

Task 1

- ❑ Create account and products resource on mockAPI website : <https://mockapi.io/>
- ❑ Create schema and generate resources for products.
- ❑ The schema consists of: id, title, description, price, image.
- ❑ In the code:-
- ❑ Create a function to fetch all products from an API endpoint and display them in a table in the html document.
- ❑ Create button to add new product, when click the button show a form contain input for title, textarea for description, input to price and url input to image and save button to submit the form data to the api endpoint.

Task 1 *cont*

- ❑ In the table of products where you are show the fetched products from api
- ❑ Create at the end of each row 3 buttons (show, edit, delete) buttons.
- ❑ Show button display the product of this row.
- ❑ Edit button show form for edit the product from it.
- ❑ Delete button show alert to confirm delete product, if the user confirmed delete the product.
- ❑ For those requests create function to add product, edit product, delete product by using API.



Task 1 *cont*

1) Show all product

- Details:
- Image: Displays a thumbnail (e.g. 50x50px) from the image URL.
- Title, Description, Price: Text content, trimmed if too long.
- Actions:
- 👁 View — opens a modal with full product info
- ✎ Edit — opens a pre-filled modal
- 🗑 Delete — prompts for confirmation before deleting

Add product


Example Table Columns:

Image	Title	Description	Price	Actions
	Product A	A great item	\$10.00	[👁 View] [✎ Edit] [🗑 Delete]
	Product B	Another great one	\$25.00	[👁 View] [✎ Edit] [🗑 Delete]

Task 1 *cont*

- 2. Add Product Button
- Located at the top (typically top-right).
- Opens a modal or form panel to enter:
- Title (text input)
- Description (textarea)
- Price (number input or text input with validation)
- Image URL (url input)
- On submit: Sends a POST request to the API → refreshes table.

Task 1 *cont*

- 2. Add Product Button
- Located at the top (typically top-right).
- Opens a modal or form panel to enter:
- Title (text input)
- Description (textarea)
- Price (number input or text input with validation)
- Image URL (url input)
- On submit: Sends a POST request to the API → refreshes table.
-  Add Product Modal
- All fields except id (auto-generated).
- Validates inputs (e.g., price must be numeric, image must be valid URL).

Task 1 *cont*

3. View Product Modal

- Read-only view showing:
- Full-size product image
- Title
- Description
- Price

4. Edit Product Modal

- Pre-filled with existing values.
- On submit, sends PUT /api/products/:id.

5. Delete button: prompts for confirmation before deleting