Pagination Exercise Lab:

**Objective:** Implement pagination for a list of student items displayed on a webpage.

**Step 1: Set up the HTML Structure**

- Create an HTML file with the necessary structure.
- Include a **<ul>** element with the class **student-list** to hold the list of student items.
- Add a **<div>** element with the class **pagination** to hold the pagination links.

**Step 2: Fetch Elements and Define Variables**

- Use JavaScript to select the **.student-list** element and store it in a variable named **studentlist**.
- Use **studentlist.children** to get all the student items and store them in a variable named **studentItem**.
- Define a constant **perPage** to specify the number of items to display per page.
- Initialize a variable **page** to keep track of the current page number, starting from 1.
-

**Step 3: Define the showPage Function**

- Define a function named **showPage** that takes two parameters: **list** (the list of student items) and **page** (the current page number).
- Loop through each student item in the **list**.
- If the index of the student item is within the range of the current page (**(page - 1) * perPage** to **page * perPage - 1**), set its **display** style property to **"block"**, else set it to **"none"**.

**Step 4: Call showPage to Display Initial Page**

- Call the **showPage** function with **studentItem** and **0** as arguments to display the first page of student items initially.

**See: script_1.js**

**Step 5: Define the appendPageLinks Function to show the buttons**

- Define a function named **appendPageLinks** that takes a parameter **list** (the list of student items).
- Calculate the total number of pages (**totalPages**) by dividing the length of the **list** by **perPage** and rounding up to the nearest integer using **Math.ceil**.
- Select the **.pagination** element and store it in a variable named **ItemList**.
- Loop from **1** to **totalPages** to create pagination buttons.
- Create a new **<li>** element for each page and set its innerHTML to contain an anchor tag with the page number.
- Append each button to the **ItemList**.

**See: script_2.js**

### Step 6: Add Click Event Listeners to Pagination Buttons

- Select all the pagination buttons (**<a>** elements) and store them in a variable named **itembuttons**.
- Remove the **active** class from all pagination buttons.
- Add a click event listener to each pagination button.
- Inside the event listener, call the **showPage** function with the current page number as the argument.
- Add the **active** class to the clicked pagination button.

### Step 7: Call appendPageLinks to Generate Pagination

- Call the **appendPageLinks** function with **studentItem** as the argument to generate pagination links based on the list of student items.

**See: script_3.js**

### Step 8: Test the Pagination

- Open the HTML file in a web browser and test the pagination functionality by clicking on different page numbers.

### Step 9: Read from Local data file

- Data is given in **data.js**.
- Attach the dataset
- Create a **Render function** that implement html code for one user
- Create a **init function** that loop through the student list and render all items.
- Add a line to show the total number of users.

**See: script_2_1.js**