📚 Guided Project: Fetch & Manage Books Data Asynchronously

Objective

Trainees will **fetch book data asynchronously** and apply JavaScript techniques such as:
✅ **Callbacks** to determine book properties dynamically.
✅ **Higher-order array methods** like `.map()`, `.filter()`, `.sort()`.
✅ **Event-driven logic** for UI-like interactions (e.g., showing a modal for special users).


```
[
  {
    "id": 1,
    "title": "To Kill a Mockingbird",
    "author": "Harper Lee",
    "genre": "Fiction",
    "year": 1960,
    "pages": 281,
    "publisher": "J.B. Lippincott & Co.",
    "description": "A novel about the serious issues of rape and racial inequality, told through the eyes of a young girl in the Deep South.",
    "image": "https://m.media-amazon.com/images/I/81gepf1eMqL.jpg"
  },
  {
    "id": 2,
    "title": "1984",
    "author": "George Orwell",
    "genre": "Dystopian",
    "year": 1949,
    "pages": 328,
    "publisher": "Secker & Warburg",
    "description": "A dystopian novel set in a totalitarian society ruled by the Party, which has total control over every aspect of people's lives.",
    "image": "https://m.media-amazon.com/images/I/71kxa1-0zfL.jpg"
  },
  {
    "id": 3,
    "title": "The Great Gatsby",
    "author": "F. Scott Fitzgerald",
```

```
    "genre": "Fiction",
    "year": 1925,
    "pages": 180,
    "publisher": "Charles Scribner's Sons",
    "description": "A story of the fabulously wealthy Jay Gatsby and his love for the beautiful Daisy Buchanan.",
    "image": "https://m.media-amazon.com/images/I/81af+MCATTL.jpg"
  },
  {
    "id": 4,
    "title": "Pride and Prejudice",
    "author": "Jane Austen",
    "genre": "Romance",
    "year": 1813,
    "pages": 279,
    "publisher": "T. Egerton, Whitehall",
    "description": "A romantic novel of manners that follows the character development of Elizabeth Bennet, the protagonist.",
    "image": "https://m.media-amazon.com/images/I/71uGj3+S4fL.jpg"
  },
  {
    "id": 5,
    "title": "The Catcher in the Rye",
    "author": "J.D. Salinger",
    "genre": "Fiction",
    "year": 1951,
    "pages": 234,
    "publisher": "Little, Brown and Company",
    "description": "A story about Holden Caulfield's experiences in New York City after being expelled from prep school.",
    "image": "https://m.media-amazon.com/images/I/81OthjkJBuL.jpg"
  },
  {
    "id": 6,
    "title": "The Hobbit",
    "author": "J.R.R. Tolkien",
    "genre": "Fantasy",
    "year": 1937,
    "pages": 310,
    "publisher": "George Allen & Unwin",
    "description": "A fantasy novel about the adventures of Bilbo Baggins, a hobbit, who sets out on a quest to win a share of a dragon's treasure.",
    "image": "https://m.media-amazon.com/images/I/91b0C2YNSrL.jpg"
  },
```

```
 {
   "id": 7,
   "title": "Fahrenheit 451",
   "author": "Ray Bradbury",
   "genre": "Dystopian",
   "year": 1953,
   "pages": 249,
   "publisher": "Ballantine Books",
   "description": "A dystopian novel about a future society where books are banned, and
'firemen' burn any that are found.",
   "image": "https://m.media-amazon.com/images/I/81fs6OavHXL.jpg"
 },
 {
   "id": 8,
   "title": "Moby-Dick",
   "author": "Herman Melville",
   "genre": "Adventure",
   "year": 1851,
   "pages": 635,
   "publisher": "Harper & Brothers",
   "description": "The story of Captain Ahab's obsessive quest to kill the white whale,
Moby-Dick.",
   "image": "https://m.media-amazon.com/images/I/81NnNwUCuXL.jpg"
 },
 {
   "id": 9,
   "title": "War and Peace",
   "author": "Leo Tolstoy",
   "genre": "Historical Fiction",
   "year": 1869,
   "pages": 1225,
   "publisher": "The Russian Messenger",
   "description": "A novel that chronicles the French invasion of Russia and the impact of the
Napoleonic era on Tsarist society.",
   "image": "https://m.media-amazon.com/images/I/81j7uPEEAtL.jpg"
 },
 {
   "id": 10,
   "title": "The Odyssey",
   "author": "Homer",
   "genre": "Epic Poetry",
   "year": -800,
   "pages": 541,
   "publisher": "Unknown",
```

```
    "description": "An epic poem that follows the Greek hero Odysseus on his journey home after the fall of Troy.",
    "image": "https://m.media-amazon.com/images/I/81c+GKhWhkL.jpg"
  }
]
```


