# Given resources:

1. A red circle with a cross

   AI-generated content may be incorrect. index.html
2. A red circle with a cross

   AI-generated content may be incorrect. test.js
3. A pencil with a red eraser

   AI-generated content may be incorrect. script.js
4. A red circle with a cross

   AI-generated content may be incorrect. All files in subfolder “api/”

**Important:**

* You can only edit script.js
* There are PHP files in subfolder “api/”.
  + Place all the files in your WAMP/MAMP web root.
  + Access via your WAMP/MAMP.

# Instructions:

Refer to the [appendix](#_Appendix:_API_documentation) for the documentation of the given Products API (**./api/products.php**).

You are to complete the following functions in **script.js** such that index.html works as described below.

1. **function populateCategories()**
   1. Invoke the appropriate Products API endpoint to retrieve and populate the “product category” dropdown list – select element with id "category".
   2. This function is called once upon page load completion.
2. **function updateProductTable()**
   1. Invoke the appropriate Products API endpoint to retrieve products of the category selected in the “product category” dropdown list
   2. Display the category’s products in a HTML table in the div element with id “output”.
   3. A sample code of the desired HTML table with Bootstrap CSS classes for styling is given in index.html.
   4. When there is no category selected, the div element should have only 1 line “Select a product category”, i.e. no HTML table.
3. **function doAdd()**
   1. Invoke the appropriate Products API endpoint to create a new product as specified by the user in the 2 text fields for the current selected category in the dropdown.
   2. You may assume the user will always provide valid inputs. Form fields validation is not required.
   3. Upon successful creation, clear the 2 text fields and update the table of products by reusing the function updateProductTable().

You may add more functions in **script.js**.

If you have done the above correctly, index.html should work as described below:

1. Initial loading   
   A screenshot of a computer

   AI-generated content may be incorrect.  
   1. The “product category” dropdown list  
      A blue and white rectangle with a blue stripe

      AI-generated content may be incorrect.
   2. The “Test area” is for running tests to check your solution. More on it later.
2. Select “fruits”  
   A screenshot of a computer

   AI-generated content may be incorrect.
3. Select “drinks”  
   A screenshot of a computer

   AI-generated content may be incorrect.
4. Add “Orange Juice” 2.20 to category “drinks”  
   A screenshot of a computer

   AI-generated content may be incorrect.
   1. After clicking “Add” button  
      A screenshot of a computer

      AI-generated content may be incorrect.

# Test area A screenshot of a computer AI-generated content may be incorrect.

* Click “Reset data” button to restore data to starting state.
* Click “Run tests” button to check your solution.
  + It will take a few seconds for the tests to complete.
  + You can view the tests’ details on the web page or the browser console.
  + A screenshot of a test

    AI-generated content may be incorrect.

Important: For grading,

* The test values may be modified, and new test values may be added.
* The products API may have different categories and the products’ names & prices may change.

# Appendix: API documentation of ./api/products.php

**1. GET /products.php**

Returns all products organized by category.

**Example Response:**

Status Code: 200 OK

|  |
| --- |
| {  "fruits": [  {"name": "Pack of 4 apples", "price": 3.95},  {"name": "Pack of 3 oranges", "price": 5.15}  ],  "drinks": [  {"name": "100 Plus", "price": 1.5},  {"name": "Coke Zero", "price": 1.8},  {"name": "Coke", "price": 1.7}  ]  } |

**2. GET /products.php?category={categoryName}**

Returns all products in a specific category.

**Parameters:**

* category (query parameter, string, required) – The category name

**Example Request:**

GET /products.php?category=fruits

**Example Success Response:**

Status Code: 200 OK (if category exists)

|  |
| --- |
| [  {"name": "Pack of 4 apples", "price": 3.95},  {"name": "Pack of 3 oranges", "price": 5.15}  ] |

**Example Error Response:**

Status Code: 404 Not Found (if category doesn't exist)

|  |
| --- |
| {"error": "Unknown category"} |

**3. POST /products.php**

Adds a new product to an existing category.

**Request Schema:**

|  |
| --- |
| {  "category": "string",  "name": "string",  "price": "number"  } |

**Example Request:**

|  |
| --- |
| {  "category": "fruits",  "name": "Grapes",  "price": 6.45  } |

**Example Success Response:**

Status Code: 201 Created (if successful)

|  |
| --- |
| {  "message" "Product added successfully"  } |

**Example Error Response:**

Status Code: 404 Not Found (if category doesn't exist)

|  |
| --- |
| {  "error": "Unknown category"  } |