Responsive Image Gallery with Filters

Duration: 3hrs

Objective:

Create a fully functional, responsive image gallery using ReactJS. The gallery should display images fetched from a static JSON file, allow users to filter by categories, and include a modal for viewing images in larger detail with captions.

Requirements:

1. **Static JSON File for Images:**
   - Create a data.json file in your project that contains an array of image objects. Each object should have the following properties:
   - eg. json

     ```
     [
       {
         "id": 1,
         "url": "path_to_image1.jpg",
         "caption": "Beautiful Sunset",
         "category": "Nature"
       },
       {
         "id": 2,
         "url": "path_to_image2.jpg",
         "caption": "Modern Architecture",
         "category": "Architecture"
       },
       {
         "id": 3,
         "url": "path_to_image3.jpg",
         "caption": "Wild Animals in Forest",
         "category": "Animals"
       }
     ]
     ```

   - Include at least 15-20 images divided across categories like Nature, Architecture, and Animals.

1. **Gallery Layout:**
   - Display images in a grid layout.
   - Use CSS or Bootstrap to make the gallery responsive:

- Images should adjust their size based on the screen width.
- Show 4 images per row on large screens, 2 per row on tablets, and 1 per row on mobile devices.

2. **Category Filters:**
   - Add filter buttons for categories (e.g., "All", "Nature", "Architecture", "Animals").
   - Clicking a filter button should update the gallery to show only images from the selected category.
   - The "All" button should display all images.

3. **Modal for Viewing Images:**
   - When a user clicks on an image:
     - Open a modal that displays the image in a larger view.
     - Show the caption below the image inside the modal.
   - Include a close button to close the modal.

4. **Interactivity:**
   - Use React state to:
     - Store the list of images to be displayed.
     - Track the selected category.
     - Manage the open/close state of the modal.

5. **Responsive Design:**
   - Ensure the gallery and modal are mobile-friendly.
   - Use media queries or Bootstrap grid classes for responsiveness.

Additional Features (Optional):

- Add animations for:
  - Modal opening/closing.
  - Images appearing in the gallery after a category filter is applied.
- Highlight the selected filter button.
- Add a hover effect on images in the gallery (e.g., zoom or shadow).

Expected Output:

- A responsive image gallery that dynamically updates based on category filters.
- A modal that provides an immersive viewing experience for individual images.

Technologies:

- **ReactJS** for building the application.
- **CSS/Bootstrap** for responsive styling.
- Optional: **React-Bootstrap** for modal components.