

## SVU Day-16 Assignment

04/03/2025

### Assignment: Fetch & Display Products and Users with Sorting, Filtering & Searching

---

#### Overview:

In this assignment, you will build a multi-page website that fetches data from two APIs and displays them dynamically. The website will include a **Home page**, a **Products page**, and a **User's page** with sorting, filtering, and a search bar using **debouncing**. Clicking on an item will navigate to a **detailed page** displaying all relevant information about the selected product or user.

---

#### Features to Implement:

##### 1 Pages to Create:

- ✓ index.html - **Products Listing Page (Default Page)**
  - ✓ users.html - **Users Listing Page**
  - ✓ product-detail.html - **Product Detail Page**
  - ✓ user-detail.html - **User Detail Page**
  - ✓ styles.css - **CSS Styling for UI**
  - ✓ script.js - **JavaScript for API Calls & Functionality**
- 

##### 2 Features in index.html (Products Page - Default Page):

###### ✓ Fetch & Display Products Dynamically:

- Fetch product data from <https://fakestoreapi.com/products>
- Display products in a **grid format** with the following details:
  - Image
  - Name
  - Price
  - Category
  - Rating
  - "View Details" button

###### ✓ Sorting & Filtering:

- **Sort products** by Price (Low to High, High to Low) and Rating.
- **Filter products** by Category (e.g., Electronics, Clothing, Jewellery).

✓ **Search Bar with Debouncing:**

- Users should be able to search for products by name.
- Implement **debouncing** (300ms delay) for performance optimization.

✓ **Navigation Bar:**

- The navbar should have **Home**, **Products**, and **Users** options.
- Clicking **Products** should load index.html.
- Clicking **Users** should load users.html.

✓ **Product Details Page:**

- Clicking on a product should redirect to product-detail.html, showing full details.
- 

### 3 Features in users.html (Users Listing Page):

✓ **Fetch & Display Users Dynamically:**

- Fetch user data from <https://fakestoreapi.com/users>.
- Display users in a **grid format** with the following details:
  - Name
  - Username
  - Email
  - Address (City, Street)
  - "View Details" button

✓ **Sorting & Filtering:**

- **Sort users** by Name (A-Z, Z-A).
- **Filter users** by Location (Filter users based on City).

✓ **Search Bar with Debouncing:**

- Users should be able to search for users by name or email.
- Implement **debouncing** (300ms delay).

✓ **User Details Page:**

- Clicking on a user should redirect to user-detail.html, showing all details.
- 

### 4 Features in product-detail.html (Product Details Page):

- ✅ Retrieve the selected **Product ID** from localStorage or URL parameters.
  - ✅ Fetch full product details using <https://fakestoreapi.com/products/{id}>.
  - ✅ Display the following details:
    - Image
    - Title
    - Description
    - Price
    - Category
    - Rating
  - ✅ Provide a "**Back to Products**" button to navigate back to index.html.
- 

### 5 Features in user-detail.html (User Details Page):

- ✅ Retrieve the selected **User ID** from localStorage or URL parameters.
  - ✅ Fetch full user details using <https://fakestoreapi.com/users/{id}>.
  - ✅ Display the following details:
    - Name
    - Username
    - Email
    - Phone Number
    - Address (Street, Suite, City, Zip code)
    - Company Name
  - ✅ Provide a "**Back to Users**" button to navigate back to users.html.
- 

### 📌 6 Additional Enhancements:

- ✅ **Loading Spinner** while fetching data.
  - ✅ **Mobile-Friendly UI** using CSS **media queries**.
  - ✅ **Dark Mode Toggle** for UI customization.
  - ✅ **Persist Search & Filter Preferences** using local Storage.
  - ✅ **Error Handling**: Show an error message if API fails to load data.
- 

### 🎯 Bonus Challenges:

- ◆ **Pagination**: Implement pagination to load limited products or users per page.
- ◆ **Favourite Feature**: Allow users to mark a product or user as a favourite using local Storage.
- ◆ **Infinite Scroll**: Load more products/users as the user scrolls down the page.



### Submission Guidelines:

- ✚ Submit the **entire project folder** with all files and images.
- ✚ Ensure proper **code structure and comments** explaining key functionalities.
- ✚ Make sure your website is **fully responsive**.



**Happy Coding!** 🎉