1. for cheqing whether the data is loading from API or not use----console.log() after the load function in js file
2. for displaying each of the multiple Data first cheque by using forEach function and console.log into display function in js file
3. for dynamically displaying individual API data we use 4 step process—
4. access the container by its id where all the dynamically created html tags will be displayed from HTML file.
5. create a div tag into js file where all the new dynamically created will be stored.
6. create required class or tags of html into this <div>
7. set all the crated tags into this container by its id from HTML file.
8. After completing all the process in JS file delete all the daisy UI card info from HTML file .
9. now setting each of the image dynamically from api by ${tool.image}
10. setting each of the description dynamically from api by ${tool.features}
11. for arranging all these data use grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4 class In HTML file
12. set the features 1 2 3 by ${tools.features[0]}, ${tools.features[1], ${tools.features[2]}
13. remove w-96 as it makes the width fixed from js file

10.add mb-4 class in html file (sort by button)

11. add container and mx-auto class into the main tag in HTML file to set automatic width for different type of device

12. set some tailwind css class in the card section of js file for styling

**13.** for search section **------collect from daisy ui** search button and search area **and put it into a new section in HTML file**

**14. for searching** add the parameter inside the load **function in JS file---const loadPhone=async(searchText)**

**and add the new api for search item**

**const res=await fetch(`https://openapi.programming-hero.com/api/phones?search=${searchText}`);**

14. add a onclick and handleSearch() function to search button in HTML file and in js file

15. add an id into search field area in HTML file and access by that id into JS files function area

16. searchField.value is used to read the input from search field area

17. after every search the previously loaded phone will be clear—phoneContainer.textContnet=’’;

modal operation using a button

1. **copy the responsive modal card from** daisy uI and set it into HTML file in another section.

**2. change the button function name in the card as** show-details-modal.showModal() function and set it into the JS file showToolsFunction() arrow function.

**3.** creat **two arrow-modal-function in the js file and** add parameter from api(id)**--- const loadToolsModal & showToosModal=()=>{ }**

4. dynamically set the api each details data by **button onclick=”loadToolsModal (‘${tool.id}’)** in js card

5. **add an ID with<p> tag** into HTML file modal and **use this id into showToolsModal()** arrow function in JS file

6. **add an <div> tag with ID (show-details-container)**into HTML files modal and use this id into JS file **showToolsModal()** arrow function