🛠️ Step 1: Fetch the Book Data Asynchronously

- Fetch the dataset from an API or **simulate** it using a `setTimeout` function.
- Ensure error handling using `try...catch` or `.catch()` in Promises.
- Log the data to verify it's received correctly.

📝 **Task for Trainees:**

- Write a function that retrieves book data asynchronously.
- Ensure the function handles errors gracefully.

⚡ Step 2: Use Callbacks to Determine Special Book Criteria

- Introduce a **callback function** that evaluates books based on a specific property.
- Example: If a book is from the **Dystopian** genre, mark it as `"Caution: Dystopian Future"`.
- Trainees should **call the callback function** dynamically within the data.

📝 **Task for Trainees:**

- Implement a callback function that flags books from a particular genre or **high page count (e.g., > 500 pages)**.
- Log the books with a warning message if they match the criteria.

🔄 Step 3: Apply Higher-Order Functions to Manipulate the Data

🗂️ Use `.map()` to Display Books

- Create an array of formatted book summaries.
- Example:

```
[
  "📖 1984 by George Orwell - Dystopian (328 pages)",
  "📖 The Great Gatsby by F. Scott Fitzgerald - Fiction (180 pages)"
]
```

- Display these summaries in a structured format.

🎯 Use `.filter()` for Advanced Book Searching

- Allow filtering books based on **genre** or **year published**.
- Example: Get all books published before **1950**.

🔢 Use `.sort()` to Arrange Books

- Sort books by **year** (ascending or descending).
- Sort books by **number of pages**.

📝 **Task for Trainees:**

- Implement `.map()` to format book data neatly.
- Use `.filter()` to extract books based on **genre** or **publication year**.
- Apply `.sort()` to arrange books **alphabetically**, **by year**, or **page count**.

📌 Step 4: Dynamic Filtering & UI Simulation

- Build a function that dynamically applies filters **based on user input** (e.g., filter books live).
- Example: A user selects "Fantasy," and only **The Hobbit** appears.
- Implement a **simulated UI logic** (e.g., show a modal for books that match a special condition).

📝 **Task for Trainees:**

- Implement a function that **accepts a genre input** and returns only books matching that genre.

- Create a function that **sorts books by year or pages** dynamically.
- Simulate a **modal alert** (e.g., `"This book is a classic!"` if published before **1900**).

📝 Final Task: Create a Powerful Book Searching & Sorting System

- **Combine all the functions**:
  ✅ Fetch the book data.
  ✅ Apply `.map()`, `.filter()`, and `.sort()`.
  ✅ Dynamically filter books by genre, year, or page count.
  ✅ Display results in a neat format.

💡 Stretch Goals:

- Allow multiple filters at once (e.g., **Fantasy books published after 1950**).
- Simulate a UI button that sorts books in **ascending or descending order**.
- Improve error handling for **empty search results**.

🚀 Expected Outcome

By the end of this project, trainees will:
✔ Understand **asynchronous JavaScript** by fetching and handling data.
✔ Use **callbacks** dynamically.
✔ Master **higher-order array methods** (`.map()`, `.filter()`, `.sort()`).
✔ Build **real-world filtering and sorting logic**.