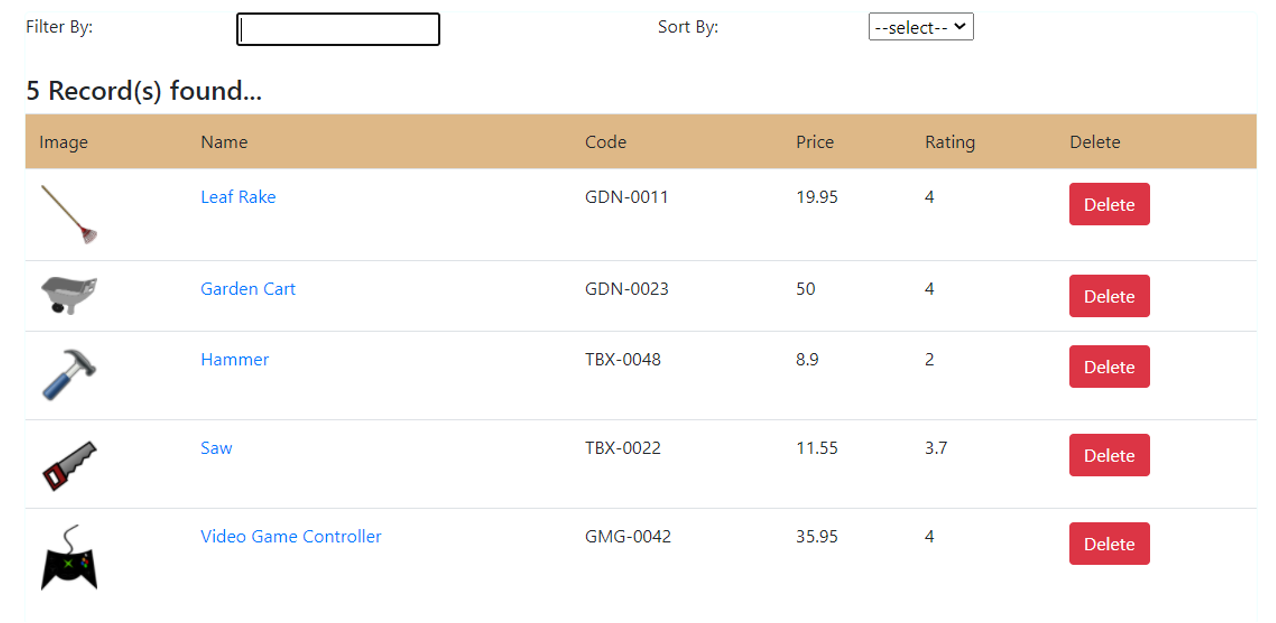
1. Create a web application using React JS which will display certain product records (refer the screenshot)

**Objective**: Use React JS concepts: Lifecycle, using asynchronous calls, using Promises, handling API calls from React JS component lifecycle methods and react hooks

**Capability Code**: WEBR0022 [Creating asynchronous and event-based programs]



Description:

1. Refer the same application that you completed as assignment earlier (*refer word document: ‘Assignment\_WebTech\_MERNStack\_WEB0017\_ReactJS.docx’*):
   1. You have to add two more functionalities:
      1. Sorting: Products in the table must be sorted based on the choice/selection made from the drop-down list present on the right side of the view. Products must be sorted by default by ‘Product Id’. Sorting options are: Name, Price, Rating, Code and Id
      2. Filtering: The products must be filtered by the text entered (product name) in the input box on the left side. The text entered could be a partial name or full name of the product. Only those products, whose name contains that text should be displayed in the table, otherwise all the products should be displayed as usual
   2. *Using Ref*: the filter text box must have focus. Instead of using callback ref, use createRef() method of React JS
   3. *Making HTTP requests:* Instead of using local array of products, now the application must fetch the data from the file of product records, hosted by JSON-server, by making HTTP request. The code for doing that must be written in the methods of the same service class written earlier, to do operations against the array
   4. *Handling asynchronous requests and promises:* The data returned by the HTTP request must be handled asynchronously in the component lifecycle methods
2. Create a web application using React JS which will display certain product records (refer the screenshot) using React Hooks

*Using React Hooks:* Use the previously completed application. Instead of using class components with lifecycle methods and state property, convert all those class components into functional components and use corresponding hooks for lifecycle events and state management in the component