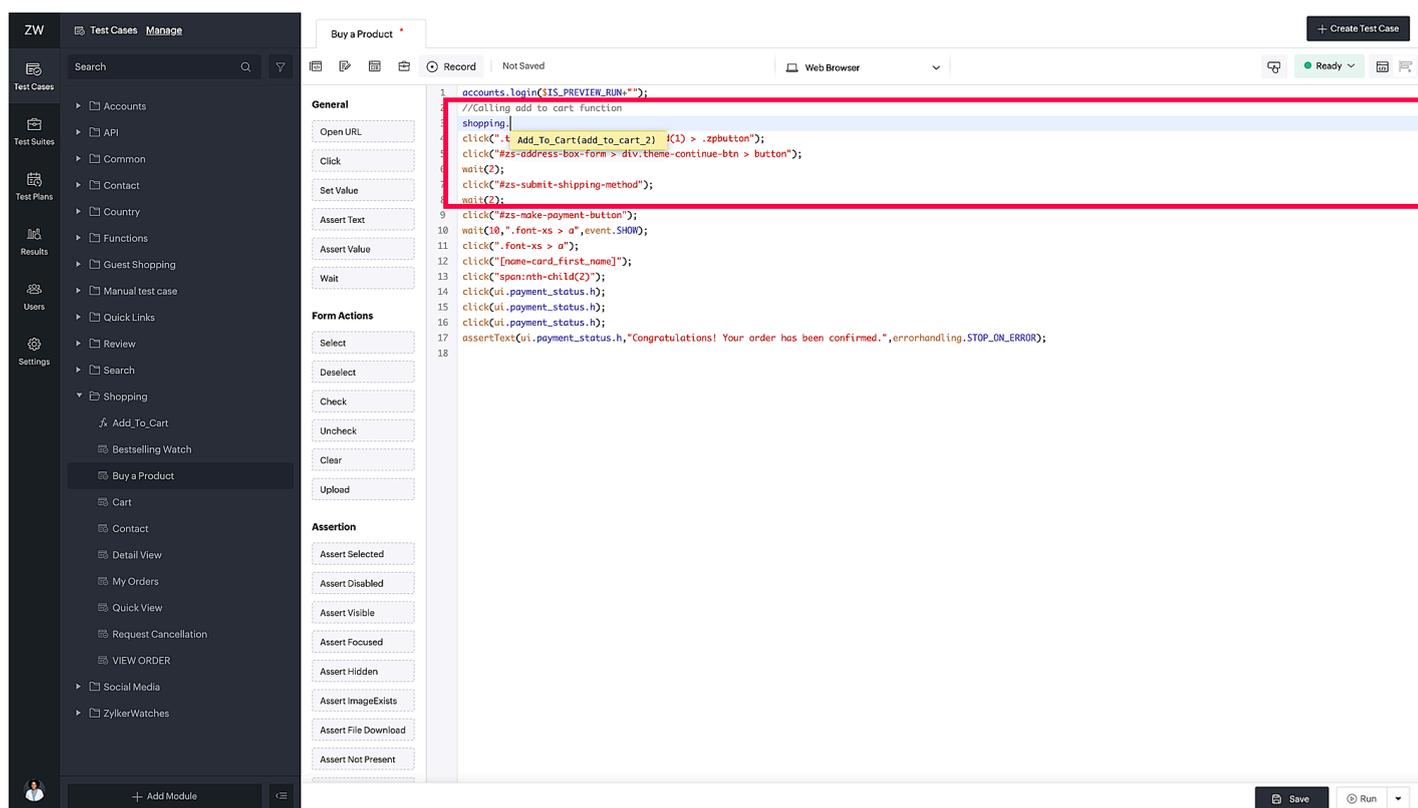
Parameter	Description
module_link_name	Link name of the module in which the function has been defined
function_link name	Name of the function to be invoked
arguments	Parameters of the function, if any

For example, if the module name is *Shopping* and the test case name is *Add to Cart*, in this case will be *shopping.add_to_cart()*;



Let's look at an example to understand how functions are used inside test cases. Take the case of the *Add to Cart* function of the website Zylker Watches. The tester created a new function to handle the Add to Cart functionality in an ecommerce web application, in order to implement the function inside the test case Buy a Product.

1. The test case belongs to the Shopping module. The Add to Cart function is called inside the Buy a Product test case. This automatically adds the function's script in the test case, as displayed below:



Understanding the use of functions is crucial. Now that we have learned how to create functions, the next step is to learn how to use them. Choose between [recorder](#), or [editor and builder](#) mode to write the functions. Execute and analyze the functions with the preview run options: [Web](#), Android, and iOS, for more reliable and reusable function scripts.

Related Links

- [Web Testing using Builder and Editor mode](#)